ISSN 0555-2923

РОССИЙСКАЯ АКАДЕМИЯ НАУК

Проблемы передачи информации

НАУКА
1727

том 60 вып. 4

2024

РОССИЙСКАЯ АКАДЕМИЯ НАУК

ПРОБЛЕМЫ ПЕРЕДАЧИ ИНФОРМАЦИИ

Журнал основан в январе 1965 г. **ISSN: 0555-2923** **Выходит 4 раза в год**

Том 60, 2024 **Вып. 4**

Октябрь–Ноябрь–Декабрь

Москва

СОДЕРЖАНИЕ

Семинары

CONTENTS

Coding Theory

Communication Network Theory

Image Processing

Seminars

ПРОБЛЕМЫ ПЕРЕДАЧИ ИНФОРМАЦИИ

Том 60 **2024** **Вып. 4**

УДК 621.391 : 519.725

© 2024 г. **Ж. Рифа[1], М. Вильянуэва[1], В.А. Зиновьев[2], Д.В. Зиновьев[2]**

О КРОНЕКЕРОВСКОЙ КОНСТРУКЦИИ РЕГУЛЯРНЫХ МАТРИЦ АДАМАРА И БЕНТ-ФУНКЦИЙ

Классическая кронекеровская конструкция применяется для построения новых матриц Адамара с новыми значениями ранга и размерности ядра. В частности, по двум матрицам Адамара H_1 и H_2 порядка n наша новая конструкция дает матрицу Адамара H порядка n^2. Если одна из исходных матриц Адамара линейна (т.е. строки матрицы, представленные в двоичном виде, замкнуты относительно их покомпонентного сложения), то получающаяся матрица Адамара H сводится к регулярной матрице, когда все строки имеют один и тот же вес, равный $n^2/2 - n/2$ (при двоичном $(0,1)$-представлении получившейся матрицы Адамара H). В частности, таким способом мы получаем бент-функции, т.е. строки полученной матрицы Адамара H являются бент-функциями. Построены матрицы Адамара, в которых каждая строка и каждый столбец является бент-функцией.

Ключевые слова: матрица Адамара, конструкция Кронекера, размерность ядра, схема Менона, ранг, регулярная матрица Адамара, бент-функция.

DOI: 10.31857/S0555292324040016, **EDN:** LGMDZT

§ 1. Введение

Пусть $v > k > \lambda \geqslant 0$ – натуральные числа. *Симметричная* (v, k, λ)*-схема* – это структура инцидентности (X, B), где $X = \{x_1, \ldots, x_v\}$ – множество из v элементов, а $B = \{B_1, \ldots, B_v\}$ – семейство k-подмножеств множества X (называемых блоками), таких что любые два различных элемента x_i, x_j встречаются вместе ровно в λ блоках из семейства B. Такую схему можно описать ее *матрицей инцидентности*, а именно двоичной матрицей $A = [a_{i,j}]$ порядка v, где $a_{i,j} = 1$ тогда и только тогда, когда $x_i \in B_j$.

Матрица Адамара H порядка n – это $(n \times n)$-матрица из элементов $+1$ и -1, такая что

$$HH^t = nI_n,$$

где I_n – двоичная диагональная матрица порядка n, а H^t – транспонированная матрица H. Хорошо известно, что порядок n матрицы Адамара H равен 1, 2 или $4m$ для любого натурального числа m [1]. Две матрицы Адамара *эквивалентны*, если одна может быть получена из другой перестановкой строк и/или столбцов и умножением строк и/или столбцов на -1. Используя эти операции, матрицу Адамара

[1] Исследования выполнены при частичной поддержке Национального гранта правительства Испании PID2022-137924NB-I00 (AEI 10.13039/501100011033), а также гранта правительства Каталонии (SGR 2021-00643).

[2] Исследования выполнены в ИППИ им. А.А. Харкевича РАН в рамках проводимых фундаментальных исследований по теме “Математические основы теории корректирующих кодов”.

всегда можно представить в *нормализованном* виде, где первая строка и первый столбец содержат только единицы. Будем говорить, что матрица Адамара *нормализована по строкам*, если первая строка содержит только единицы, и *нормализована по столбцам*, если первый столбец содержит только единицы.

Для двух заданных матриц $A = [a_{r,s}]$ и $B = [b_{i,j}]$ над одним и тем же кольцом без делителей нуля определим новую матрицу H, являющуюся *кронекеровым* (или *прямым*) *произведением* $H = A \otimes B$, где H получена заменой любого элемента $a_{r,s}$ матрицы A на матрицу $a_{r,s}B$. Самое первое известное семейство матриц Адамара, полученное Сильвестром, состоит из матриц порядка 2^n, где $n \geqslant 1$. Эти матрицы, называемые *сильвестровыми матрицами Адамара*, строятся с помощью кронекеровского произведения $\otimes^n(S_1)$, т.е. итерацией тензорного произведения матрицы

$$S_1 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

на саму себя.

Если все элементы $+1$ матрицы H заменить элементом 0, а элементы -1 заменить элементом 1, то получим *двоичную* $(0,1)$*-матрицу Адамара*, которую будем обозначать через H_b. Так как любые две строки совпадают в $n/2$ позициях и различны в $n/2$ позициях, легко видеть, что эти две строки находятся на расстоянии Хэмминга $n/2$ друг от друга. Двоичный $(n, 2n, n/2)$-код, состоящий из строк двоичной матрицы Адамара порядка n и дополнительных к ним строк, называется (*двоичным*) *кодом Адамара*. Если матрица Адамара линейна, то соответствующий двоичный код Адамара представляет собой хорошо известный код Рида – Маллера первого порядка $R(1, n)$ длины $n = 2^m$ и размерности $m + 1$.

Булева функция f – это отображение из двоичного пространства $\mathbb{Z}_2^m$ всех двоичных векторов длины m в кольцо $\mathbb{Z}_2$. Степень нелинейности булевой функции f определяется как минимальное расстояние Хэмминга между f и всеми аффинными функциями пространства $\mathbb{Z}_2^m$. Другими словами, это ее расстояние до кода Рида – Маллера первого порядка. Это расстояние ограничено сверху величиной

$$2^{m-1} - 2^{\frac{m}{2}-1},$$

причем в случае равенства (которое возможно только для четного m) функция называется *бент-функцией*. Одним из важных классов бент-функций является класс Майораны – Мак-Фарланда, введенный в [2] и представляющий собой булевы функции $f(x, y)$ от $2m$ переменных вида

$$f(x, y) = \langle x, \pi(y) \rangle + h(y) \quad \text{для любых } x, y \in \mathbb{Z}_2^m, \tag{1}$$

где π – произвольная перестановка на множестве $\mathbb{Z}_2^m$, а h – произвольная булева функция от m переменных.

Двумя важными параметрами двоичных кодов являются ранг и размерность ядра. *Ранг* двоичного кода C, обозначаемый $\mathrm{rank}(C)$, представляет собой размерность линейной оболочки $\langle C \rangle$, образованной кодовыми словами кода C. *Ядро* $\mathrm{K}(C)$ двоичного кода C длины n определяется как

$$\mathrm{K}(C) = \{\boldsymbol{x} \in \mathbb{Z}_2^n : \boldsymbol{x} + C = C\}.$$

Если код C содержит нулевой вектор, то ядро $\mathrm{K}(C)$ является линейным подкодом кода C. Легко видеть, что если C линеен, то ядро совпадает с кодом:

$$\mathrm{K}(C) = C = \langle C \rangle.$$

Обозначим через $\ker(C)$ размерность ядра кода C. Два этих параметра могут быть использованы для выяснения эквивалентности различных матриц Адамара или соответствующих кодов (см., например, [3]).

Хорошо известно, что нормализованная двоичная матрица Адамара порядка $4m$ существует тогда и только тогда, когда существует симметричная $(4m-1, 2m-1, m-1)$-схема [4]. Однако в дальнейшем мы будем рассматривать другой тип (v, k, λ)-схем, также связанных с матрицами Адамара. Будем говорить, что матрица Адамара порядка n *регулярна по строкам* (соответственно, *регулярна по столбцам*), если сумма элементов каждой строки (соответственно, каждого столбца) одна и та же для всех строк (соответственно, всех столбцов). Кроме того, будем говорить, что матрица Адамара порядка n *регулярна*, если сумма элементов каждой строки и каждого столбца одна и та же для всех ее строк и столбцов [1].

Хорошо известно, что регулярная двоичная матрица Адамара порядка ν является матрицей инцидентности симметричной $(\nu^2, \nu^2/2-\nu/2, \nu^2/4-\nu/2)$-схемы, обычно называемой *схемой Менона* [5]. И наоборот, матрица инцидентности симметричной $(\nu^2, \nu^2/2-\nu/2, \nu^2/4-\nu/2)$-схемы представляет собой двоичную регулярную матрицу Адамара порядка ν^2. Нетрудно видеть (и также хорошо известно), что матрица инцидентности симметричной (v, k, λ)-схемы при условии ортогональности $v = 4(k-\lambda)$ является двоичной регулярной матрицей Адамара порядка v, где v представляет собой полный квадрат.

Следующие две леммы относятся к регулярным матрицам Адамара. Первая из них представляет собой вариант хорошо известного результата Райзера [6], связанного с симметричными блок-схемами. Вторая лемма представляет собой важный результат о величине ядра любой двоичной регулярной матрицы Адамара.

Лемма 1. *Матрица Адамара, регулярная по строкам, регулярна, и ее порядок равен ν^2, где ν – сумма элементов строки или столбца.*

Доказательство. Пусть H – регулярная по строкам матрица Адамара порядка n, и предположим, что сумма каждой строки равна ν. Следовательно, $H\boldsymbol{u} = \nu\boldsymbol{u}$, где $\boldsymbol{u}$ – вектор из всех единиц. Тогда получаем

$$H^t H\boldsymbol{u} = \nu H^t\boldsymbol{u},$$

и так как H – матрица Адамара, то $n\boldsymbol{u} = \nu H^t\boldsymbol{u}$. Поэтому получаем

$$H^t\boldsymbol{u} = \frac{n}{\nu}\boldsymbol{u}.$$

Следовательно, все столбцы имеют одну и ту же сумму, равную $\frac{n}{\nu}$. Так как общая сумма всех элементов матрицы H совпадает с аналогичной суммой для матрицы H^t, получаем, что $n\frac{n}{\nu} = n\nu$, или $\frac{n}{\nu} = \nu$ (т.е. $n = \nu^2$). Это доказывает, что суммы по столбцам равны суммам по строкам, и поэтому матрица H регулярна. Мы также получили, что эта постоянная сумма ν удовлетворяет условию $\nu^2 = n$. ▲

Лемма 2. *Ядро двоичной регулярной матрицы Адамара порядка ν^2 содержит только нулевой вектор.*

Доказательство. Пусть H – двоичная регулярная матрица Адамара порядка n. Предположим, что $\boldsymbol{x}$ – ненулевая строка матрицы H, т.е. двоичный вектор длины $\nu^2 = n$, такой что $H + \boldsymbol{x} = H$. Это означает, что для любого вектора-строки матрицы H, скажем, $\boldsymbol{r}_1$, имеем $\boldsymbol{r}_1 + \boldsymbol{x} = \boldsymbol{r}_2$, где $\boldsymbol{r}_2$ – также вектор-строка матрицы H. Векторы $\boldsymbol{r}_1$ и $\boldsymbol{r}_2$ отличаются в $n/2$ позициях, следовательно, вес вектора $\boldsymbol{x}$ равен $n/2$. Зафиксируем любую ненулевую позицию вектора $\boldsymbol{x}$. Если прибавить $\boldsymbol{x}$ ко всем строкам матрицы H, мы получим в этой фиксированной позиции (где $\boldsymbol{x}$

имеет ненулевую позицию) столбец $\boldsymbol{h}$ матрицы $H + \boldsymbol{x}$ одного из двух весов:

$$n/2 + \sqrt{n}/2 \quad \text{или} \quad n/2 - \sqrt{n}/2.$$

Так как $\mathrm{wt}(\boldsymbol{h}) \neq n - \mathrm{wt}(\boldsymbol{h})$, эта операция меняет число единиц матрицы H в этом столбце. Прибавление же нулевых позиций вектора $\boldsymbol{x}$ не меняет числа единиц в соответствующем столбце. Так как $\mathrm{wt}(\boldsymbol{x}) = n/2$, заключаем, что число единиц в матрицах H и $H + \boldsymbol{x}$ различно, откуда следует, что $\boldsymbol{x}$ – нулевой вектор. ▲

В 1962 г. Менон [5] построил класс симметричных блок схем с параметрами

$$(2^{2m}, 2^{2m-1} \pm 2^{m-1}, 2^{2m-2} \pm 2^{m-1})$$

с помощью разностных множеств в абелевых группах, что непосредственно приводит к бент-функциям. В работе [7] эта конструкция была изучена в терминах полностью регулярных кодов, причем в ней был построен еще один класс таких функций. Затем идеи этой конструкции были перенесены на бент-функции в [8], где было получено очень простое описание всех симметричных квадратичных бент-функций. Симметричные ортогональные схемы (или регулярные матрицы Адамара) являются предметом активных исследований в течение более 50 лет (см. работу [1] и библиографию в ней).

Мейснер [9] предложил весьма общую конструкцию регулярных матриц Адамара, основанную на кронекеровском произведении. В качестве исходных матриц он использовал a^2 матриц Адамара порядка v и $2a$ матриц Адамара порядка a. При некоторых условиях на исходные матрицы получаемая матрица оказывается регулярной матрицей Адамара порядка a^2v. В частности, его конструкция дает симметричные и кососимметричные регулярные матрицы Адамара.

Цель настоящей статьи – описать общую конструкцию регулярных матриц Адамара и бент-функций. Наша конструкция представляет собой вариант конструкции Кронекера. В отличии от конструкции Мейснера в [9], наша конструкция фактически основана на четырех различных исходных матрицах порядка ν и ν^2, а именно на двух произвольных матрицах Адамара одинакового порядка ν, специальной двоичной перестановочной матрице порядка ν^2 и произвольном векторе длины ν^2 с элементами ± 1. Получаемая матрица L всегда является матрицей Адамара порядка ν^2 (теорема 1). Если одна из исходных матриц линейна, то получаемая матрица L является регулярной, а в случае, когда эта исходная матрица является сильвестровой матрицей Адамара, в получаемой матрице L каждая строка или каждый столбец (в зависимости от использования исходных матриц Адамара, т.е. от того, какая из этих матриц линейна) является бент-функцией (теорема 2).

В случае, когда обе начальные матрицы Адамара линейны и нормализованы, можно получить верхнюю границу на ранг получаемой двоичной матрицы Адамара L_b и гарантировать существование бент-функций максимальной алгебраической степени (теорема 3). В § 3 даны две явные конструкции, для которых почти во всех случаях мы знаем алгебраическую степень бент-функций и ранг получаемой матрицы Адамара. Используя конструкцию, приведенную в п. 3.2, мы получаем регулярные матрицы Адамара, в которых каждая строка и каждый столбец является бент-функцией (теорема 4).

§ 2. Матрицы Адамара порядка ν^2

Начнем с рассмотрения перестановок, имеющих некоторые специфические свойства, позволяющие нам получать бент-функции в каждой строке матрицы Адамара, задаваемой формулой (8).

Пусть $L = [\ell_{i,j}]$ – квадратная матрица порядка ν^2. Перенумеруем ее строки и столбцы парами чисел из алфавита $\{1, \ldots, \nu\}$. В этих обозначениях произвольный

элемент L имеет номер

$$\ell_{(a-1)\nu+x,(b-1)\nu+y} = \ell_{(a,x),(b,y)}, \quad 1 \leqslant a, x, b, y \leqslant \nu. \tag{2}$$

Назовем *блоком* с номером (a, b) подматрицу матрицы L, образованную элементами вида (2), где числа a и b фиксированы. Элемент $\ell_{(a,x),(b,y)}$ матрицы L называется элементом (x, y) блока (a, b).

Определение 1. Назовем c-перестановкой (блоковой перестановкой) перестановочную матрицу $P^{(c)}$ порядка ν^2, содержащую в каждом блоке ровно один ненулевой элемент 1.

Латинским квадратом порядка ν называется $(\nu \times \nu)$-матрица P, элементами которой являются ν разных чисел $1, 2, \ldots, \nu$, так что каждое число встречается по одному разу в каждой строке и в каждом столбце этой матрицы. Два латинских квадрата $P = [p_{ij}]$ и $Q = [q_{ij}]$ порядка ν ортогональны, если упорядоченная пара чисел (p_{ij}, q_{ij}) встречается ровно один раз, когда i, j пробегают все значения из множества $\{1, \ldots, \nu\}$.

Чтобы работать с блоковыми перестановками, удобно использовать двоичные квадратные матрицы порядка ν только с одним ненулевым элементом. Обозначим через $Z_{i,j}$ такую матрицу с одним ненулевым элементом 1 в позиции (i, j). Обозначим через σ произвольное отображение множества $\{Z_{i,j}\}$ в себя. В этих обозначениях c-перестановку $P^{(c)}$ можно представить в следующем виде:

$$P^{(c)} = \sum_{i,j=1}^{\nu} Z_{i,j} \otimes \sigma(Z_{i,j}).$$

Определение 2. Назовем lc-перестановкой (латинско-блоковой перестановкой) блоковую перестановочную матрицу $P^{(c)}$ порядка ν^2 следующего вида:

$$P^{(c)} = \sum_{i,j=1}^{\nu} Z_{i,j} \otimes \sigma(Z_{i,j}), \tag{3}$$

где σ – отображение множества $\{Z_{i,j}\}$ в себя, такое что для каждого индекса $i \in \in \{1, \ldots, \nu\}$ матрица $\sum\limits_{j=1}^{\nu} \sigma(Z_{i,j})$ является перестановочной.

Перед тем как рассматривать дальнейшие результаты, напомним некоторые известные факты о кронекеровском произведении. Пусть A, B, C, D – матрицы над одним и тем же кольцом без делителей нуля, такие что имеют смысл произведения матриц (AB) и (CD). Тогда имеют место следующие равенства:

$$(AB) \otimes (CD) = (A \otimes C)(B \otimes D). \tag{4}$$

Если A и B – квадратные матрицы порядка ν, то имеет место следующее равенство:

$$A \otimes B = K_{\nu^2}(B \otimes A)K_{\nu^2}, \tag{5}$$

где K_{ν^2} – перестановочная матрица, обычно называемая *коммутационной матрицей* порядка ν^2 [10] и определяемая как

$$K_{\nu^2} = \sum_{i,j=1}^{\nu} Z_{i,j} \otimes Z_{i,j}^t. \tag{6}$$

Лемма 3. *Пусть*

$$P^{(c)} = \sum_{i,j=1}^{\nu} Z_{i,j} \otimes \sigma(Z_{i,j})$$

– lc-перестановочная матрица порядка ν^2, σ – соответствующее отображение из множества $\{Z_{i,j}\}$ в себя, а K_{ν^2} – коммутационная матрица порядка ν^2, где $\nu \geqslant 4$. Тогда матрица $P^{(c)}K_{\nu^2}$ является lc-перестановочной.

Доказательство. Из соотношений (3)–(6) следует, что

$$P^{(c)}K_{\nu^2} = \left(\sum_{i,j=1}^{\nu} Z_{i,j} \otimes \sigma(Z_{i,j})\right)\left(\sum_{k,s=1}^{\nu} Z_{k,s} \otimes Z_{k,s}^t\right) =$$

$$= \sum_{i,j,k,s=1}^{\nu} Z_{i,j}Z_{k,s} \otimes \sigma(Z_{i,j})Z_{k,s}^t.$$

Матрица $Z_{i,j}Z_{k,s}$ всегда нулевая, кроме случая, когда $k = j$. Следовательно,

$$Z_{i,j}Z_{k,s} = \delta_{kj}Z_{i,s},$$

где $\delta_{kj} = 1$, если $k = j$, и $\delta_{kj} = 0$ в остальных случаях. Таким образом,

$$P^{(c)}K_{\nu^2} = \sum_{i,j,s=1}^{\nu} Z_{i,s} \otimes \sigma(Z_{i,j})Z_{j,s}^t = \sum_{i,s=1}^{\nu} Z_{i,s} \otimes \left(\sum_{j=1}^{\nu} \sigma(Z_{i,j})Z_{s,j}\right).$$

Так как $P^{(c)}$ является lc-перестановочной матрицей, то для любого индекса $i \in \in \{1, \ldots, \nu\}$ матрица $\sum\limits_{j=1}^{\nu} \sigma(Z_{i,j})$ является перестановочной. Заметим, что $\sum\limits_{j=1}^{\nu} \sigma(Z_{i,j})$ будет перестановочной матрицей, если и только если $\sigma(Z_{i,j}) = Z_{\alpha_j,\beta_j}$, где оба элемента α_j и β_j покрывают весь диапазон $\{1, \ldots, \nu\}$, когда $j \in \{1, \ldots, \nu\}$.

Для $j \in \{1, \ldots, \nu\}$ произведение $\sigma(Z_{i,j})Z_{s,j}$ всегда равно нулю, кроме случая, когда индекс $\sigma(Z_{i,j})$ имеет вид $Z_{x,s}$ для некоторого индекса x, который всегда существует, так как по определению lc-перестановки сумма $\sum\limits_{j=1}^{\nu} \sigma(Z_{i,j})$ является перестановочной матрицей. Это означает, что существует некоторое значение индекса j, скажем, y, такое что

$$\sum_{j=1}^{\nu} \sigma(Z_{i,j})Z_{s,j} = Z_{x,y}.$$

Тем самым, можно определить отображение

$$\sigma'(Z_{i,s}) = Z_{x,y}.$$

Таким образом, получаем, что

$$P^{(c)}K_{\nu^2} = \sum_{i,s=1}^{\nu} Z_{i,s} \otimes \sigma'(Z_{i,s}).$$

В силу определения 2, чтобы доказать, что $P^{(c)}K_{\nu^2}$ является lc-перестановочной, нужно проверить, что для любого индекса $i \in \{1, \ldots, \nu\}$ матрица $\sum\limits_{s=1}^{\nu} \sigma'(Z_{i,s})$

представляет собой перестановочную матрицу. Еще раз перепишем

$$\sum_{s=1}^{\nu}\sigma'(Z_{i,s})=\sum_{s,j=1}^{\nu}\sigma(Z_{i,j})Z_{s,j}=\sum_{j=1}^{\nu}\left(\sigma(Z_{i,j})\left(\sum_{s=1}^{\nu}Z_{s,j}\right)\right)=\sum_{j=1}^{\nu}\sigma(Z_{i,j})M_j,$$

где M_j – матрица, имеющая единицы только в j-м столбце и нули в остальных.

Следовательно, справедливо равенство

$$\sum_{s=1}^{\nu}\sigma'(Z_{i,s})=\sum_{j=1}^{\nu}(Z_{\alpha_j,\beta_j}M_j)=\sum_{j=1}^{\nu}Z_{\alpha_j,j}.$$

Теперь ясно, что оба индекса α_j и j покрывают весь диапазон $\{1,\dots,\nu\}$, когда $j\in\{1,\dots,\nu\}$. ▲

Пусть $\boldsymbol{c}=(c_1,c_2,\dots,c_{\nu^2})$ – вектор длины ν^2 с элементами ±1. Этот вектор $\boldsymbol{c}$ можно представить в блоковом виде $\boldsymbol{c}=(\boldsymbol{c}_1,\dots,\boldsymbol{c}_\nu)$, где каждый блок $\boldsymbol{c}_k$, $k=1,\dots,\nu$, имеет длину ν. Назовем такой вектор $\boldsymbol{c}$ *блочно-постоянным*, если он имеет постоянное значение на всех позициях каждого блока $\boldsymbol{c}_i$, т.е. блок $\boldsymbol{c}_i$ имеет вид $\boldsymbol{c}_i=c_i(1\,1\,\dots\,1\,1)$, где $c_i\in\{\pm1\}$. Следующее утверждение тривиально.

Лемма 4. *Пусть $\boldsymbol{c}$ – блочно-постоянный ±1-вектор длины ν^2, а M – матрица порядка $\nu\geqslant4$. Тогда*

$$\operatorname{diag}(\boldsymbol{c})\,(I_\nu\otimes M)=(I_\nu\otimes M)\operatorname{diag}(\boldsymbol{c}),$$

где $\operatorname{diag}(\boldsymbol{c})$ *означает квадратную двоичную диагональную матрицу, содержащую вектор $\boldsymbol{c}$ на диагонали и нули во всех других позициях.*

Лемма 5. *Пусть $P^{(c)}$ является c-перестановочной матрицей порядка ν^2, σ – соответствующее отображение множества $\{Z_{i,j}\}$ в себя, а $\boldsymbol{c}=(c_1,c_2,\dots,c_{\nu^2})$ – произвольный вектор длины ν^2 с элементами ±1, где $\nu\geqslant4$. Тогда*

$$P^{(c)}\operatorname{diag}(\boldsymbol{c})=\sum_{i,j=1}^{\nu}Z_{i,j}\otimes\varepsilon_{ij}\sigma(Z_{i,j}),$$

где $\sigma(Z_{i,j})=Z_{k,s}$ и $\varepsilon_{ij}=c_{s+(j-1)\nu}$.

Доказательство. Вектор $\boldsymbol{c}=(c_1,c_2,\dots,c_{\nu^2})$ можно представить в блоковом виде следующим образом: $\boldsymbol{c}=(\boldsymbol{c}_1,\dots,\boldsymbol{c}_\nu)$, где каждый блок $\boldsymbol{c}_t$ имеет вид

$$\boldsymbol{c}_t=(c_{1+(t-1)\nu},c_{2+(t-1)\nu},\dots,c_{\nu+(t-1)\nu}).$$

Следовательно, $\boldsymbol{c}=(c_{r+(t-1)\nu})$, где $r,t\in\{1,\dots,\nu\}$.

Поэтому можно записать

$$\operatorname{diag}(\boldsymbol{c})=\sum_{r,t=1}^{\nu}c_{r+(t-1)\nu}(Z_{t,t}\otimes Z_{r,r}).\tag{7}$$

Таким образом,

$$P^{(c)}\operatorname{diag}(\boldsymbol{c})=\left(\sum_{i,j=1}^{\nu}Z_{i,j}\otimes\sigma(Z_{i,j})\right)\operatorname{diag}(\boldsymbol{c})=$$

$$=\left(\sum_{i,j=1}^{\nu}Z_{i,j}\otimes\sigma(Z_{i,j})\right)\left(\sum_{r,t=1}^{\nu}c_{r+(t-1)\nu}(Z_{t,t}\otimes Z_{r,r})\right).$$

Используя (4) и тот факт, что для любых $a, b, c, d \in \{1, \ldots, \nu\}$ справедливо равенство $Z_{a,b}Z_{c,d} = \delta_{bc}Z_{a,d}$, где $\delta_{bc} = 1$, если $b = c$, и $\delta_{bc} = 0$ в противном случае, получаем

$$P^{(c)}\operatorname{diag}(\boldsymbol{c}) = \sum_{i,j,r,t=1}^{\nu} c_{r+(t-1)\nu}\delta_{jt}Z_{i,t} \otimes \delta_{sr}Z_{k,r},$$

где $\sigma(Z_{i,j}) = Z_{k,s}$. Следовательно,

$$P^{(c)}\operatorname{diag}(\boldsymbol{c}) = \sum_{i,j=1}^{\nu} Z_{i,j} \otimes c_{r+(j-1)\nu}\sigma(Z_{i,j}) = \sum_{i,j=1}^{\nu} Z_{i,j} \otimes \varepsilon_{ij}\sigma(Z_{i,j}),$$

где $\varepsilon_{ij} = c_{s+(j-1)\nu}$. ▲

Лемма 6. *Пусть $\boldsymbol{c}$ – произвольный вектор длины ν^2 с элементами ± 1, а K_{ν^2} – коммутационная матрица порядка ν^2, где $\nu \geqslant 4$. Тогда существует вектор $\boldsymbol{d}$ длины ν^2 с элементами ± 1, такой что*

$$\operatorname{diag}(\boldsymbol{c})K_{\nu^2} = K_{\nu^2}\operatorname{diag}(\boldsymbol{d}).$$

Доказательство. Так как K_{ν^2} является перестановочной матрицей, в качестве вектора $\boldsymbol{d}$ с элементами ± 1 можно выбрать вектор $\boldsymbol{d}$, такой что

$$\operatorname{diag}(\boldsymbol{d}) = K_{\nu^2}^{-1}\operatorname{diag}(\boldsymbol{c})K_{\nu^2}. \quad ▲$$

Заметим, что вектор $\boldsymbol{d}$ из элементов ± 1 длины ν^2 задается выражением

$$d_{t+(r-1)\nu} = c_{r+(t-1)\nu}.$$

Следующая теорема приведена в [11, 12].

Теорема 1. *Пусть ν – произвольное натуральное число, такое что существует матрица Адамара порядка ν. Пусть $P^{(c)}$ – c-перестановочная матрица порядка ν^2, а $\boldsymbol{c}$ – произвольный ± 1-вектор длины ν^2, и пусть H_1 и H_2 – любые матрицы Адамара порядка ν. Пусть L – матрица следующего вида*:

$$L = (I_\nu \otimes H_1)\, P^{(c)} \operatorname{diag}(\boldsymbol{c})\, (I_\nu \otimes H_2)\,. \tag{8}$$

Тогда L является матрицей Адамара порядка ν^2.

Доказательство этой теоремы можно найти в [13].

Одним из основных результатов данной статьи является следующий.

Теорема 2. *Пусть $P^{(c)}$ – lc-перестановочная матрица порядка ν^2, $\boldsymbol{c}$ – произвольный вектор длины ν^2 с элементами ± 1, H_1 – произвольная матрица Адамара порядка ν, и H_2 – сильвестрова матрица Адамара порядка $\nu \geqslant 4$.*

Пусть

$$L = (I_\nu \otimes H_1)\, P^{(c)} \operatorname{diag}(\boldsymbol{c})\, (I_\nu \otimes H_2)\,.$$

Тогда

(i) *Матрица L представляет собой матрицу Адамара с двумя возможными значениями $\pm\nu$ весов (т.е. суммы элементов) ее строк, и меняя знаки строк (умножением строк на -1), меняя тем самым сумму $-\nu$ на сумму ν, мы получаем регулярную матрицу Адамара L^* с суммой элементов каждой строки и каждого столбца, равной ν;*

(ii) *Каждая строка двоичной матрицы L_b (т.е. матрицы L_b^*, полученной из L^*) является бент-функцией типа Майораны – Мак-Фарланда.*

Доказательство. Так как $P^{(c)}$ является lc-перестановочной матрицей и, следовательно, c-перестановочной, из теоремы 1 мы получаем, что L является матрицей Адамара. Теперь вычислим скалярное произведение любой строки матрицы L с любой строкой сильвестровой матрицы Адамара порядка ν^2, которую можно представить в виде $H \otimes H$, где H – сильвестрова матрица Адамара порядка ν. Возьмем также в качестве H_2 сильвестрову матрицу Адамара H. Тогда получаем

$$\begin{aligned}
&L(H \otimes H)^t = (I_\nu \otimes H_1)\, P^{(c)}\, \mathrm{diag}(\boldsymbol{c})\, (I_\nu \otimes H)\, (H^t \otimes H^t) = \\
&= (I_\nu \otimes H_1)\, P^{(c)}\, \mathrm{diag}(\boldsymbol{c})(H^t \otimes \nu I_\nu) = \\
&= \nu\, (I_\nu \otimes H_1)\, P^{(c)}\, \mathrm{diag}(\boldsymbol{c}) K_{\nu^2} (I_\nu \otimes H^t) K_{\nu^2} = \\
&= \nu\, (I_\nu \otimes H_1)\, P^{(c)} K_{\nu^2}\, \mathrm{diag}(\boldsymbol{d}) (I_\nu \otimes H^t) K_{\nu^2}, \qquad (9)
\end{aligned}$$

где K_{ν^2} – коммутационная матрица (см. (6)), а $\boldsymbol{d}$ – вектор длины ν^2 с элементами ± 1, указанный в лемме 6. Из леммы 3 получаем, что $P^{(c)} K_{\nu^2}$ – lc-перестановочная матрица, и так как K_{ν^2} – перестановочная матрица, мы заключаем, что матрица

$$\frac{1}{\nu} L(H \otimes H)^t$$

является матрицей Адамара. Элементами матрицы $L(H \otimes H)^t$ являются $\pm\nu$. Следовательно, так как скалярное произведение любой строки L с любой строкой сильвестровой матрицы Адамара соответствует элементам матрицы $L(H \otimes H)^t$, мы получаем, что скалярные произведения равны $\pm\nu$. Так как первая строка сильвестровой матрицы Адамара $H \otimes H$ – это вектор из всех единиц, мы заключаем, что сумма элементов каждой строки матрицы L принимает два значения $\pm\nu$. Меняя знаки элементов строк с суммой $-\nu$ на противоположные (умножением этих строк на -1), мы получаем матрицу L^* с постоянной суммой элементов всех строк, равной ν, что означает регулярность по строкам матрицы L^*, а значит, по лемме 1, и регулярность с суммой элементов каждой строки и каждого столбца, равной ν, что доказывает утверждение (i).

Для доказательства утверждения (ii) необходимо проверить расстояние Хэмминга от любой строки L_b до сильвестровой матрицы Адамара $(H \otimes H)_b$. Из предыдущих рассуждений известно, что значения скалярного произведения строк матрицы L со строками матрицы $H \otimes H$ равны $\pm\nu$. Это означает, что расстояние Хэмминга между строками L_b и строками $(H \otimes H)_b$ равно $\dfrac{\nu^2 \pm \nu}{2}$, и поэтому минимальное расстояние между строками этих двух матриц равно $\dfrac{\nu^2 - \nu}{2}$. Отсюда вытекает, что любая строка матрицы L_b дает бент-функцию. Тот факт, что полученные бент-функции являются функциями типа Майораны – Мак-Фарланда, доказан в [13]. ▲

Теперь мы переходим к рассмотрению алгебраической степени построенных регулярных матриц Адамара. Напомним, что преобразование Мёбиуса булевой функции связывает таблицу истинности функции с ее алгебраической нормальной формой (АНФ). Здесь мы следуем работе [14].

В начале перечислим все векторы пространства $\mathbb{Z}_2^m$ в виде

$$\alpha_0 = (0, \ldots, 0), \quad \alpha_1 = (0, \ldots, 1), \quad \ldots, \quad \alpha_{2^m-1} = (1, \ldots, 1),$$

где α_i – двоичное представление целого числа i. Таблица истинности булевой функции f на $\mathbb{Z}_2^m$ представляет собой двоичную последовательность, определяемую следующим образом:

$$(f(\alpha_0), f(\alpha_1), \ldots, f(\alpha_{2^m-1})).$$

Функцию f можно единственным образом представить в виде АНФ как

$$f(x_1,\dots,x_m)=\bigoplus_{(a_1,\dots,a_m)} g(a_1,\dots,a_m)x_1^{a_1}\dots x_m^{a_m}, \tag{10}$$

где $(a_1,\dots,a_m)\in\mathbb{Z}_2^m$, а g – функция на $\mathbb{Z}_2^m$, которая называется *преобразованием Мёбиуса* функции f и обозначается через $g=\mu(f)$.

Для булевой функции f число переменных в самом длинном мономе ее АНФ называется ее *алгебраической степенью* и обозначается $\deg(f)$. Хорошо известно (см., например, [15]), что для любой бент-функции f с $2m$ переменными справедливо неравенство

$$\deg(f)\leqslant m.$$

Если $\deg(f)=m$, то f называется бент-функцией с *максимальной алгебраической степенью*.

Преобразование Мёбиуса задается двоичной $(2^m\times 2^m)$-матрицей T_m, i-я строка которой представляет собой таблицу истинности монома $x_1^{a_1}\dots x_m^{a_m}$, где $(a_1,\dots,a_m)$ – двоичное представление целого числа i. Матрицу T_m можно рекуррентно представить в следующем виде:

$$T_s=\begin{pmatrix} T_{s-1} & T_{s-1}\\ 0_{s-1} & T_{s-1}\end{pmatrix},$$

где 0_{s-1} – нулевая матрица размера $2^{s-1}\times 2^{s-1}$, матрица T_1 имеет вид

$$T_1=\begin{pmatrix} 1 & 1\\ 0 & 1\end{pmatrix},$$

а $s\geqslant 2$ – любое целое число. Более того, матрица T_m обладает следующим свойством:

$$T_m^{-1}=T_m,$$

и для заданной булевой функции f имеем

$$(f(\alpha_0),f(\alpha_1),\dots,f(\alpha_{2^m-1}))T_m=(g(\alpha_1),\dots,g(\alpha_{2^m-1})). \tag{11}$$

Алгебраическая степень бент-функции, заданной строками матрицы L_b в теореме 2, почти всегда максимальна, т.е. достигает максимума для бент-функции. Это максимальное значение равно m, где $\nu=2^m$. Следующая теорема, в которой добавлены некоторые ограничения на условия теоремы 2, показывает, что при изменении вектора $\boldsymbol{c}$ по крайней мере одна строка в матрице L_b даст бент-функцию с максимальной алгебраической степенью. Кроме того, мы получаем верхнюю границу на ранг полученной регулярной матрицы Адамара L_b^*, ассоциированной с матрицей L_b.

Теорема 3. *Пусть $P^{(c)}$ является lc-перестановочной матрицей порядка ν^2, где $\nu=2^m\geqslant 4$. Пусть $\boldsymbol{c}$ – произвольный вектор длины ν^2 с элементами ±1, и пусть H_1,H_2 – сильвестровы матрицы Адамара порядка ν.*

Пусть

$$L=(I_\nu\otimes H_1)\,P^{(c)}\,\mathrm{diag}(\boldsymbol{c})\,(I_\nu\otimes H_2),$$

и пусть L^* – регулярная матрица Адамара, ассоциированная с L (т.е. полученная умножением соответствующих строк на -1). Пусть L_b и L_b^* – двоичные матрицы, отвечающие матрицам L и L^* соответственно. Наконец, пусть i – любое число в диапазоне $i\in\{1,\dots,\nu^2\}$. Тогда

(i) $\mathrm{rank}(L_b^*) \leqslant 2\nu - 2$, *если* $\boldsymbol{c}$ *– постоянный блоковый вектор, и* $\mathrm{rank}(L_b^*) \leqslant 2\nu - 1$ *в противном случае*;
(ii) $\ker(L_b^*) = (0, 0, \ldots, 0)$;
(iii) *для заданного* $i \in \{1, \ldots, \nu^2\}$ *существует вектор* $\boldsymbol{c}$, *такой что* i*-я строка матрицы* L_b^* *является бент-функцией с максимальной алгебраической степенью* m.

Доказательство. Для доказательства первого утверждения заметим, что

$$P^{(c)} = \sum_{i,j=1}^{\nu} Z_{i,j} \otimes \sigma(Z_{i,j}),$$

где $\sigma(Z_{i,j}) = Z_{k,s}$, так что в множестве пар $\{(k,s)\}$, индуцированных при фиксации индекса i, имеются все значения $k, s \in \{1, \ldots, \nu\}$.

Выберем сильвестровы матрицы Адамара $H_1 = H_2 = H$. Возьмем также в качестве $\boldsymbol{c}$ постоянный вектор из всех единиц (в противном случае применим лемму 4). Тогда получаем

$$L = (I_\nu \otimes H_1)\, P^{(c)}\, (I_\nu \otimes H_2) =$$
$$= \sum_{i,j=1}^{\nu} (I_\nu \otimes H)\, (Z_{i,j} \otimes Z_{k,s})\, (I_\nu \otimes H) = \sum_{i,j=1}^{\nu} (Z_{i,j} \otimes H Z_{k,s} H).$$

Матрицу L можно рассматривать как матрицу, состоящую из блоков размера $\nu \times \nu$. Каждый блок матрицы L соответствует разным значениям индексов i, j (т.е. каждый из ν^2 блоков задан одной упорядоченной парой индексов (i,j)),

$$L = \sum_{i,j=1}^{\nu} (Z_{i,j} \otimes H Z_{k,s} H).$$

Пусть H_b и L_b – двоичные матрицы, отвечающие матрицам H и L соответственно. Заметим, что элемент с координатами (a,b) в блоке с координатной парой (i,j) матрицы $HZ_{k,s}H$ равен $h_{a,k}h_{s,b}$, где h_{xy} обозначает элемент (x,y) матрицы H. Для удобства обозначим через $\bar{h}_{a,b}$ двоичное значение элемента $h_{a,b}$, т.е. $\bar{h}_{a,b} = 0$, когда $h_{a,b} = 1$, и $\bar{h}_{a,b} = 1$, когда $h_{a,b} = -1$. Следовательно, элементом с координатами (a,b) матрицы $(HZ_{k,s}H)_b$ (т.е. блока (i,j)) является следующая сумма двух элементов:

$$\bar{h}_{a,k} + \bar{h}_{s,b}. \tag{12}$$

Рассмотрим теперь две строки матрицы L_b внутри одного и того же строчного блока. Величина $\bar{h}_{x,y}$ равна 0 или 1 в матрице H_b. Каждую строку можно разбить на ν последовательных столбцевых блоков, и можно убедиться, что элементы первого столбца этих двух строк равны $(\bar{h}_{a,k} + \bar{h}_{s,b})$ и $(\bar{h}_{a',k} + \bar{h}_{s,b})$ соответственно, где индексы k, s, a, a' фиксированы и $b \in \{1, \ldots, \nu\}$. Складывая эти две строки матрицы L_b, мы получим вектор, такой что все элементы в координатах первого столбцевого блока равны $\bar{h}_{a,k} + \bar{h}_{a',k}$. Для всех других элементов столбцевых блоков имеем такое же выражение, но с другим k. Так как величина $\bar{h}_{a,k} + \bar{h}_{a',k}$ не зависит от b, то это означает, что полученный вектор имеет постоянное значение 0 или 1 в каждом блоке. Если мы отождествим все координаты одного столбцевого блока в одну координату (а это можно сделать, так как все эти координаты имеют одно и то же значение), мы получим сумму двух строк с индексами a и a' матрицы H_b после перестановки столбцов в соответствии с σ.

Применяя теперь это же рассуждение ко всем ν строчным блокам в матрице L_b, мы каждый раз будем получать сумму тех же самых двух строк в матрице H_b с ин-

дексами a и a' после некоторой перестановки столбцов. Следовательно, множество S всех двоичных векторов, получаемых как сумма двух строк матрицы L_b в одном и том же строчном блоке, эквивалентны с точки зрения линейной зависимости множеству S' всех двоичных векторов, полученных как сумма двух строк матрицы H_b после некоторой перестановки столбцов. Следовательно, ранг S не выше $\nu-2$. Чтобы получить L^* из L, нужно добавить вектор из всех единиц. Следовательно, прибавляя вектор из всех единиц к строкам множества S', мы получим множество векторов длины ν четного веса, откуда

$$\operatorname{rank}(S^*) \leqslant \nu - 1,$$

где через S^* обозначено множество строк S с добавлением вектора из всех единиц.

Чтобы вычислить $\operatorname{rank}(L_b)$, рассмотрим вышеуказанное множество векторов S^*, а также все строки из V, где V – подмножество строк матрицы L_b, не содержащее двух строк из одного и того же строчного блока. Например, можно взять все строки в матрице H_b с координатами

$$(\bar{h}_{1,k} + \bar{h}_{s,1}, \bar{h}_{1,k} + \bar{h}_{s,2}, \ldots),$$

где $k, s \in \{1, \ldots, \nu\}$. Каждый вектор из множества V представляет собой последовательное повторение ν блоков, где каждый блок – это строка матрицы H_b. Полное число векторов в множестве V равно ν, но при этом сложение всех векторов из V дает нулевой вектор. Следовательно, $\operatorname{rank}(V) \leqslant \nu - 1$, и поэтому

$$\operatorname{rank}(L_b^*) \leqslant \nu - 1 + \nu - 1 = 2\nu - 2.$$

Если теперь $\boldsymbol{c}$ является блочно-постоянным вектором, но не вектором из всех единиц, то используя лемму 4 и приведенные выше рассуждения, легко убедиться, что ранг матрицы L_b^* удовлетворяет той же самой верхней границе.

Пусть теперь $\boldsymbol{c}$ является не блочно-постоянным вектором, а произвольным вектором с элементами ± 1. В этом случае используем лемму 5. При этом выражение (12) принимает вид

$$\bar{h}_{a,k} + \bar{h}_{s,b} + \delta_{ks},$$

где δ_{ks} принимает двоичные значения 0 или 1 в зависимости от значения $+1$ или -1 величины ε_{ij} в лемме 5 соответственно.

Теперь с помощью тех же рассуждений, которые использовались в случае, когда $\boldsymbol{c}$ является вектором из всех единиц, $\operatorname{rank}(S^*)$ вычисляется точно так же, без каких-либо изменений, но при построении множества V мы не можем гарантировать, что сложение всех векторов множества V дает нулевой вектор. Следовательно, мы можем лишь заключить, что $\operatorname{rank}(V) \leqslant \nu$, и поэтому

$$\operatorname{rank}(L_b^*) \leqslant 2\nu - 1.$$

На этом заканчивается доказательство первого утверждения теоремы.

Второе утверждение теоремы вытекает непосредственно из леммы 2.

Для доказательства третьего утверждения, следуя формуле (11), вычислим $g = fT_m$, где $\nu = 2^m$, f – любая строка матрицы L_b, а g задает АНФ-представление бент-функции, соответствующей строке f.

Рассмотрим $T_m^{(a)}$, т.е. a-й столбец матрицы T_m, где

$$a = 2 + 2^m(2^{m-1} - 1).$$

Вес Хэмминга двоичного представления числа a равен m. Это означает, что когда $fT_m^{(a)} = 1$, в выражении $g = fT_m$ заведомо имеется моном алгебраической степени m.

Следовательно, для доказательства утверждения (iii) нужно вычислить $fT_m^{(a)}$ для строк f матрицы L_b и проверить, что результат вычисления равен 1.

Выберем строку в матрице L_b, например, i-ю строку блока, для которого матрицы $Z_{k,s}$ имеют индексы $\{(k_1, s_1), \ldots, (k_\nu, s_\nu)\}$. Заметим, что в столбце $T_m^{(a)}$ всюду стоят нули, кроме следующих ν координатных позиций:

$$1,\ 2,\ \nu+1,\ \nu+2,\ 2\nu+1,\ 2\nu+2,\ \ldots,\ \Big(\frac{\nu}{2}-1\Big)\nu+1,\ \Big(\frac{\nu}{2}-1\Big)\nu+2.$$

Элементами выбранной строки являются $\bar{h}_{i,k} + \bar{h}_{s,j}$, где индекс i фиксирован, а $j \in \in \{1, \ldots, \nu\}$. Пара индексов (k, s) пробегает значения $\{(k_1, s_1), \ldots, (k_\nu, s_\nu)\}$. Вычисляя $fT_m^{(a)}$, получаем

$$fT_m^{(a)} = \bar{h}_{i,k_1} + \bar{h}_{s_1,1} + \bar{h}_{i,k_1} + \bar{h}_{s_1,2} + \bar{h}_{i,k_2} + \bar{h}_{s_2,1} + \bar{h}_{i,k_2} + \bar{h}_{s_2,2} + \ldots +$$
$$+ \bar{h}_{i,k_\mu} + \bar{h}_{s_\mu,1} + \bar{h}_{i,k_\mu} + \bar{h}_{s_\mu,2} = \bar{h}_{s_1,1} + \bar{h}_{s_1,2} + \bar{h}_{s_2,1} + \bar{h}_{s_2,2} + \ldots + \bar{h}_{s_\mu,1} + \bar{h}_{s_\mu,2},$$

где $\mu = \nu/2$. Это значение соответствует сложению всех элементов координатных позиций обоих столбцов, 1-го и 2-го, всех $\nu/2$ строк матрицы H_b с номерами $s_1, s_2, \ldots, s_\mu$. Так как H_b – нормализованная двоичная матрица Адамара, все элементы первого столбца являются нулями, и поэтому величина fT_m равна сумме всех координат второго столбца матрицы H_b, соответствующих $\nu/2$ разным строкам с номерами $s_1, s_2, \ldots, s_\mu$. Эта сумма не обязательно равна 1, однако мы можем изменить символ в координате s_1, сохраняя остальные значения в координатах $s_2, \ldots, s_\mu$ без изменения. Возьмем в качестве $\boldsymbol{c}$ вектор, все элементы которого равны 1, за исключением одного элемента -1 в позиции $c_{s_1+(j-1)\nu}$, где $\sigma(Z_{i,j}) = Z_{k_1,s_1}$. Теперь утверждение (iii) следует из леммы 5. ▲

Из теоремы 2 мы видим, что из lc-перестановочной матрицы порядка ν^2 мы получаем матрицу Адамара

$$L = (I_\nu \otimes H_1)\, P^{(c)}\, \mathrm{diag}(\boldsymbol{c})\, (I_\nu \otimes H_2)\,,$$

строками которой являются бент-функции. Однако столбцы L_b не обязательно дают бент-функции. Это происходит из-за того, что определение lc-перестановочной матрицы не "симметрично". Пусть $P^{(c)}$ – c-перестановочная матрица порядка ν^2. Чтобы эта матрица $P^{(c)}$ была lc-перестановочной, необходимо, чтобы выполнялось равенство

$$P^{(c)} = \sum_{i,j=1}^{\nu} Z_{i,j} \otimes \sigma(Z_{i,j}),$$

где σ – отображение множества матриц $\{Z_{i,j}\}$ в себя, такое что для каждого индекса i сумма таких матриц по j

$$\sum_{j=1}^{\nu} \sigma(Z_{i,j})$$

представляет собой перестановочную матрицу (напомним, что двоичная матрица Z_{ij} размера $\nu \times \nu$ имеет только один ненулевой элемент в позиции (i, j)). Чтобы получить матрицу Адамара L, в которой каждая строка и каждый столбец матрицы L_b является бент-функцией, мы должны гарантировать, что для каждого индекса j сумма таких матриц по индексу i дает перестановочную матрицу, т.е. для каждого

индекса j матрица

$$\sum_{i=1}^{\nu} \sigma(Z_{i,j})$$

является перестановочной. Достаточным условием для этого является наличие матрицы G с парой (k, s) в i-й строке и j-м столбце, такой что $\sigma(Z_{i,j}) = Z_{k,s}$. Такую матрицу можно задать с помощью двух латинских квадратов. Это приводит к следующему определению.

Определение 3. Пусть G и D – два взаимно ортогональных латинских квадрата порядка ν. Определим биективное отображение

$$\sigma(Z_{i,j}) = Z_{k,s},$$

где $G_{ij} = k$ и $D_{ij} = s$. Определим olc-перестановочную матрицу $P^{(c)}$ порядка ν^2 следующим образом:

$$P^{(c)} = \sum_{i,j=1}^{\nu} Z_{i,j} \otimes \sigma(Z_{i,j}). \tag{13}$$

Теперь непосредственным образом получаем следующий результат.

Теорема 4. *Пусть G и D – два произвольных взаимно ортогональных латинских квадрата порядка ν. Зададим биективное отображение $\sigma(Z_{i,j}) = Z_{k,s}$, где $G_{ij} = k$ и $D_{ij} = s$, и соответствующую olc-перестановочную матрицу*

$$P^{(c)} = \sum_{i,j=1}^{\nu} Z_{i,j} \otimes \sigma(Z_{i,j})$$

согласно определению 3. *Пусть H – сильвестрова матрица Адамара порядка ν. Тогда двоичная матрица Адамара L_b вида*

$$L = (I_\nu \otimes H)\, P^{(c)}\, (I_\nu \otimes H)$$

такова, что каждая ее строка и каждый ее столбец представляет собой бент-функцию.

Доказательство. Этот результат следует из теоремы 2, если принять во внимание, что обе суммы

$$\sum_{i=1}^{\nu} \sigma(Z_{i,j}) \quad \text{и} \quad \sum_{j=1}^{\nu} \sigma(Z_{i,j})$$

являются перестановочными матрицами для всех индексов $i, j \in \{1, \ldots, \nu\}$. В этом случае обе матрицы $P^{(c)}$ и $P^{(c)t}$ являются lc-перестановочными. ▲

Далее в §3 мы построим два бесконечных семейства lc-перестановочных матриц $P^{(c)}$ порядка ν^2 для $\nu = 2^m$, где $m \geqslant 2$ – натуральное число (но четное для первой конструкции). С помощью первой конструкции полученные lc-перестановки индуцируют двоичные матрицы Адамара L_b, в которых все их строки являются бент-функциями согласно теореме 2. В некоторых случаях эти бент-функции имеют максимальную алгебраическую степень в соответствии с теоремой 3. Размерность ядра равна нулю, а значения ранга матрицы L_b для небольших значений ν были вычислены, и эти результаты мы приводим ниже. Все матрицы L_b из семейства,

построенного второй конструкцией в п. 3.2, имеют то дополнительное свойство, что каждый столбец их матрицы L_b также является бент-функцией.

§ 3. Построение *lc*-перестановочных и *olc*-перестановочных матриц

Чтобы говорить об lc-перестановках согласно определению 2, нам нужно, чтобы сумма блоковых матриц $\sigma(Z_{i,j})$ для любого i давала перестановочную матрицу, скажем, $P_i^{(c)}$. Идея состоит в том, чтобы вернуться назад и из c-перестановок $P_i^{(c)}$ построить блоковые матрицы $\sigma(Z_{i,j})$ для матрицы $P^{(c)}$. На этой идее основаны следующие две конструкции (более подробно эти конструкции описаны в [13]).

3.1. Семейство *lc*-перестановок. Для каждого четного положительного натурального числа $m \geqslant 2$ предположим, что нам известны $\nu = 2^m$ c-перестановок порядка ν, которые обозначим через $P_1^{(c)}, \ldots, P_\nu^{(c)}$. Наша цель – представить $P_k^{(c)}$ в виде суммы блоковых матриц типа $Z_{i,j}$. Проблема, с которой мы сталкиваемся при разложении каждой матрицы $P_k^{(c)}$ на блоковые матрицы типа $Z_{i,j}$, состоит в том, что, например, две матрицы $Z_{i,a}$ и $Z_{i,b}$ появляются с разными значениями $a \neq b$, но с одним и тем же индексом i. В этом случае матрица $P^{(c)}$, которую мы хотим найти, не будет c-перестановочной порядка ν^2. Чтобы справиться с этой проблемой, помимо перестановок $P_k^{(c)}$ для $k \in \{1, \ldots, \nu\}$ мы будем строить перестановочную матрицу $P^{(c)}$, используя латинский квадрат Q порядка ν.

Конструкция 1. Пусть $m \geqslant 2$ – четное натуральное число, и пусть $\nu = 2^m$. Построение lc-перестановочной матрицы $P^{(c)}$ порядка ν^2 из ν c-перестановок $P_1^{(c)}$, $P_2^{(c)}, \ldots, P_\nu^{(c)}$ порядка ν и латинского квадрата $Q = [q_{i,j}]$ порядка ν состоит в следующем. Для каждой матрицы $P_k^{(c)}$ рассмотрим отдельно каждую ее строку, например, строку j, и найдем в этой строке позицию, где стоит элемент 1. Пусть, например, этот элемент стоит в i-й позиции. Далее мы вычисляем *индикатор* $\tau = q_{k,i}$. Строка с номером j матрицы $P_k^{(c)}$ записывается в качестве строки с номером j блока (k, τ) матрицы $P^{(c)}$, которую мы строим. Таким образом мы гарантируем, что построенная матрица $P^{(c)}$ будет lc-перестановочной.

3.2. Семейство *olc*-перестановок. Теперь опишем другую конструкцию.

Конструкция 2. Возьмем пару ортогональных латинских квадратов $G = [g_{i,j}]$ и $D = [d_{i,j}]$ порядка ν и построим olc-перестановочную матрицу $P^{(c)}$ следующим образом:

$$P^{(c)} = \sum_{i,j=1}^{\nu} Z_{i,j} \otimes \sigma(Z_{i,j}),$$

где $\sigma(Z_{i,j}) = Z_{k,s}$, число k – элемент латинского квадрата G с координатами (i, j), т.е. $g_{i,j} = k$, а число s – элемент латинского квадрата D с координатами (i, j), т.е. $s = d_{i,j}$.

В табл. 1 для небольших значений ν приведены полученные значения ранга $\mathrm{rank}(L_b^*)$ для постоянного или произвольного вектора $\boldsymbol{c}$ с элементами ± 1 для матриц L_b^*, построенных конструкцией 2. Кроме того, приведена верхняя граница (ВГ), заданная теоремой 3.

Матрицы L^*, построенные с помощью конструкции 2, имеют такие же свойства, как и приведенные в теоремах 1 и 2, лемме 2 и теореме 3 для случая lc-перестановочных матриц. Однако имеются некоторые исключения, такие как, например, тот факт, что не только строки, но и столбцы соответствующей двоичной матрицы L_b^*

Таблица 1

Ранги кодов из конструкции 2 и верхние границы

	Постоянный $\boldsymbol{c}$		Произвольный $\boldsymbol{c}$	
ν	Полученные ранги	ВГ	Полученные ранги	ВГ
2^2	6	6	6, 7	7
2^3	$[10, \ldots, 14]$	14	$[10, \ldots, 15]$	15
2^4	$[22, \ldots, 30]$	30	$[22, \ldots, 31]$	31
2^5	$[56, \ldots, 61]$	62	$[58, \ldots, 62]$	63
2^6	$[118, \ldots, 125]$	126	$[122, \ldots, 126]$	127

являются бент-функциями. Более того, результаты вычислений для небольших значений m показывают, что величина $\operatorname{rank}(L_b^*)$ почти всегда удовлетворяет верхней границе, указанной в теореме 3. В табл. 1 приведены некоторые результаты вычислений, полученных с помощью системы компьютерной алгебры Magma [16].

СПИСОК ЛИТЕРАТУРЫ

1. *Beth T., Jungnickel D., Lenz B.* Design Theory. Cambridge, UK: Cambridge Univ. Press, 1986.
2. *McFarland R.L.* A Family of Difference Sets in Non-cyclic Groups // J. Combin. Theory Ser. A. 1973. V. 15. № 1. P. 1–10. https://doi.org/10.1016/0097-3165(73)90031-9
3. *Phelps K.T., Rifà J., Villanueva M.* Rank and Kernel of Binary Hadamard Codes // IEEE Trans. Inform. Theory. 2005. V. 51. № 11. P. 3931–3937. https://doi.org/10.1109/TIT.2005.856940
4. *Bose R.C., Shrikhande S.S.* A Note on a Result in the Theory of Code Construction // Inform. Control. 1959. V. 2. № 2. P. 183–194. https://doi.org/10.1016/S0019-9958(59)90376-6
5. *Kesava Menon P.* On Difference Sets Whose Parameters Satisfy a Certain Relation // Proc. Amer. Math. Soc. 1962. V. 13. № 5. P. 739–745. https://doi.org/10.1090/S0002-9939-1962-0142471-0
6. *Ryser H.J.* A Note on a Combinatorial Problem // Proc. Amer. Math. Soc. 1950. V. 1. № 4. P. 422–424. https://doi.org/10.1090/S0002-9939-1950-0036732-5
7. *Borges J., Rifà J., Zinoviev V.* New Families of Completely Regular Codes and Their Corresponding Distance Regular Coset Graphs // Des. Codes Cryptogr. 2014. V. 70. № 1–2. P. 139–148. https://doi.org/10.1007/s10623-012-9713-3
8. *Rifà J., Zinoviev V.A.* On Binary Quadratic Symmetric Bent and Semi-Bent Functions // Mosc. Math. J. 2023. V. 23. № 1. P. 121–128. https://doi.org/10.17323/1609-4514-2023-23-1-121-128
9. *Meisner D.B.* On a Construction of Regular Hadamard Matrices // Atti Accad. Naz. Lincei Cl. Sci. Fis. Mat. Natur. Rend. Lincei Matem. Appl. Ser. 9. 1992. V. 3. № 4. P. 233–240.
10. *Magnus J.R., Neudecker H.* The Commutation Matrix: Some Properties and Applications // Ann. Statist. 1979. V. 7. № 2. P. 381–394. https://doi.org/10.1214/aos/1176344621
11. *Семаков Н.В., Зайцев Г.В., Зиновьев В.А.* Корреляционное декодирование блочных кодов методом быстрого преобразования Фурье – Адамара // Тр. 4-го Симпоз. по проблеме избыточности в информационных системах. Ч. 2. Тез. докл. Ленинград, 1970. С. 545–550.
12. *Зайцев Г.В., Зиновьев В.А., Семаков Н.В.* Быстрое корреляционное декодирование блочных кодов // Кодирование и передача дискретных сообщений в системах связи. М.: Наука, 1976. С. 76–85.
13. *Rifà J., Villanueva M., Zinoviev D.V., Zinoviev V.A.* On Constructions of Regular Hadamard Matrices and Bent Functions // Probl. Inf. Transm. 2024. V. 60. № 4 (to appear). https://doi.org/10.1134/S003294602404001X

14. *Pieprzyk J., Wang H., Zhang X.-M.* Möbius Transforms, Coincident Boolean Functions and Non-coincidence Property of Boolean Functions // Int. J. Comput. Math. 2011. V. 88. № 7. P. 1398–1416. https://doi.org/10.1080/00207160.2010.509428
15. *Rothaus O.S.* On "Bent" Functions // J. Combin. Theory Ser. A. 1976. V. 20. № 3. P. 300–305. https://doi.org/10.1016/0097-3165(76)90024-8
16. *Bosma W., Cannon J., Playoust C.* The Magma Algebra System. I: The User Language // J. Symbolic Comput. 1997. V. 24. № 3–4. P. 235–265. https://doi.org/10.1006/jsco.1996.0125

Рифа Жузеп (Rifà, Josep)
Вильянуэва Мерсе (Villanueva, Mercè)
Факультет информационно-коммуникационных технологий,
Независимый университет Барселоны,
Серданьола-дель-Вальес, Каталония, Испания
josep.rifa@uab.cat
merce.villanueva@uab.cat
Зиновьев Виктор Александрович
Зиновьев Дмитрий Викторович
Институт проблем передачи информации
им. А.А. Харкевича РАН, Москва
vazinov@iitp.ru
dzinov@gmail.com

Поступила в редакцию
10.07.2024
После доработки
21.11.2024
Принята к публикации
18.12.2024

ПРОБЛЕМЫ ПЕРЕДАЧИ ИНФОРМАЦИИ

Том 60 **2024** **Вып. 4**

УДК 621.391 : 519.725

© 2024 г. **П.В. Трифонов, Г.А. Трофимюк**

ПОСТРОЕНИЕ ПОЛЯРНЫХ КОДОВ С БОЛЬШИМИ ДВОИЧНЫМИ ЯДРАМИ[1,2]

Предложены методы для вычисления пропускной способности и параметров Бхаттачарьи битовых подканалов, задаваемых двоичным поляризующим преобразованием с большими ядрами. Верхние и нижние границы, связывающие пропускную способность и параметр Бхаттачарьи канала, используются для уточнения полученных оценок. Полученные оценки могут быть использованы для выбора множества замораживания в конструкции полярных кодов. Кроме того, представлен метод поиска оптимальной последовательности ядер в полярных кодах со смешанными ядрами.

Ключевые слова: полярные коды, большие ядра, полярные подкоды.

DOI: 10.31857/S0555292324040028, **EDN:** MEWIDB

§ 1. Введение

Открытие Э. Ариканом явления поляризации привело к созданию полярных кодов, которые позволяют достичь симметричной пропускной способности дискретного канала W без памяти с двоичным входом, а также обладают алгоритмами построения, кодирования и декодирования малой сложности [2]. Полярные коды были включены в стандарт беспроводной связи 5G. Полярные коды также можно применять в каналах множественного доступа и каналах ретрансляции [3–5].

Однако классическая конструкция Арикана поляризует канал довольно медленно. Поэтому для кодов практически применимой длины приходится передавать некоторые информационные символы по битовым подканалам посредственного качества. Это приводит к низкой корректирующей способности таких полярных кодов при декодировании методом последовательного исключения (ПИ). Улучшенные кодовые конструкции, такие как полярные коды с циклическим контролем по избыточности [6] и полярные подкоды [7, 8], для получения хорошей корректирующей способности требуют списочного ПИ-декодирования (СПИ-декодирования) с большим размером списка. Альтернативный подход – это использовать технику последовательного декодирования, что позволяет добиться корректирующей способности, близкой к корректирующей способности СПИ-декодера, с гораздо меньшей сложностью [9–11].

Ситуация значительно улучшается, если заменить матрицу Арикана размера 2×2, называемую ядром, на более крупную. Было показано, что полярные коды с достаточно большими ядрами обеспечивают более высокую скорость поляризации [12, 13] и достигают оптимальной экспоненты масштабирования [14, 15]. Более того, в некоторых случаях сложность СПИ-декодирования, необходимого полярным

[1] Исследование выполнено за счет гранта Российского научного фонда № 22-11-00208.

[2] В статье расширяются результаты, представленные в [1].

подкодам с большим ядром для достижения определенной требуемой корректирующей способности, ниже, чем в случае кодов с ядром Арикана [16].

Полярные коды с ядром Арикана можно строить с помощью метода эволюции плотности [17, 18], гауссовской аппроксимации [19], рекурсии двоичного стирающего канала [2], или моделирования по методу Монте-Карло [20]. Для больших ядер описаны только два последних метода, за исключением случая ядер, полученных из матрицы Арикана [21]. Однако коды, построенные для двоичного стирающего канала [22], могут оказаться не оптимальными для АБГШ-канала, что на практике гораздо важнее.

В настоящей статье представлен приближенный метод вычисления пропускной способности и параметров Бхаттачарьи битовых подканалов, индуцированных двоичным поляризующим преобразованием, для АБГШ-канала с двоичной фазовой манипуляцией. Показано, что этот метод можно использовать не только для выбора замороженного множества для полярных кодов, но и для поиска оптимальной последовательности ядер в конструкции полярных кодов со смешанными ядрами. Предлагаемый подход опирается на семейство функций, зависящих от ядра, которые строятся однократно для каждого ядра методом Монте-Карло, а затем используются для построения кодов с произвольной комбинацией ядер с весьма низкой сложностью.

Статья построена следующим образом. В § 2 приводятся некоторые сведения о полярных кодах. В § 3 изложен предлагаемый подход к оценке надежности битовых подканалов. В § 4 изучается применение предложенного метода к вопросу упорядочивания ядер в полярных кодах со смешанными ядрами. Результаты моделирования представлены в § 5.

Расширенная версия данной статьи представлена в [23].

§ 2. Необходимые сведения

2.1. Полярные коды. Поляризующее преобразование задается матрицей

$$A = \mathcal{K}_1 \otimes \mathcal{K}_2 \otimes \ldots \otimes \mathcal{K}_m,$$

где $\mathcal{K}_i$ – ядра поляризации размера l_i, т.е. невырожденные $(l_i \times l_i)$-матрицы, которые перестановками столбцов нельзя перевести в верхнетреугольные [12]. Можно показать, что симметричный канал без памяти с двоичным входом

$$\mathbf{W}(y \,|\, c) = \boldsymbol{W}_0^{(0)}(y \,|\, c)$$

вместе с матрицей A порождает $n = \prod\limits_{i=1}^{m} l_i$ битовых подканалов

$$\mathbf{W}_m^{(i)}(y_0^{n-1}, u_0^{i-1} \,|\, u_i) = \mathbf{W}_{m,A}^{(i)}(y_0^{n-1}, u_0^{i-1} \,|\, u_i) =$$
$$= \frac{1}{2^{n-1}} \sum_{u_{i+1}^{n-1} \in \mathbb{F}_2^{n-i-1}} \prod_{j=0}^{n-1} \mathbf{W}_0^{(0)}(y_j \,|\, (u_0^{n-1}A)_j). \tag{1}$$

Если все подканалы $\mathcal{K}_i$ одинаковы, то при стремлении m к бесконечности пропускные способности этих подканалов сходятся к 0 и 1, а доля бесшумных подканалов приближается к пропускной способности C канала $\mathbf{W}$ [12]. На практике может оказаться целесообразным использовать конструкции со смешанными ядрами, т.е. задавать $\mathcal{K}_i$ различными матрицами [24].

Полярный (n, k)-код над $\mathbb{F}_2$ – это множество кодовых слов $c_0^{n-1} = u_0^{n-1}A$, где $u_i = 0$, $i \in \mathcal{F}$, $\mathcal{F} \subset [n]$ – множество замораживания, $[n] = \{0, 1, \ldots, n-1\}$ и $|\mathcal{F}| = n-k$.

В классической конструкции полярных кодов предполагается, что $\mathcal{F}$ – множество индексов подканалов $\mathbf{W}_m^{(i)}$ с наименьшими значениями пропускной способности.

Удобно рассматривать вероятности

$$\begin{aligned}
&\mathbf{W}_m^{(i)}(u_0^i \,|\, y_0^{n-1}) = \sum_{u_{i+1}^{n-1}} \mathbf{W}_m^{(n-1)}(u_0^{n-1} \,|\, y_0^{n-1}) = \\
&= \sum_{u_{i+1}^{n-1}} \mathbf{W}_m^{(n-1)}(y_0^{n-1} \,|\, u_0^{n-1}) \frac{P\{u_0^{n-1}\}}{W(y_0^{n-1})} = \frac{2^{-n}}{W(y_0^{n-1})} \sum_{u_{i+1}^{n-1}} \prod_{i=0}^{n-1} \mathbf{W}(y_i \,|\, (u_0^{n-1}A)_i) = \\
&= \underbrace{\frac{\prod\limits_{i=0}^{n-1} W(y_i)}{W(y_0^{n-1})}}_{\alpha} \sum_{u_{i+1}^{n-1}} \prod_{i=0}^{n-1} \mathbf{W}((u_0^{n-1}A)_i \,|\, y_i), \qquad (2)
\end{aligned}$$

где α – нормирующий множитель, а $W(y)$ – плотность вероятности на выходе канала. Эти вероятности можно вычислить рекуррентно как

$$\mathbf{W}_p^{(l_p i+s)}(u_0^{l_p i+s} \,|\, y_0^{L_p-1}) = \alpha' \sum_{u_{l_p i+s+1}^{l_p(i+1)-1}} \prod_{j=0}^{l_p-1} \mathbf{W}_{p-1}^{(i)} \left(v_{0j}, \ldots, v_{ij} \,|\, y_{j,l_p}^{L_p-1}\right), \qquad (3)$$

где $v_{tj} = (u_{l_p t}^{l_p(t+1)-1} \mathcal{K}_p)_j$, $L_p = \prod\limits_{j=1}^{p} l_j$ – размер поляризующего преобразования, полученного на p-м уровне,

$$y_{j,l}^{n-1} = (y_j, y_{j+l}, \ldots, y_{j+n-l}), \quad 0 < p \leqslant m, \quad 0 \leqslant i < L_{p-1}, \quad 0 \leqslant s < l_p,$$

а α' – еще один нормирующий множитель. Эта операция известна как обработка ядра, или маргинализация ядра [25, 26]. Вычисление этого выражения сводится к мягкому декодированию в смежном классе по несистематическому линейному коду $\mathcal{C}_i$, порожденному строками $i, \ldots, l_p - 1$ матрицы $\mathcal{K}_p$. Его можно реализовать, используя алгоритм типа BCJR (Bahl, Cocke, Jelinek, Raviv) над расширенной кодовой решеткой этого кода [27]. Удобно определить логарифмическое отношение правдоподобия (ЛОПП)

$$\mathbf{L}_p^{(i)}(u_0^{i-1}, y_0^{L_p-1}) = \log \frac{\mathbf{W}_p^{(i)}(u_0^{i-1}, 0 \,|\, y_0^{L_p-1})}{\mathbf{W}_p^{(i)}(u_0^{i-1}, 1 \,|\, y_0^{L_p-1})}. \qquad (4)$$

Декодирование полярных кодов можно реализовать с помощью ПИ-алгоритма, принимающего решения

$$\widehat{u}_i = \begin{cases} \arg\max\limits_{u_i \in \mathbb{F}_2} \mathbf{W}_m^{(i)}(\widehat{u}_0^{i-1}, u_i \,|\, y_0^{n-1}), & i \notin \mathcal{F}, \\ \text{замороженное значение } u_i, & i \in \mathcal{F}. \end{cases} \qquad (5)$$

В [10] было предложено заменить сумму в (2) и (3) максимальным членом, т.е. рассматривать вероятности

$$\widetilde{\mathbf{W}}_p^{(l_p j+i)}(u_0^{l_p i+s} \,|\, y_0^{L_p-1}) = \max_{u_{l_p i+s+1}^{l_p(i+1)-1}} \prod_{j=0}^{l_p-1} \mathbf{W}_{p-1}^{(i)} \left(v_{0j}, \ldots, v_{ij} \,|\, y_{j,l_p}^{L_p-1}\right).$$

Эти вероятности можно использовать для аппроксимации $\mathbf{W}_m^{(j)}(u_0^j \,|\, y_0^{N-1})$ в алгоритме ПИ-декодирования. Еще один подход состоит в том, чтобы использовать приближенные значения ЛОПП

$$\mathcal{L}_p^{(i)}(u_0^{i-1}, y_0^{L_p-1}) = \log \frac{\widetilde{\mathbf{W}}_p^{(i)}(u_0^{i-1}, 0 \,|\, y_0^{L_p-1})}{\widetilde{\mathbf{W}}_p^{(i)}(u_0^{i-1}, 1 \,|\, y_0^{L_p-1})}. \tag{6}$$

Для вычисления этих значений имеются быстрые точные и приближенные алгоритмы обработки ядра [10,16,28–30]. Можно показать, что соответствующие алгоритмы декодирования обеспечивают очень хороший компромисс между корректирующей способностью и сложностью [31].

2.2. Параметр Бхаттачарьи. Надежность битовых подканалов $\mathbf{W}_\lambda^{(i)}$ удобно характеризовать их параметрами Бхаттачарьи $Z_{\lambda,i}$.

Симметричная пропускная способность I и параметр Бхаттачарьи Z симметричного канала без памяти с двоичным входом удовлетворяют неравенствам [2]

$$\log_2 \frac{2}{1+Z} \leqslant I \leqslant \sqrt{1-Z^2}. \tag{7}$$

Более точная оценка дана в [32, формула (23)]:

$$\widetilde{\mathcal{I}}(Z) = 1 - Z \leqslant I \leqslant \widehat{\mathcal{I}}(Z), \tag{8}$$

где величина $\widehat{\mathcal{I}}(Z)$ такова, что $B(1, \widehat{\mathcal{I}}(Z)) = \dfrac{Z+1}{2}$,

$$B(\rho, C) = \frac{\left((h^{-1}(1-C))^{\frac{1}{1+\rho}} + (1-h^{-1}(1-C))^{\frac{1}{1+\rho}}\right)^{1+\rho}}{2^\rho}$$

и $h(p) = -p\log_2 p - (1-p)\log_2(1-p)$. Это выражение можно упростить, получая

$$\widehat{\mathcal{I}}(Z) = 1 - h\left(\frac{Z^2}{2(1+\sqrt{1-Z^2})}\right).$$

Альтернативный вариант – переписать (8) в виде

$$\underbrace{1-I}_{\widetilde{\mathcal{Z}}(I)} \leqslant Z \leqslant \underbrace{2\sqrt{h^{-1}(1-I)(1-h^{-1}(1-I))}}_{\widehat{\mathcal{Z}}(I)}. \tag{9}$$

Выражения для параметров Бхаттачарьи битовых подканалов, индуцированных поляризационной матрицей Арикана, были получены в [33].

2.3. Показатели качества ядра. Говорят, что ядро $\mathcal{K}$ имеет скорость поляризации (называемую также экспонентой ошибки) $E(\mathcal{K})$, если параметры Бхаттачарьи $Z_m^{(i)}$ битовых подканалов, индуцированных матрицей $\mathcal{K}^{\otimes m}$, удовлетворяют следующим условиям [12]:

- Для любого фиксированного $\beta < E(\mathcal{K})$ выполнено равенство

$$\liminf_{n\to\infty} \Pr\left[Z_n \leqslant 2^{-\ell^{n\beta}}\right] = C;$$

- Для любого фиксированного $\beta > E(\mathcal{K})$ выполнено равенство

$$\liminf_{n\to\infty} \Pr\left[Z_n \geqslant 2^{-\ell^{n\beta}}\right] = 1.$$

Скорость поляризации можно вычислить как

$$E(\mathcal{K}) = \frac{1}{l}\sum_{i=0}^{l-1} \log_l D_i,$$

где D_i – i-е частичное расстояние для $\mathcal{K}$, т.е. минимальное расстояние Хэмминга между i-й строкой матрицы $\mathcal{K}$ и линейным пространством, образованным строками $i+1,\ldots,l-1$ из $\mathcal{K}$ [12]. Вероятность ошибки ПИ-декодирования для полярного кода со скоростью $\dfrac{k}{n} < C$ удовлетворяет неравенству

$$P_{SC} \leqslant 2^{-n^\beta}, \quad \beta < E(K).$$

Экспонента масштабирования μ для семейства корректирующих кодов со скоростью R определяет длину $n = O((C-R)^{-\mu})$ наиболее короткого кода из этого семейства, с помощью которого можно передавать информацию по каналу с пропускной способностью C с заданной вероятностью ошибки. Численный метод нахождения экспоненты масштабирования для ядра двоичного стирающего канала приведен в [34].

2.4. Построение множества замораживания. Множество замораживания $\mathcal{F}$ для полярных кодов обычно выбирается как множество индексов i наименее надежных битовых подканалов $\mathbf{W}_m^{(i)}$. Если исходный канал $\mathbf{W}_0^{(0)}$ – это двоичный стирающий канал, то все $\mathbf{W}_m^{(i)}$ также таковы, и их пропускные способности можно узнать через поляризационное поведение соответствующих ядер [22, 35, 36]. Кроме того, для оценки вероятностей ошибок в этих подканалах можно использовать моделирование методом Монте-Карло ПИ-декодера с участием джинна [2]. Сложность такого подхода составляет $O(Tn\log n)$ операций обработки ядра, где $T = \dfrac{\tau}{p}$ – число итераций метода Монте-Карло, p – вероятность ошибки в наименее надежном подканале, соответствующем незамороженному символу, а τ – достаточно большое целое число.

В случае полярных кодов с ядром Арикана можно вычислить ухудшенные и улучшенные приближения $\mathbf{W}_m^{(i)}$, а также их параметры Бхаттачарьи, с произвольно хорошей точностью и полиномиальной сложностью [18]. Также можно вычислить плотности вероятностей $\mathbf{L}_p^{(i)}(0, y_0^{L_p-1})$ или $\mathcal{L}_p^{(i)}(0, y_0^{L_p-1})$ в предположении передачи нулевого кодового слова [17, 37]. Поляризационная конструкция с весами, введенная в [38], позволяет получать коды с высокой эффективностью при списочном ПИ-декодировании. К сожалению, эти методы пока не были распространены на случай больших ядер, поскольку в общем случае крайне сложно охарактеризовать распределение ЛОПП $\mathbf{L}_p^{(i)}$ или $\mathcal{L}_p^{(i)}$.

Метод гауссовской аппроксимации был предложен в [19] для случая полярных кодов с ядром Арикана. В работе [21] он был распространен на случай некоторых других подобных ядер.

2.5. Полярные подкоды. Известно, что классические полярные коды имеют довольно низкие минимальные расстояния [39]. Их эффективность при СПИ-декодировании можно существенно улучшить, заменив статические ограничения замораживания на символы $u_i = 0$, $i \in \mathcal{F}$, динамическими ограничениями

$$u_i = \sum_{j<i} V_{s_i,j} u_j, \quad i \in \mathcal{F}, \tag{10}$$

где V – матрица ограничений размера $(n-k)\times n$, в которой различные строки заканчиваются[3] в различных столбцах $i\in\mathcal{F}$, а s_i – номер строки, заканчивающейся в столбце i. При таких ограничениях из классического полярного кода исключаются многие кодовые слова с малым весом. Символы u_i с хотя бы одним слагаемым в правой части (10) называются динамически замороженными, а символы с $V_{s_i,j}=0$, $j<s_i$, – статически замороженными. Полученные коды называются полярными подкодами. Алгебраическая конструкция полярных подкодов была представлена в [7]. Она позволяет получить коды с улучшенным минимальным расстоянием. Однако на практике более хорошие характеристики получаются при использовании рандомизированных конструкций [8, 40–42].

В [1] была предложена рандомизированная конструкция полярных подкодов с большими ядрами. Пусть $(\mathcal{D}_0,\dots,\mathcal{D}_{n-1})$ – частичный профиль расстояний матрицы A. Множество замораживания для рандомизированного полярного (n,k)-подкода определяется как

$$\mathcal{F}=\mathcal{F}_0\cup\mathcal{F}_A\cup\mathcal{F}_B,$$

где $\mathcal{F}_0$ – множество индексов $n-k-t_A-t_B$ наименее надежных битовых подканалов $\mathbf{W}_m^{(i)}$, $\mathcal{F}_A$ – множество из t_A максимальных индексов в $[n]\setminus\mathcal{F}_0$ с наименьшим $\mathcal{D}_i$ (динамически замороженные символы типа A), $\mathcal{F}_B$ – множество индексов $i\in[n]\setminus(\mathcal{F}_0\cup\mathcal{F}_A)$ наименее надежных битовых подканалов $\mathbf{W}_m^{(i)}$ (динамически замороженные символы типа B), t_A и t_B – параметры конструкции. Коэффициенты $V_{s_i,j}$, $i\in\mathcal{F}_0$, $j\ne i$, полагаются равными 0 (статически замороженные символы), а остальные выбираются как независимые равновероятные двоичные значения. Оптимальные значения параметров t_A и t_B обычно получают путем моделирования.

Такая конструкция гарантирует, что код имеет достаточно низкую вероятность ошибки ПИ-декодирования, а также позволяет избежать большинства кодовых слов малого веса. Это позволяет полярным подкодам обеспечить хорошую корректирующую способность при использовании алгоритма списочного декодирования Таля – Варди [6]. Другой способ получения кодов с улучшенными характеристиками – использовать предварительное сверточное кодирование [43–45].

Заметим, что использование динамически замороженных символов не влияет на вероятность ошибки в битовых подканалах, если декодер следует по правильному пути, поэтому методы, разработанные для оценки надежности битовых подканалов для классических полярных кодов, можно непосредственно перенести на случай полярных подкодов.

§ 3. Оценка надежности битовых подканалов

В этом параграфе мы опишем несколько методов оценки надежности битовых подканалов и, объединяя их, получим новый метод построения полярных кодов с большими ядрами.

3.1. Пропускные способности кодированных каналов. Рассмотрим случай $m=1$ и ядра $\mathcal{K}_1=K$ размерности l. Рассмотрим битовый подканал

$$\mathbf{W}_1^{(i)}(Y_0^{l-1},U_0^{i-1}\,|\,U_i)=\frac{1}{2^{l-1}}\sum_{u_{i+1}^{l-1}}\mathbf{W}(Y_0^{l-1}\,|\,U_0^{l-1}K),$$

[3] Для заданного двоичного вектора a_0^{n-1} мы говорим, что он заканчивается в позиции j, если $a_j=1$ и $a_t=0$, $j<t<n$.

индуцированный некоторым ядром K размера $l \times l$ и двоичным входным каналом $\mathbf{W}(Y \,|\, C)$, где

$$\mathbf{W}(Y_0^{l-1} \,|\, C_0^{l-1}) = \prod_{i=0}^{l-1} \boldsymbol{W}(Y_i \,|\, C_i),$$

а U_i, C_i, Y_i – случайные величины, отвечающие входным значениям поляризующего преобразования, входным и выходным символам канала, соответственно. Взаимная информация для $\mathbf{W}_1^{(i)}$ определяется следующим образом:

$$\mathbb{I}_i = I(Y_0^{l-1}, U_0^{i-1}; U_i) = I(Y_0^{l-1}; U_i \,|\, U_0^{i-1}) + I(U_0^{i-1}; U_i).$$

Поскольку случайные величины U_j, $0 \leqslant j < l$, предполагаются независимыми, последнее слагаемое равно 0. Далее, согласно правилу сложения для взаимной информации получаем

$$\mathbb{I}_i = I(Y_0^{l-1}; U_i \,|\, U_0^{i-1}) = I(Y_0^{l-1}; U_i^{l-1} \,|\, U_0^{i-1}) - I(Y_0^{l-1}; U_{i+1}^{l-1} \,|\, U_0^{i}). \tag{11}$$

Для симметричного канала $\mathbf{W}_0^{(0)}$ величина $I(Y_0^{l-1}; U_i^{l-1} \,|\, U_0^{i-1})$ не зависит от значений U_0^{i-1}, и эти значения можно считать равными 0. Действительно, значения U_0^{i-1} задают смежный класс по коду $\mathcal{C}_i$, порожденному строками $i, \ldots, l-1$ ядра K. Для симметричного канала существует перестановка $\pi(c)$, такая что $\pi = \pi^{-1}$ и $\mathbf{W}(y \,|\, 0) = \mathbf{W}(\pi(y) \,|\, 1)$. Отсюда следует, что для фиксированного смежного класса можно применить обратимое преобразование на выходе канала и получить эквивалентный выход канала, соответствующий передаче кодового слова из $\mathcal{C}_i$.

Тогда

$$I_i = I(Y_0^{l-1}; U_i^{l-1} \,|\, U_0^{i-1} = 0)$$

становится пропускной способностью канала, состоящего из кодера $\mathcal{C}_i$ и исходного канала $\mathbf{W}_0^{(0)}$.

Из определения взаимной информации следует, что

$$\begin{aligned}
\mathbb{I}_i &= 2^{i-l} \times \\
&\times \int\limits_{y_0^{l-1} \in \mathbb{R}^l} \sum_{u_i^{l-1} \in \mathbb{F}_2^{l-i}} \mathbf{W}(y_0^{l-1} \,|\, u_i^{l-1} K^{(i)}) \log_2 \frac{2^{l-i} \mathbf{W}(y_0^{l-1} \,|\, u_i^{l-1} K^{(i)})}{\sum\limits_{v_i^{l-1} \in \mathbb{F}_2^{l-i}} \mathbf{W}(y_0^{l-1} \,|\, v_i^{l-1} K^{(i)})} \, dy_0^{l-1} = \\
&= l - i + \int\limits_{y_0^{l-1} \in \mathbb{R}^l} \mathbf{W}(y_0^{l-1} \,|\, 0) \log_2 \frac{\mathbf{W}(y_0^{l-1} \,|\, 0)}{\sum\limits_{c' \in \mathcal{C}_i} \mathbf{W}(y_0^{l-1} \,|\, c')} \, dy_0^{l-1},
\end{aligned} \tag{12}$$

где $K^{(i)}$ – матрица, состоящая из строк $i, \ldots, l-1$ матрицы K. В следующих пунктах описаны методы, которые можно использовать для вычисления $\mathbb{I}_i$ для некоторых каналов. Как и можно было бы ожидать, эти выражения основаны на совершенно разных свойствах ядра K для разных типов исходного канала. Влияние этих различий изучается в § 5.

Двоичный симметричный канал. Рассмотрим случай, когда $\mathbf{W}_0^{(0)}$ – двоичный симметричный канал с переходной вероятностью p. В [46, 47] было показано, что в этом случае

$$I(Y_0^{l-1}; U_i^{l-1} \,|\, U_0^{i-1} = 0) = l - i + H(R_i) - nh(p),$$

где $h(p)$ – функция двоичной энтропии, а $H(R_i)$ – энтропия случайной величины, соответствующей строке из стандартной таблицы Слепяна, содержащей выходной вектор канала. Ее можно вычислить следующим образом:

$$H(R_i) = -\sum_{j=1}^{2^i} P_{ij} \log (P_{ij}),$$

где

$$P_{ij} = \sum_{s=0}^{l} w_i(j,s) p^s (1-p)^{l-s},$$

а $w_i(j,s)$ – число векторов веса s в j-й строке таблицы стандартной расстановки (т.е. в j-м смежном классе) для кода $\mathcal{C}_i$. Отсюда получаем

$$\mathbb{I}_i = 1 + H(R_i) - H(R_{i+1}).$$

Это позволяет вычислить пропускные способности подканалов поляризующего преобразования с одним уровнем поляризации. К сожалению, полученные подканалы не являются двоичными симметричными, поэтому этот подход невозможно обобщить на многоуровневое поляризующее преобразование с $m > 1$. Вычисление полного распределения веса всех смежных классов линейного кода также представляет собой достаточно интересную задачу.

Этот подход можно было бы применить ко всей матрице A, если найти способ вычисления соответствующих значений $w_i(j,s)$ с разумной сложностью.

Двоичный стирающий канал. Если $\mathbf{W}$ – двоичный стирающий канал с вероятностью стирания $Z_{0,0}$, то все подканалы $\mathbf{W}_p^{(i)}$ также являются таковыми [22]. Следовательно, их пропускные способности можно вычислить как $I_{p,i} = 1 - Z_{p,i}$, где $Z_{p,i}$ – соответствующий параметр Бхаттачарьи, т.е. вероятность стирания. Эту величину можно найти как

$$Z_{p,li+j} = \sum_{s=1}^{l} B_s^{(j)} Z_{p-1,i}^s (1 - Z_{p-1,i})^{l-s}, \quad 0 < p \leqslant m,$$

где $B_s^{(j)}$ – число конфигураций стирания веса s, в результате которых 0-й информационный символ кода $\mathcal{C}_j$ невозможно восстановить. Эффективные алгоритмы для вычисления этих величин приведены в [22, 35, 36].

АБГШ-канал. Для каналов общего вида точная оценка (12) не представляется возможной. Однако на случай полярных кодов можно распространить полуаналитический метод EXIT-функций (extrinsic information transfer), который с большим успехом используется для построения МПП-кодов и турбокодов [48].

Из (11) получаем

$$\mathbb{I}_i = I(Y_0^{l-1}, U_0^{i-1}; U_i) = 1 + \int\limits_{y_0^{l-1} \in \mathbb{R}^l} \mathbf{W}(y_0^{l-1} \,|\, 0) \log_2 \frac{\sum\limits_{c \in \mathcal{C}_{i+1}} \mathbf{W}(y_0^{l-1} \,|\, c)}{\sum\limits_{c' \in \mathcal{C}_i} \mathbf{W}(y_0^{l-1} \,|\, c')} \, dy_0^{l-1} =$$

$$= 1 - \int\limits_{y_0^{l-1} \in \mathbb{R}^l} \mathbf{W}(y_0^{l-1} \,|\, 0) \log_2 \left(1 + 1/\mathcal{R}_i(y_0^{l-1})\right) dy_0^{l-1}, \tag{13}$$

где

$$\mathcal{R}_i(y_0^{l-1}) = \frac{\sum\limits_{c\in\mathcal{C}_i^{(0)}} \mathbf{W}(y_0^{l-1}\,|\,c)}{\sum\limits_{c'\in\mathcal{C}_i^{(1)}} \mathbf{W}(y_0^{l-1}\,|\,c')} = \frac{\sum\limits_{c\in\mathcal{C}_i^{(0)}} \mathbf{W}(c\,|\,y_0^{l-1})}{\sum\limits_{c'\in\mathcal{C}_i^{(1)}} \mathbf{W}(c'\,|\,y_0^{l-1})} = \frac{\mathbf{W}_1^{(i)}(\mathbf{0},0\,|\,y_0^{l-1})}{\boldsymbol{W}_1^{(i)}(\mathbf{0},1\,|\,y_0^{l-1})}$$

– ЛОПП для i-го символа, $\mathcal{C}_i^{(0)} = \mathcal{C}_{i+1}$, $\mathcal{C}_i^{(1)} = \mathcal{C}_i \setminus \mathcal{C}_{i+1}$. Вместо вычисления интеграла в (13) по $y_0^{l-1} \in \mathbb{R}^l$ можно вычислять

$$\mathbb{I}_i = 1 - \int\limits_{-\infty}^{\infty} p_i(\xi\,|\,0)\log_2(1+e^{-\xi})\,d\xi, \tag{14}$$

где $p_i(\xi\,|\,0)$ – плотность вероятности ЛОПП $\xi = \log\mathcal{R}_i(Y_0^{l-1})$, где величины Y_j подчиняются распределению $\mathbf{W}(y\,|\,0)$. Такой интеграл можно вычислить методом Монте-Карло, т.е. подавая случайные векторы Y_0^{l-1} на декодер, вычисляющий $\log\mathcal{R}_i(Y_0^{l-1})$, и усредняя соответствующие значения $\log_2(1+e^{-\xi})$ [48].

Однако точное вычисление величин $\log\mathcal{R}_i(Y_0^{l-1})$ может оказаться слишком сложным для практической реализации. Оказывается, что подстановка этих значений в приближенные ЛОПП, задаваемые формулой (6), приводит к неверным результатам (например, $\mathbb{I}_i < 0$). Поэтому мы предлагаем переписать (13) в виде

$$\begin{aligned}
\mathbb{I}_i &= 1 - \int\limits_{\mathbb{R}^l} \mathbf{W}(y_0^{l-1}\,|\,0)\log_2\left(1+\frac{\mathbf{W}_1^{(i)}(\mathbf{0},1\,|\,y_0^{l-1})}{\boldsymbol{W}_1^{(i)}(\mathbf{0},0\,|\,y_0^{l-1})}\right)dy_0^{l-1} \approx \\
&\approx 1 - \int\limits_{-\infty}^{\infty} f_i(\psi\,|\,0)\log_2\left(1+\frac{P\{u_i=1\,|\,\psi\}}{P\{u_i=0\,|\,\psi\}}\right)d\psi = \\
&= 1 - \int\limits_{-\infty}^{\infty} f_i(\psi\,|\,0)\log_2\left(1+\frac{f_i(-\psi\,|\,0)}{f_i(\psi\,|\,0)}\right)d\psi,
\end{aligned} \tag{15}$$

где последнее равенство справедливо для симметричного канала, $f_i(\psi\,|\,0)$ – плотность вероятности ЛОПП $\mathcal{L}_1^{(i)}(\mathbf{0}\,|\,y_0^{l-1})$, заданная (6), а $P\{u_i = c\,|\,\psi\}$ – вероятность того, что $u_i = c$ при условии $\mathcal{L}_1^{(i)}(\mathbf{0}\,|\,y_0^{l-1}) = \psi$. Следовательно, можно оценить $\mathbb{I}_i$, построив гистограмму [48] для $f_i(\psi\,|\,0)$ по выходу алгоритма, вычисляющего (6).

Аналогично можно вычислить параметр Бхаттачарьи для $\mathbf{W}_1^{(i)}$ как [49, формула (4.62)]

$$\mathbb{Z}_i = \int\limits_{-\infty}^{\infty} g_i(\psi)e^{-\psi/2}\,dx, \tag{16}$$

где $g_i(\psi\,|\,0)$ – плотность вероятности для $\mathbf{L}_1^{(i)}(0\,|\,y_0^{l-1})$. Ее можно аппроксимировать как $g_i(\psi\,|\,0) \approx f_i(\psi\,|\,0)$, так что ее можно получить из той же гистограммы для $f_i(\psi\,|\,0)$, что и $\mathbb{I}_i$.

3.2. Гауссовская аппроксимация.

Общий подход. Для построения полярного кода по некоторым ядрам $\mathcal{K}_p$, $1 \leqslant p \leqslant m$, мы предлагаем считать, что все подканалы $\mathbf{W}_p^{(i)}$, $0 \leqslant p \leqslant m$, $0 \leqslant i < L_p$, явля-

ются гауссовскими, так что их можно полностью охарактеризовать соответствующей взаимной информацией. Мы предлагаем построить таблицы значений пропускной способности $\mathbb{I}_{p,i}(C)$ битовых подканалов $\mathbf{W}_{1,\mathcal{K}_p}^{(i)}$, индуцированных каждым ядром $\mathcal{K}_p$, для некоторого конечного набора параметров исходного АБГШ-канала $\mathbf{W}$, где C – пропускная способность этого канала $\mathbf{W}$. Эти таблицы можно использовать для интерполяции значений $\mathbb{I}_{p,i}(C)$ для любого значения $C \in [0,1]$.

Для построения полярного (l^m, k)-кода для АБГШ-канала с пропускной способностью C мы предлагаем рекуррентно вычислять пропускную способность $I_{p,t}(C)$ битового подканала $\mathbf{W}_p^{(t)}$ как

$$I_{p,l_p i+j}(C) \approx \mathbb{I}_{p,j}(I_{p-1,i}(C)), \quad 0 \leqslant j < l_p, \quad p \geqslant 1, \tag{17}$$

где $I_{0,0}(C) = C$, и объявить замороженными символы u_i, где i – номера подканалов с наименьшими значениями $I_{m,i}(C)$, $0 \leqslant i < n$.

Альтернативным вариантом является построение аналогичных интерполяционных таблиц $\mathbb{Z}_{p,i}(Z)$ для параметра Бхаттачарьи $\mathbb{Z}_i$ битовых подканалов $\mathbf{W}_{1,\mathcal{K}_p}^{(i)}$, индуцированных каждым ядром $\mathcal{K}_p$, где Z – параметр Бхаттачарьи исходного канала. Тогда параметры Бхаттачарьи для битовых подканалов, индуцированных матрицей A, можно оценить как

$$Z_{p,l_p i+j}(Z) \approx \mathbb{Z}_{p,j}(Z_{p-1,i}(Z)), \quad 0 \leqslant j < l_p, \quad p \geqslant 1. \tag{18}$$

Для построения полярного кода мы предлагаем объявить замороженными символы u_i, где i – номера подканалов с наибольшими значениями $Z_{m,i}(C)$.

Хвосты функций надежности. К сожалению, интерполяционные таблицы для $\mathbb{I}_{p,j}(C)$ и $\mathbb{Z}_{p,j}(Z)$, полученные методом Монте-Карло, оказываются очень неточными для значений C и Z, близких к 0 и 1. Это может привести к довольно плохим конструкциям длинных полярных кодов. Здесь мы предложим инструменты, которые помогут улучшить точность этих функций вблизи их граничных точек.

Лемма 1. *Если* $\mathbf{W}$ *– АБГШ-канал с двоичным входом, то* $\mathbb{I}_i(C)$, *где* $C \in [0,1]$ *– пропускная способность канала* $\mathbf{W}$, *является непрерывной бесконечно дифференцируемой функцией.*

Из леммы 1 следует, что существует разложение в ряд Тейлора для $\mathbb{I}_{p,i}(C)$. Так как $\mathbb{I}_{p,i}(0) = 0$, то $\mathbb{I}_{p,i}(C) = \alpha_{p,i} C^{\delta_{p,i}} + o(C^{\delta_{p,i}})$ для некоторых $\alpha_{p,i} \in \mathbb{R}$, $\delta_{p,i} \in \mathbb{N}$. Поэтому, чтобы не использовать зашумленные оценки Монте-Карло в окрестности $C = 0$, мы предлагаем использовать аппроксимацию

$$\mathbb{I}'_{p,i}(C) \approx \begin{cases} \mathbb{I}_{p,i}(C), & C > C_0, \\ \alpha_{p,i} C^{\delta_{p,i}}, & C \leqslant C_0, \end{cases} \tag{19}$$

где пороговое значение C_0 можно выбирать в зависимости от точности оценки по методу Монте-Карло для $\mathbb{I}_{p,i}(C)$. Параметры $\alpha_{p,i}$, $\delta_{p,i}$ можно найти численным методом подгонки кривых, используя значения $\mathbb{I}_{p,i}(C)$, полученные методом Монте-Карло для достаточно больших C. Точность подгонки кривых можно улучшить, если разрешить использовать вещественные значения $\delta_{p,i}$.

В [12, формула (14)] было показано, что параметры Бхаттачарьи битовых подканалов, индуцированных ядром, удовлетворяют неравенствам

$$Z^{D_{pj}} \leqslant \mathbb{Z}_{p,j} \leqslant 2^{l-i} Z^{D_{pj}},$$

т.е. $\mathbb{Z}_{p,i} = \Theta(Z^{D_{p,i}})$, где $D_{p,i}$ – частичные расстояния для ядра $\mathcal{K}_p$. Поэтому мы предлагаем использовать аппроксимацию

$$\mathbb{Z}'_{p,i}(Z) \approx \begin{cases} \mathbb{Z}_{p,i}(Z), & Z > Z_{p0}, \\ \beta_{p,i} Z^{D_{p,i}}, & Z \leqslant Z_{p0}, \end{cases} \tag{20}$$

где $\beta_{p,i}$ можно найти путем численной подгонки кривых, а пороговое значение Z_0 можно выбирать в зависимости от точности оценки по Монте-Карло для $\mathbb{Z}_{p,i}(Z)$.

Гибридный подход. Обе формулы (17), (18) приводят к довольно неточным значениям для подканалов $\mathbf{W}_m^{(i)}$, если некоторые промежуточные значения в этих рекурсиях становятся близкими к 1, где не применимы уточненные оценки (19)–(20). Это связано как с неточностью гауссовой аппроксимации, так и с ошибками интерполяции. Однако из $C \approx 1$ следует $Z \approx 0$ и наоборот. Следовательно, можно надеяться, что хотя бы одно из этих выражений может привести к достаточно точным результатам.

Поэтому мы предлагаем для получения уточненных оценок объединить границы (8) и (9). А именно, мы предлагаем использовать среднее значение этих верхних и нижних границ и оценок, данных формулами (19)–(20). Кроме того, необходимо убедиться, что полученное значение не меньше соответствующей нижней границы. Более конкретно, мы предлагаем вычислить

$$I_{p,l_p i+j}(C, Z) = \begin{cases} \iota_{ij}, & \iota_{ij} \leqslant \varepsilon, \\ \max\left(\widetilde{\mathcal{I}}(\zeta_{ij}), \dfrac{\widetilde{\mathcal{I}}(\zeta_{ij}) + \iota_{ij} + \widehat{\mathcal{I}}(\zeta_{ij})}{3}\right), & \iota_{ij} > \varepsilon, \end{cases} \tag{21}$$

и

$$Z_{p,l_p i+j}(C, Z) = \begin{cases} \zeta_{ij}, & \zeta_{ij} \leqslant \delta, \\ \max\left(\widetilde{\mathcal{Z}}(\iota_{ij}), \dfrac{\widetilde{\mathcal{Z}}(\iota_{ij}) + \zeta_{ij} + \widehat{\mathcal{Z}}(\iota_{ij})}{3}\right), & \zeta_{ij} > \delta, \end{cases} \tag{22}$$

где

$$\iota_{ij} = \mathbb{I}'_{p,j}(I_{p-1,i}(C, Z)), \quad \zeta_{ij} = \mathbb{Z}'_{p,j}(Z_{p-1,i}(C, Z)),$$

а Z и C – параметр Бхаттачарьи и пропускная способность исходного канала соответственно. Эмпирическим путем было установлено, что пороговые значения $\varepsilon = 0{,}9997$ и $\delta = 0{,}7$ обеспечивают достаточно хорошую точность.

Таким образом, предлагаемый метод построения полярного кода включает в себя следующие шаги:

1. Предварительные вычисления: с помощью метода Монте-Карло построить гистограммы ЛОПП плотностей распределения $f_i(\psi \,|\, 0)$ и построить таблицы значений $\mathbb{I}_{p,i}(C)$ и $\mathbb{Z}_{p,i}(Z)$ по формулам (15)–(16) для битовых подканалов $\mathbf{W}_{1,\mathcal{K}_p}^{(i)}$, $0 \leqslant i < l_p$, индуцированных каждым ядром $\mathcal{K}_p$. Здесь C и Z – пропускная способность и параметры Бхаттачарьи АБГШ-канала с двоичным входом с дисперсией шума σ^2. Кроме того, вычислить значения $\alpha_{p,j}$, $\beta_{p,j}$, $D_{p,j}$, $Z_{p,0}$, $0 \leqslant j < l_p$.
2. Для построения полярного кода с поляризационной матрицей $\mathcal{K}_1 \otimes \mathcal{K}_2 \otimes \ldots \otimes \mathcal{K}_m$ для АБГШ-канала с двоичным входом с пропускной способностью C и параметром Бхаттачарьи Z, предлагается вычислять $I_{m,i}(C, Z)$, $0 \leqslant i < \prod\limits_{j=1}^{m} l_j$, по формулам (21)–(22), используя $I_{0,0}(C, Z) = C$ и $Z_{0,0}(C, Z) = Z$ в качестве начальной точки рекурсии. В качестве замороженного множества $\mathcal{F}$ предлагается взять множество индексов i подканалов с наименьшими значениями $I_{m,i}(C, Z)$.

Сложность шага предварительного вычисления для каждого из ядер составляет $O(TPK)$, где T – количество итераций Монте-Карло на одну точку σ^2, P – количество различных значений σ^2, а K – общая сложность обработки ядра. Следует отметить, что этап предварительных вычислений нужно выполнить только один раз для каждого ядра, а его результаты можно в дальнейшем использовать для построения большого количества различных кодов.

Шаг построения полярного кода для предлагаемого метода требует

$$\sum_{i=1}^{m} l^i = \frac{l^{m+1}-1}{l-1} = O(n)$$

операций вычисления по формулам (21)–(22). Это намного меньше по сравнению со сложностью $O(Tn\log n)$ построения полярного кода методом Монте-Карло, где T – количество итераций в методе Монте-Карло.

Этот подход можно непосредственно применять для выбора статических и динамических замороженных символов типа B при построении полярных подкодов, описанных в п. 2.5.

§ 4. Поиск наилучшего поляризующего преобразования со смешанными ядрами

Напомним, что поляризующее преобразование со смешанными ядрами задается матрицей

$$A = \mathcal{K}_1 \otimes \mathcal{K}_2 \otimes \ldots \otimes \mathcal{K}_m,$$

где $\mathcal{K}_i$ – ядро поляризации размера $l_i \times l_i$, и таким образом, общая длина A равна $n = \prod_{i=1}^{m} l_i$. Возможные размеры l_i ядер поляризации в A зависят от разложения на множители целого числа n. Заметим, что некоторые n могут допускать много конфигураций размеров ядер, например, для $n = 2400$ имеем $2400 = 32 \cdot 15 \cdot 5 = 30 \cdot 16 \cdot 5 = 24 \cdot 20 \cdot 5 = 32 \cdot 25 \cdot 3$ и т.д.

В недавних работах [16, 50, 51] были предложены различные конструкции ядер разной длины. Таким образом, для фиксированной длины n существует множество способов построения матрицы A. Более того, в общем случае произведение Кронекера некоммутативно, то есть для двух квадратных матриц $\mathbf{A}$ и $\mathbf{B}$

$$\mathbf{A} \otimes \mathbf{B} = \mathbf{P}(\mathbf{B} \otimes \mathbf{A})\mathbf{P}^\top, \tag{23}$$

где $\mathbf{P}$ – некоторая перестановочная матрица. Другими словами, для ядер поляризации $\mathcal{K}_1$ и $\mathcal{K}_2$ поляризующие преобразования $\mathcal{K}_1 \otimes \mathcal{K}_2$ и $\mathcal{K}_2 \otimes \mathcal{K}_1$ могут привести к полярным кодам с различной корректирующей способностью при ПИ-декодировании.

Иными словами, если задан набор ядер размера l_i из разложения числа n на множители, то возникает вопрос, как скомбинировать их, чтобы получить матрицу A, приводящую к наименьшей вероятности ошибки при ПИ-декодировании. Для удобства обозначим

$$\mathbf{W}_{m,A}^{(i)} = \mathbf{W}_{m,A}^{(i)}(y_0^{n-1}, u_0^{i-1} \,|\, u_i).$$

Пусть $P_e(\mathbf{W}_{m,A}^{(i)})$ – вероятность ошибки в канале $\mathbf{W}_{m,A}^{(i)}$. Пусть $\mathcal{A}_k$ – множество индексов информационных битов, т.е. $\mathcal{A}_k = [n] \setminus \mathcal{F}$. Для некоторого канала W обозначим через $P_e(W, A, \mathcal{A}_k)$ вероятность ошибки на кодовое слово при ПИ-декодирова-

нии. Из доказательства [2, утверждение 2] следует, что

$$P_e(W, A, \mathcal{A}_k) \leqslant \sum_{i \in \mathcal{A}_k} P_e(\mathbf{W}_{m,A}^{(i)}). \tag{24}$$

Напомним, что $\mathbf{W}_{m,A}^{(i)}$ также является дискретным каналом без памяти с двоичным входом, поэтому имеем [2]

$$P_e(\mathbf{W}_{m,A}^{(i)}) \leqslant Z(\mathbf{W}_{m,A}^{(i)}), \tag{25}$$

и [52, теорема 2.3]

$$h^{-1}(1 - I(\mathbf{W}_{m,A}^{(i)})) \leqslant P_e(\mathbf{W}_{m,A}^{(i)}) \leqslant \frac{1}{2}(1 - I(\mathbf{W}_{m,A}^{(i)})). \tag{26}$$

Поскольку вычисление $h^{-1}(p)$ также численно неустойчиво, рассмотрим следующие оценки, которые вытекают из (24) и (25)–(26):

$$P_e(W, A, \mathcal{A}_k) \leqslant \sum_{i \in \mathcal{A}_k} Z(\mathbf{W}_{m,A}^{(i)}), \tag{27}$$

$$P_e(W, A, \mathcal{A}_k) \leqslant \frac{1}{2} \sum_{i \in \mathcal{A}_k} (1 - I(\mathbf{W}_{m,A}^{(i)})). \tag{28}$$

Предположим, что имеется множество $\mathbb{K}$ ядер поляризации, такое что для каждого $\mathcal{K} \in \mathbb{K}$ его размер $l(\mathcal{K})$ делится на n. Тогда определим множество $\mathcal{G}(\mathbb{K}, n)$ всех подмножеств $\{\mathcal{K}^{(1)}, \mathcal{K}^{(2)}, \ldots, \mathcal{K}^{(\overline{m})}\} \subseteq \mathbb{K}$, где $\prod_{i=1}^{\overline{m}} l(\mathcal{K}_i) = n$. Заметим, что $\overline{m}$ могут быть различными.

Итак, мы предлагаем следующий метод поиска наилучшего поляризующего преобразования со смешанными ядрами A размера n с наилучшим значением вероятности $P_e(W, A, \mathcal{A}_k)$:

1. Составим множество $\mathbb{K}$ ядер, для которых $|\mathcal{G}(\mathbb{K}, n)| > 0$;
2. Для каждого ядра $\mathcal{K} \in \mathbb{K}$ вычислим $\mathbb{I}_i$ и $\mathbb{Z}_i$ по формулам (15)–(16), $i \in [l(\mathcal{K})]$, как описано в п. 3.1;
3. Для каждого $\{\mathcal{K}^{(1)}, \mathcal{K}^{(2)}, \ldots, \mathcal{K}^{(\overline{m})}\} \in \mathcal{G}(\mathbb{K}, n)$ и каждой перестановки σ из $\overline{m}$ элементов:
   (a) Положим $A = \mathcal{K}_{\sigma(1)} \otimes \mathcal{K}_{\sigma(2)} \otimes \ldots \otimes \mathcal{K}_{\sigma(\overline{m})}$;
   (b) Вычислим $I(\mathbf{W}_{m,A}^{(i)}) = I_{m,i}(C, Z)$ с помощью предложенного гибридного метода (21), используя $\mathbb{I}_i$ и $\mathbb{Z}_i$, полученные предварительными вычислениями на шаге 2;
   (c) Вычислим аппроксимацию $\bar{P}_e$ верхней границы (28) на $P_e(W, A, \mathcal{A}_k)$, где $\mathcal{A}_k \in \in [n]$ – множество индексов k с наилучшими значениями $I_{m,i}(C, Z)$;
   (d) Сохраним v наилучших преобразований A, приводящих к наименьшей верхней границе $\bar{P}_e$ на вероятность ошибки;
4. Пусть $\mathbb{A}$ – набор наилучших преобразований, полученных на предыдущем шаге. Построим коды v с преобразованиями $A \in \mathbb{A}$ и найдем $P_e(W, A, \mathcal{A}_k)$ методом имитационного моделирования;
5. Выберем преобразование A с наилучшим значением $P_e(W, A, \mathcal{A}_k)$.

Согласно проведенным нами экспериментам мы полагаем, что достаточно использовать параметр $v = 8 \ldots 10$.

Заметим, что полный перебор по всем перестановкам на шаге 3 вряд ли проблематичен с точки зрения сложности. Действительно, в [12] было показано, что

Таблица 1

Параметры интерполяции пропускной способности и частичные расстояния для ядра K_2

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$D_{2,i}$	1	2	2	4	2	2	4	4	6	6	8	8	4	8	8	16
$\delta_{2,i}$	13,3	7,2	6,8	3,6	6,7	6,4	5,2	4,9	3,5	3,5	1,9	1,9	3,1	1,8	1,8	0,96

не существует ядер размера меньше 15 со скоростью поляризации больше $1/2$, т.е. больше скорости поляризации ядра Арикана. Из этого следует, что для кодов практически приемлемой длины, использующих ядра с высокой скоростью поляризации, число ядер $\overline{m}$ должно быть достаточно малым.

Наши эксперименты показывают, что аппроксимация верхней границы (28) на основе пропускной способности битовых подканалов, более точна, чем аппроксимация верхней границы (27) на основе параметра Бхаттачарьи. Поэтому для сравнения различных поляризующих преобразований и выбора хороших кодов мы используем формулу (28).

§ 5. Численные результаты

5.1. Длинные полярные коды с большими ядрами. В этом пункте представлены результаты моделирования для случая ядер K_1 и K_2 размера 16×16, введенных в [53], а также ядра K_3 размера 32, полученного по методу, приведенному в [50]. Эти ядра имеют скорости поляризации $E(K_1) = E(K_2) = 0{,}51$, $E(K_3) = 0{,}522$ и экспоненты масштабирования двоичного стирающего канала $\mu(K_1) = 3{,}346$, $\mu(K_2) = 3{,}45$ и $\mu(K_3) = 3{,}417$ соответственно.

На рис. 1 показаны функции пропускной способности битовых подканалов для АБГШ-канала (сплошные линии), двоичного симметричного канала (пунктирные линии) и двоичного стирающего канала (штрих-пунктирные линии). Кривые для АБГШ-канала были получены методом Монте-Карло, а кривые для двоичного симметричного и двоичного стирающего каналов были вычислены в точности как описано в п. 3.1. Одни и те же кривые построены в линейном и логарифмическом масштабах. Видно, что погрешности метода Монте-Карло приводят к некоторому шуму в области низкой пропускной способности для кривых, полученных для АБГШ-канала.

Также видно, что у кривых для разных подканалов есть точки пересечения. В зависимости от типа исходного канала эти точки оказываются при различных значениях его пропускной способности C. Таким образом, функции пропускной способности для двоичного стирающего канала и двоичного симметричного канала нельзя надежно использовать для АБГШ-канала.

На рис. 2 показаны функции параметров Бхаттачарьи $\mathbb{Z}_{1,j}$ для ядра K_2 в случае АБГШ-канала. В табл. 1 приведены значения параметров $\delta_{2,i}$ и $D_{2,i}$ для рассматриваемого ядра K_2.

Необходимо следить за тем, чтобы при построении полярных кодов не использовались зашумленные участки интерполяционных кривых пропускной способности и параметра Бхаттачарьи. Чтобы избежать этого, необходимо тщательно выбирать пороговые значения C_0 и Z_0, при которых происходит переход к аналитическому методу. Исходя из этого соображения, мы предлагаем задаться значениями $C_0 = 10^{-3}$ в (19) и $Z_0 = 10^{-4}$ в (20). Для $C \geqslant C_0$ и $Z \geqslant Z_0$ для вычисления $\mathbb{I}_{p,j}(C)$ и $\mathbb{Z}_{p,j}(Z)$ мы используем линейную интерполяцию между сохраненными значениями.

На рис. 3 показана корректирующая способность полярных кодов длины $1/2$ на основе поляризующих преобразований $A = K_1^{\otimes 4}$, $A = K_2^{\otimes 4}$ и полярного кода длины 32768 на основе $A = K_3^{\otimes 3}$ при ПИ-декодировании. Эти коды были постро-

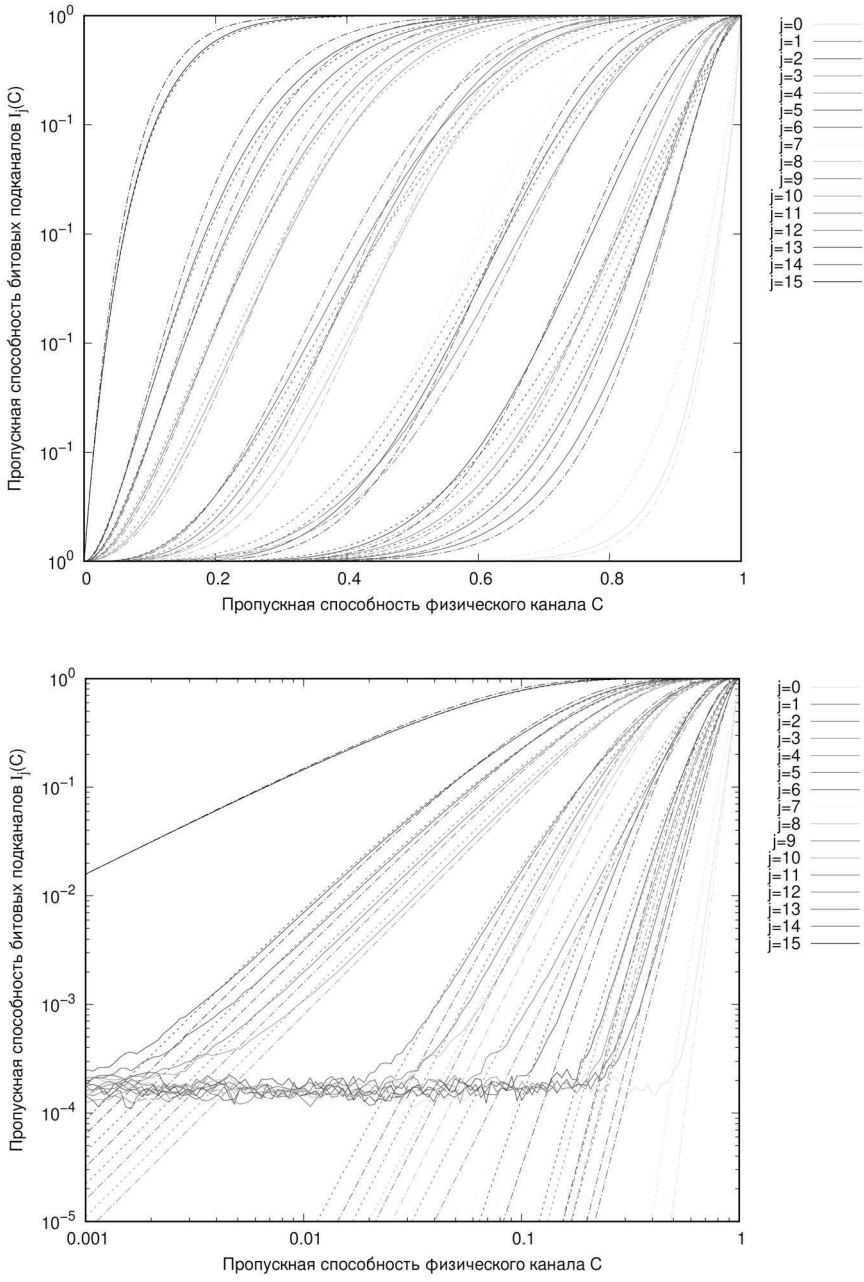


Рис. 1. Функции пропускной способности подканалов для K_2

ены для АБГШ-канала с использованием предложенного подхода, основанного на аппроксимации пропускной способности (17), аппроксимации параметра Бхаттачарьи (18), гибридного подхода к аппроксимации пропускной способности (21)–(22), а также метода Монте-Карло. Видно, что подходы, основанные на аппроксимации только пропускной способности и параметра Бхаттачарьи, могут привести к существенному снижению эффективности по сравнению с конструкцией, основанной на

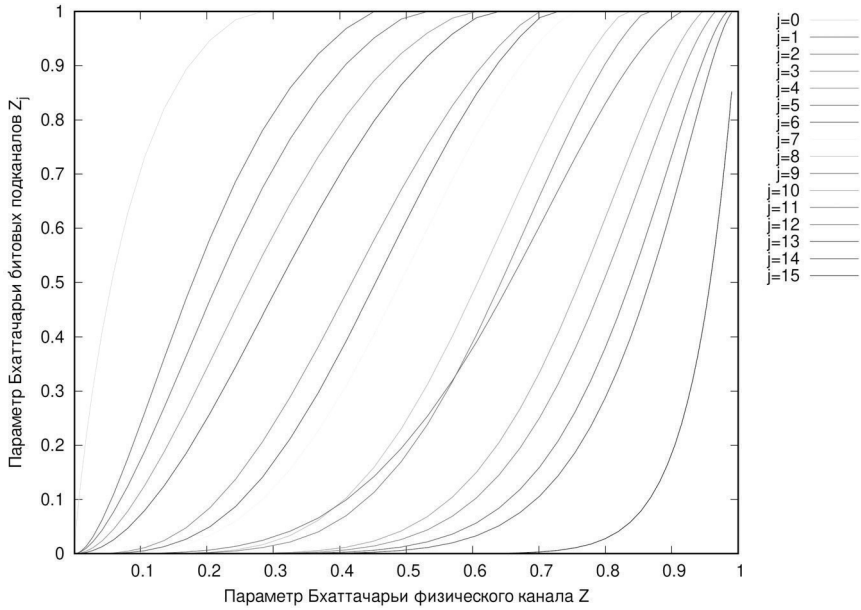


Рис. 2. Параметр Бхаттачарьи АБГШ-канала для K_2

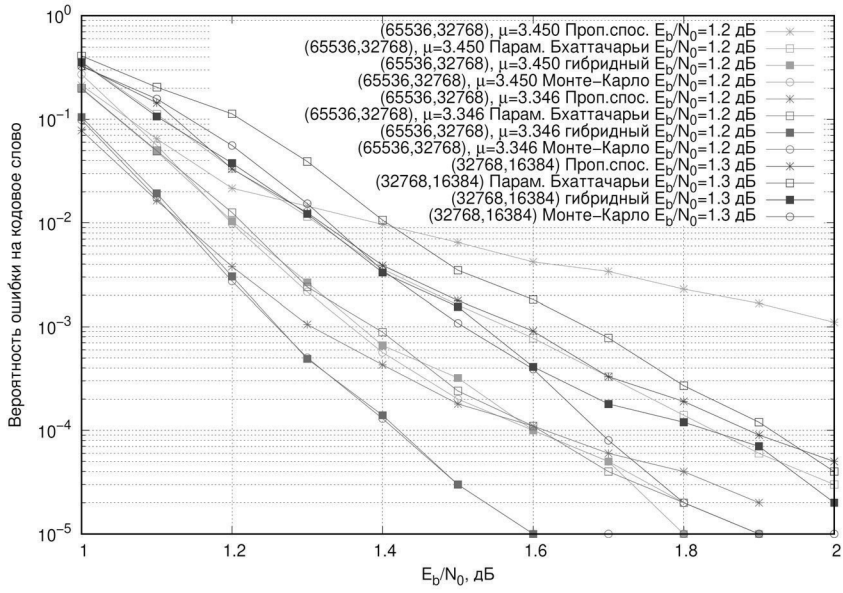


Рис. 3. Корректирующая способность длинных полярных кодов при ПИ-декодировании

методе Монте-Карло. Однако гибридный подход приводит к кодам с почти такой же корректирующей способностью, что и конструкция на основе метода Монте-Карло. Заметим также, что наилучшей корректирующей способностью обладает код, построенный с использованием ядра с наименьшей экспонентой масштабирования.

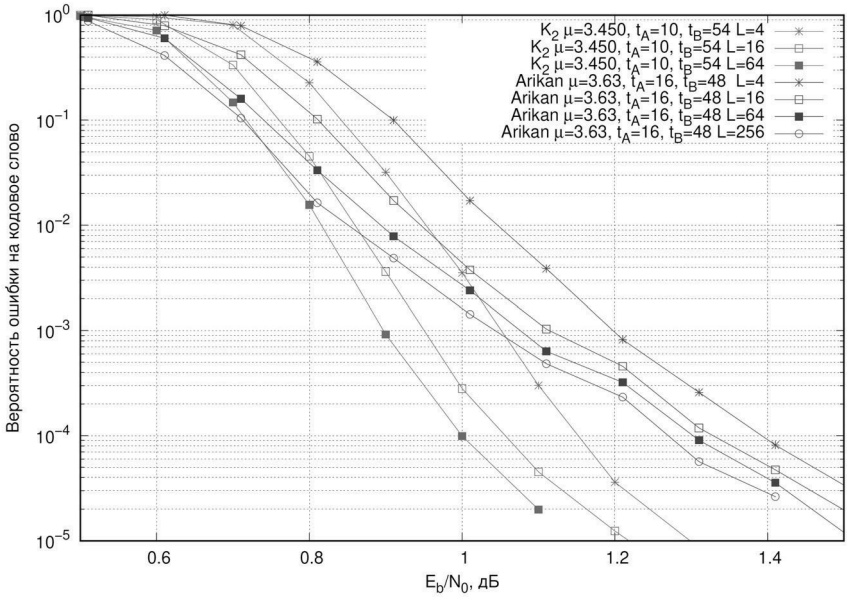


Рис. 4. Корректирующая способность полярных $(65536, 32768)$-подкодов

Полярные коды с большими ядрами все еще обладают довольно низким минимальным расстоянием, что приводит к невысокой корректирующей способности. Использование рандомизированных полярных подкодов [1] позволяет получить коды с существенно лучшим распределением весов. На рис. 4 показана корректирующая способность рандомизированных полярных подкодов со скоростью $1/2$ на ядре Арикана и ядре K_2 с t_A динамически замороженными символами типа A и t_B динамически замороженными символами типа B. Декодирование производилось с помощью последовательного алгоритма [9], который, как известно, обеспечивает почти такую же эффективность, что и списочный ПИ-декодер [6] с тем же размером списка L. Видно, что код, основанный на ядре размера 16×16, характеризуется гораздо более крутой кривой вероятности ошибки на кодовое слово и требует значительно меньшего размера списка для достижения той же вероятности ошибки, что и код, основанный на ядре Арикана.

5.2. Сравнение полярных кодов с различными поляризующими преобразованиями. Для исследования точности аппроксимации верхней границы (28) было проведено моделирование различных преобразований A длины 2400, состоящих из следующих ядер:

- матрицы Арикана $F_t = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}^{\otimes t}$ размера $2^t \times 2^t$ для $t = 2, \ldots, 5$;
- укороченные матрицы Арикана $s_l(F_t)$ размеров $l = 3, 5, 15, 20, 24, 25, 30$ с конфигурациями укорочения, взятыми из [51];
- ядро K_{32} размера 32×32 из [16] и укороченные ядра $s_l(K_{32})$ для $l = 20, 24, 25, 30$;
- ядра K_{16}, K'_{16} размера 16×16 из [16] и $s_{15}(K_{16})$;
- K_{24}, K^*_{24} и K_{20} из [50].

Прежде всего, мы сравниваем границы, основанные на приближенных значениях параметра Бхаттачарьи (27) и симметричных пропускных способностей (28). Это сравнение показано на рис. 5. Видно, что оценки пропускной способности, полученные из аппроксимации (17) и (19), недостаточно точны, чтобы правильно оценить вероятность ошибки на кодовое слово при ПИ-декодировании для значений

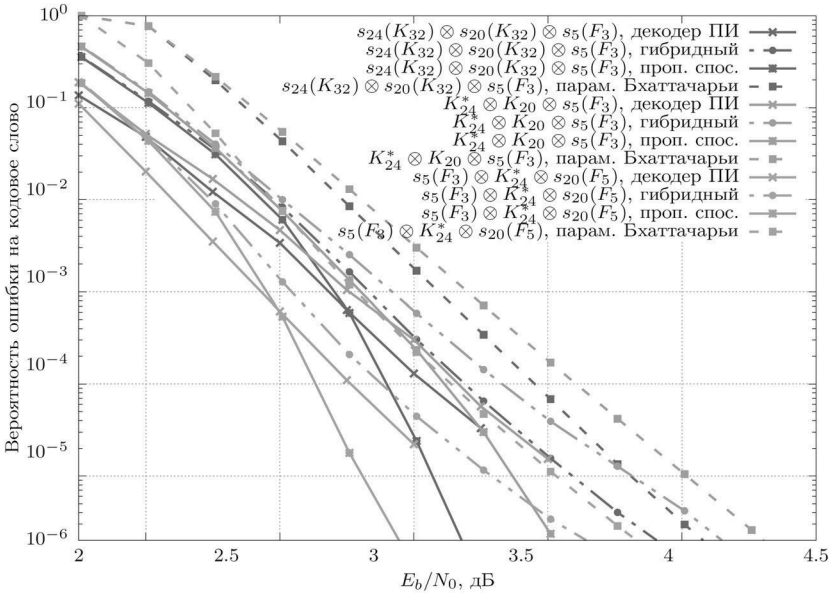


Рис. 5. Сравнение различных границ на вероятность ошибки при ПИ-декодировании

ниже 10^{-3}, а граница (27) слабее границы (28), рассчитанной с помощью предложенного гибридного подхода (см. п. 3.2).

Для исследования применимости верхней границы (28), вычисляемой по приближенным пропускным способностям битовых подканалов, полученным с помощью предложенного гибридного подхода, были построены всевозможные поляризующие преобразования A длины $n = 2400$ из вышеперечисленных ядер. В результате было получено 1716 различных преобразований, для каждого из которых был построен полярный $(2400, 1200)$-код со смешанными ядрами с отношением сигнал-шум на информационный символ $E_b/N_0 = 2{,}5$ дБ, оценена верхняя граница и смоделирована вероятность ошибки ПИ-декодирования для $E_b/N_0 = 2{,}5$ дБ. На рис. 6 представлена полученная приближенная верхняя граница как функция вероятности ошибки ПИ-декодирования. Видно, что разброс точек довольно мал, и нет ни одного случая, когда полученная моделированием вероятность ошибки на кодовое слово превышала бы приближенную верхнюю границу.

На рис. 7 приведены результаты моделирования вероятности ошибки ПИ-декодирования для полярных $(2400, 1200)$-кодов со смешанными ядрами поляризующего преобразования, заданными всеми перестановками ядер K_{24}^*, $s_{20}(F_5)$ и $s_5(F_3)$, построенных для $E_b/N_0 = 2{,}5$ дБ. Можно заметить, что один и тот же набор ядер может приводить к кодам с существенно различной корректирующая способностью. Например, для данного конкретного случая наблюдается разница в 0,4 дБ между лучшим и худшим преобразованиями. Более того, для данного графика, если упорядочить поляризующие преобразования по величине полученной моделированием вероятности ошибки и по верхней границе (28) при гибридном подходе, то оба порядка будут практически одинаковыми. Например, имеется небольшой промежуток между преобразованиями $K_{24}^* \otimes s_{20}(F_5) \otimes s_5(F_3)$ и $K_{24}^* \otimes s_5(F_3) \otimes s_{20}(F_5)$, который остается практически одинаковым как для смоделированной вероятности ошибки, так и для приближенно вычисленной верхней границы. Таким образом, можно выбирать оптимальный порядок поляризующих ядер, вычисляя только верхнюю границу.

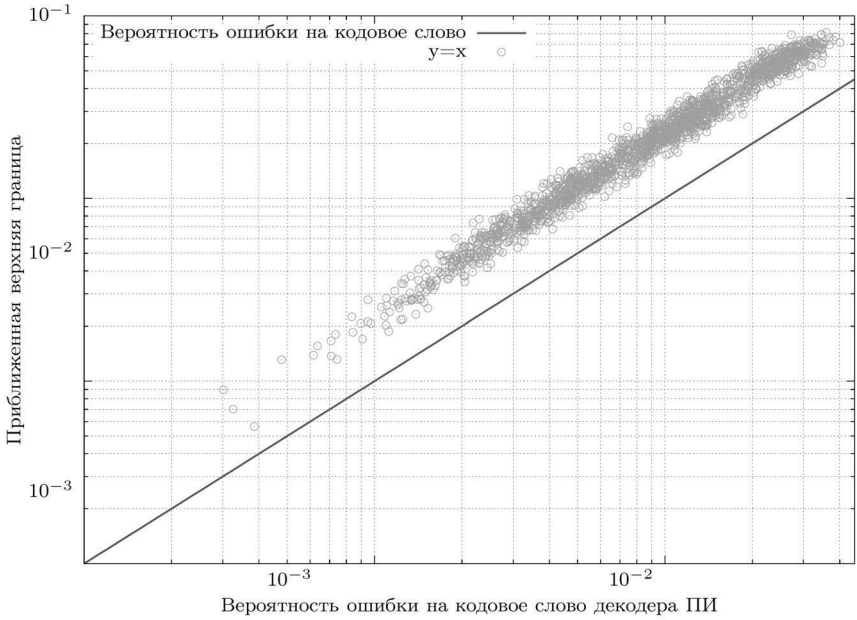


Рис. 6. Приближенная верхняя граница вероятности ошибки на кодовое слово при ПИ-декодировании как функция моделируемой вероятности ошибки на кодовое слово для различных преобразований длины 2400

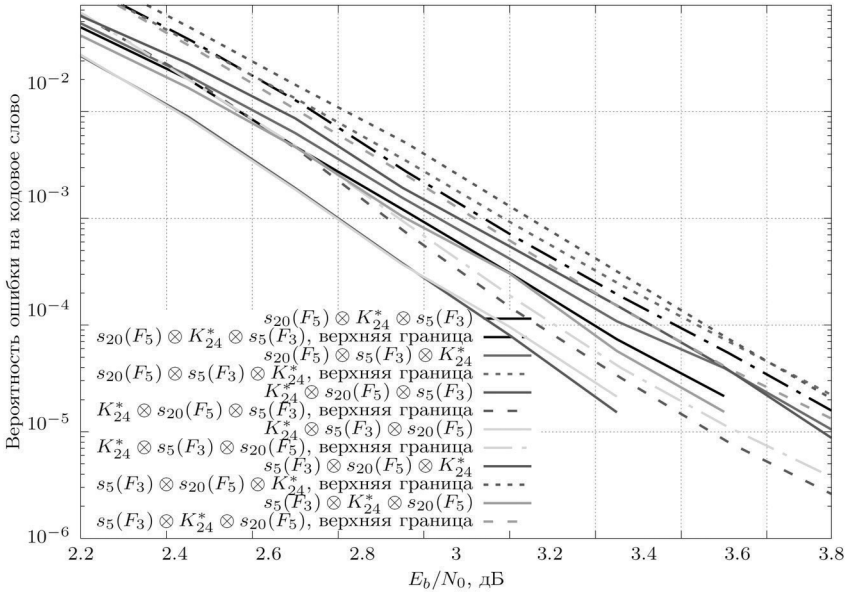


Рис. 7. Точность приближенной верхней границы вероятности ошибки на кодовое слово при ПИ-декодировании для поляризующих преобразований, заданных всеми перестановками ядер K_{24}^*, $s_{20}(F_5)$ и $s_5(F_3)$

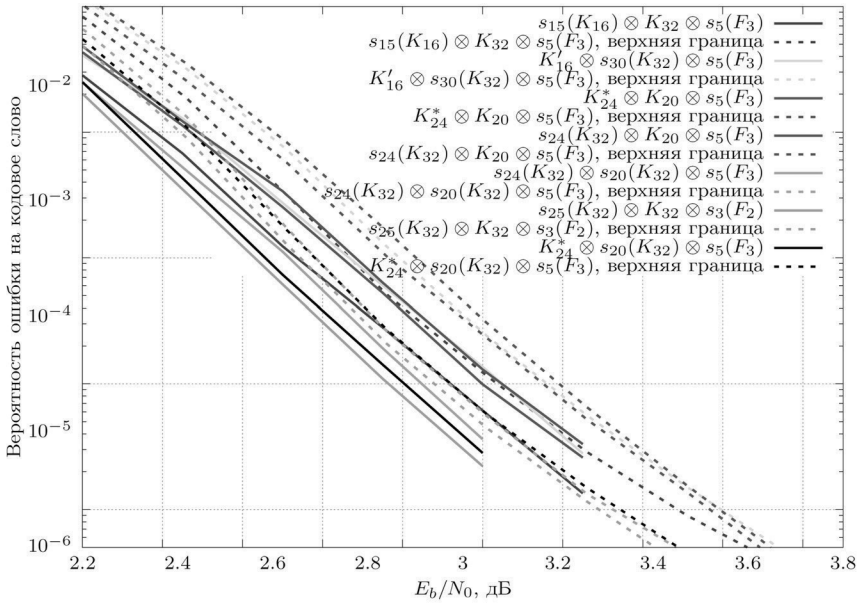


Рис. 8. Точность приближенной верхней границы вероятности ошибки на кодовое слово при ПИ-декодировании для различных поляризующих преобразований

На рис. 8 показаны приближенные верхние границы (28) и полученные моделированием вероятности ошибок для преобразований, заданных различными ядрами поляризации. Видно, что полученные оценки пропускной способности подканалов можно использовать для вычисления достаточно точной верхней границы вероятности ошибки ПИ-декодирования, что в свою очередь позволяет выбирать ядра поляризации, дающие наименьшую вероятность ошибки ПИ-декодирования.

§ 6. Заключение

В статье рассмотрены методы построения полярных кодов с большими ядрами. Изложен простой метод приближенной оценки надежности битовых подканалов, индуцированных поляризующим преобразованием с большими ядрами. Предложенный подход объединяет эмпирически полученные функции пропускной способности и параметра Бхаттачарьи для битовых подканалов в случае единственного ядра, что позволяет получить оценки пропускной способности битовых подканалов большого поляризующего преобразования, включающего несколько ядер. Было показано, что такой подход позволяет получить длинные полярные коды с почти такими же характеристиками, что и в случае построения с использованием метода Монте-Карло. Предложенный подход можно непосредственно использовать для построения полярных кодов с большими ядрами для рэлеевского канала с замираниями.

Точность предложенного подхода зависит от точности кривых интерполяции пропускной способности и параметра Бхаттачарьи, которую можно улучшить, увеличив число итераций в методе Монте-Карло, используемом для получения этих кривых. Необходимо тщательно выбирать параметры C_0 и Z_0, которые используются при переходе к аналитическому методу интерполяции. Использование границ (8) и (9) позволяет оценить точность полученных оценок пропускной способности и параметров Бхаттачарьи для битовых подканалов. Однако следует признать, что предложенный подход основан на предположении, что битовые подканалы, индуцированные ядра-

ми поляризации, ведут себя как АБГШ-каналы с двоичным входом, что на практике может оказаться неверным. Тем не менее, было показано, что предложенный подход дает достаточно точные результаты для кодов длины 65536.

Кроме того, предложен метод выбора последовательности различных ядер поляризации в кодах со смешанными ядрами. Он использует пропускные способности битовых подканалов для вычисления приближенной верхней границы вероятности ошибки ПИ-декодирования, которая, как было показано, оказывается достаточно точной.

Предложенный подход можно также использовать для построения полярных подкодов. Было показано, что полярные подкоды с большим ядром, полученные с помощью предложенного метода, требуют гораздо меньшего размера списка для достижения той же корректирующей способности, что и полярные подкоды с ядром Арикана.

СПИСОК ЛИТЕРАТУРЫ

1. *Trifonov P.* On Construction of Polar Subcodes with Large Kernels // Proc. 2019 IEEE Int. Symp. on Information Theory (ISIT'2019). Paris, France. July 7–12, 2019. P. 1932–1936. https://doi.org/10.1109/ISIT.2019.8849672
2. *Arıkan E.* Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels // IEEE Trans. Inform. Theory. 2009. V. 55. № 7. P. 3051–3073. https://doi.org/10.1109/TIT.2009.2021379
3. *Marshakov E., Balitskiy G., Andreev K., Frolov A.* A Polar Code Based Unsourced Random Access for the Gaussian MAC // Proc. 2019 IEEE 90th Vehicular Technology Conf. (VTC2019-Fall). Honolulu, HI, USA. Sept. 22–25, 2019. P. 1–5. https://doi.org/10.1109/VTCFall.2019.8891583
4. *Xie Z., Chen P., Mei Z., Long S., Cai K., Fang Y.* Polar-Coded Physical Layer Network Coding Over Two-Way Relay Channels // IEEE Commun. Lett. 2019. V. 23. № 8. P. 1301–1305. https://doi.org/10.1109/LCOMM.2019.2922633
5. *Andersson M., Rathi V., Thobaben R., Kliewer J., Skoglund M.* Nested Polar Codes for Wiretap and Relay Channels // IEEE Commun. Lett. 2010. V. 14. № 8. P. 752–754. https://doi.org/10.1109/LCOMM.2010.08.100875
6. *Tal I., Vardy A.* List Decoding of Polar Codes // IEEE Trans. Inform. Theory. 2015. V. 61. № 5. P. 2213–2226. https://doi.org/10.1109/TIT.2015.2410251
7. *Trifonov P., Miloslavskaya V.* Polar Subcodes // IEEE J. Select. Areas Commun. 2016. V. 34. № 2. P. 254–266. https://doi.org/10.1109/JSAC.2015.2504269
8. *Trifonov P., Trofimiuk G.* A Randomized Construction of Polar Subcodes // Proc. 2017 IEEE Int. Symp. on Information Theory (ISIT'2017). Aachen, Germany. June 25–30, 2017. P. 1863–1867. https://doi.org/10.1109/ISIT.2017.8006852
9. *Trifonov P.* A Score Function for Sequential Decoding of Polar Codes // Proc. 2018 IEEE Int. Symp. on Information Theory (ISIT'2018). Vail, CO, USA. June 17–22, 2018. P. 1470–1474. https://doi.org/10.1109/ISIT.2018.8437559
10. *Miloslavskaya V., Trifonov P.* Sequential Decoding of Polar Codes with Arbitrary Binary Kernel // Proc. IEEE 2014 Information Theory Workshop (ITW 2014). Hobart, TAS, Australia. Nov. 2–5, 2014. P. 376–380. https://doi.org/10.1109/ITW.2014.6970857
11. *Timokhin I., Ivanov F.* Sequential Polar Decoding with Cost Metric Threshold // Appl. Sci. 2024. V. 14. № 5. Paper No. 1847 (13 pp.). https://doi.org/10.3390/app14051847
12. *Korada S.B., Şaşoğlu E., Urbanke R.* Polar Codes: Characterization of Exponent, Bounds, and Constructions // IEEE Trans. Inform. Theory. 2010. V. 56. № 12. P. 6253–6264. https://doi.org/10.1109/TIT.2010.2080990
13. *Mori R., Tanaka T.* Channel Polarization on q-ary Discrete Memoryless Channels by Arbitrary Kernels // Proc. 2010 IEEE Int. Sympos. on Information Theory (ISIT 2010). Austin, TX, USA. June 13–18, 2010. P. 894–898. https://doi.org/10.1109/ISIT.2010.5513568

14. *Pfister H.D., Urbanke R.L.* Near-Optimal Finite-Length Scaling for Polar Codes over large alphabets // Proc. 2016 IEEE Int. Symp. on Information Theory (ISIT 2016), Barcelona, Spain, July 10–15, 2016. P. 215–219. https://doi.org/10.1109/ISIT.2016.7541292
15. *Fazeli A., Hassani H., Mondelli M., Vardy A.* Binary Linear Codes with Optimal Scaling: Polar Codes with Large Kernels // Proc. IEEE 2018 Information Theory Workshop (ITW'2018). Guangzhou, China. Nov. 25–29, 2018. P. 1–5. https://doi.org/10.1109/ITW.2018.8613428
16. *Trofimiuk G., Trifonov P.* Window Processing of Binary Polarization Kernels // IEEE Trans. Commun. 2021. V. 69. № 7. P. 4294–4305. https://doi.org/10.1109/TCOMM.2021.3072730
17. *Mori R., Tanaka T.* Performance of Polar Codes with the Construction Using Density Evolution // IEEE Commun. Lett. 2009. V. 13. № 7. P. 519–521. https://doi.org/10.1109/LCOMM.2009.090428
18. *Tal I., Vardy A.* How to Construct Polar Codes // IEEE Trans. Inform. Theory. 2013. V. 59. № 10. P. 6562–6582. https://doi.org/10.1109/TIT.2013.2272694
19. *Trifonov P.* Efficient Design and Decoding of Polar Codes // IEEE Trans. Commun. 2012. V. 60. № 11. P. 3221–3227. https://doi.org/10.1109/TCOMM.2012.081512.110872
20. *Vangala H., Viterbo E., Hong Y.* A Comparative Study of Polar Code Constructions for the AWGN Channel, https://arxiv.org/abs/1501.02473 [cs.IT], 2015.
21. *Bioglio V., Gabry F., Land I., Belfiore J.-C.* Multi-Kernel Polar Codes: Concept and Design Principles // IEEE Trans. Commun. 2020. V. 68. № 9. P. 5350–5362. https://doi.org/10.1109/TCOMM.2020.3006212
22. *Miloslavskaya V., Trifonov P.* Design of Binary Polar Codes with Arbitrary Kernels // Proc. 2012 IEEE Information Theory Workshop (ITW'2012). Lausanne, Switzerland. Sept. 3–7, 2012. P. 119–123. https://doi.org/10.1109/ITW.2012.6404639
23. *Trifonov P., Trofimiuk G.* Design of Polar Codes with Large Kernels // Probl. Inf. Transm. 2024. V. 60. № 4 (to appear). https://doi.org/10.1134/S0032946024040033
24. *Presman N., Shapira O., Litsyn S.* Mixed-Kernels Constructions of Polar Codes // IEEE J. Select. Areas Commun. 2016. V. 34. № 2. P. 239–253. https://doi.org/10.1109/JSAC.2015.2504278
25. *Trifonov P.* Binary Successive Cancellation Decoding of Polar Codes with Reed–Solomon Kernel // Proc. 2014 IEEE Int. Symp. on Information Theory (ISIT'2014). Honolulu, HI, USA. June 29 – July 4, 2014. P. 2972–2976. https://doi.org/10.1109/ISIT.2014.6875379
26. *Bioglio V., Land I.* On the Marginalization of Polarizing Kernels // Proc. 2018 IEEE 10th Int. Symp. on Turbo Codes & Iterative Information Processing (ISTC 2018). Hong Kong, China. Dec. 3–7, 2018. P. 1–5. https://doi.org/10.1109/ISTC.2018.8625378
27. *Гриссер Х., Сидоренко В.Р.* Апостериорно-вероятностное декодирование несистематических блоковых кодов // Пробл. передачи информ. 2002. Т. 38. № 3. С. 20–33. https://www.mathnet.ru/rus/ppi1313
28. *Trifonov P.* Algebraic Matching Techniques for Fast Decoding of Polar Codes with Reed–Solomon Kernel // Proc. 2018 IEEE Int. Symp. on Information Theory (ISIT 2018), Vail, CO, USA, June 17–22, 2018. P. 1475–1479. https://doi.org/10.1109/ISIT.2018.8437829
29. *Huang Z., Zhang S., Zhang F., Duanmu C., Zhong F., Chen M.* Simplified Successive Cancellation Decoding of Polar Codes With Medium-Dimensional Binary Kernels // IEEE Access. 2018. V. 6. P. 26707–26717. https://doi.org/10.1109/ACCESS.2018.2834465
30. *Trifonov P., Karakchieva L.* Recursive Processing Algorithm for Low Complexity Decoding of Polar Codes with Large Kernels // IEEE Trans. Commun. 2023. V. 71. № 9. P. 5039–5050. https://doi.org/10.1109/TCOMM.2023.3285773
31. *Трифонов П.В.* Построение и декодирование полярных кодов с большими ядрами: обзор // Пробл. передачи информ. 2023. Т. 59. № 1. С. 25–45. https://doi.org/10.31857/S0555292323010035
32. *Guillen i Fabregas A., Land I., Martinez A.* Extremes of Error Exponents // IEEE Trans. Inform. Theory. 2013. V. 59. № 4. P. 2201–2207. https://doi.org/10.1109/TIT.2012.2233271

33. *Колесников С.Г., Леонтьев В.М.* Серии формул для параметров Бхаттачарьи в теории полярных кодов // Пробл. передачи информ. 2023. Т. 59. № 1. С. 3–16. https://doi.org/10.31857/S0555292323010011
34. *Fazeli A., Vardy A.* On the Scaling Exponent of Binary Polarization Kernels // Proc. 52nd Annu. Allerton Conf. on Communication, Control, and Computing (Allerton'2014). Monticello, IL, USA. Sept. 30 – Oct. 3, 2014. P. 797–804. https://doi.org/10.1109/ALLERTON.2014.7028536
35. *Yao H., Fazeli A., Vardy A.* Explicit Polar Codes with Small Scaling Exponent // Proc. 2019 IEEE Int. Symp. on Information Theory (ISIT'2019). Paris, France. July 7–12, 2019. P. 1757–1761. https://doi.org/10.1109/ISIT.2019.8849741
36. *Ashikhmin A., Trifonov P.* Efficient Evaluation of Polarization Behavior for Large Kernels // Proc. 2023 IEEE Int. Symp. on Information Theory (ISIT 2023). Taipei, Taiwan. June 25–30, 2023. P. 1717–1722. https://doi.org/10.1109/ISIT54713.2023.10206496
37. *Kern D., Vorköper S., Kühn V.* A New Code Construction for Polar Codes Using Min-Sum Density // Proc. 2014 8th Int. Symp. on Turbo Codes and Iterative Information Processing (ISTC'2014). Bremen, Germany. Aug. 18–22, 2014. P. 228–232. https://doi.org/10.1109/ISTC.2014.6955119
38. *Zhou Y., Li R., Zhang H., Luo H., Wang J.* Polarization Weight Family Methods for Polar Code Construction // Proc. 2018 IEEE 87th Vehicular Technology Conf. (VTC Spring). Porto, Portugal. June 3–6, 2018. P. 1–5. https://doi.org/10.1109/VTCSpring.2018.8417498
39. *Hussami N., Korada S.B., Urbanke R.* Performance of Polar Codes for Channel and Source Coding // Proc. 2009 IEEE Int. Symp. on Information Theory (ISIT 2009). Seoul, Korea. June 28 – July 3, 2009. P. 1488–1492. https://doi.org/10.1109/ISIT.2009.5205860
40. *Trifonov P.* Randomized Polar Subcodes with Optimized Error Coefficient // IEEE Trans. Commun. 2020. V. 68. № 11. P. 6714–6722. https://doi.org/10.1109/TCOMM.2020.3018781
41. *Miloslavskaya V., Li Y., Vucetic B.* Design of Compactly Specified Polar Codes With Dynamic Frozen Bits Based on Reinforcement Learning // IEEE Trans. Commun. 2024. V. 72. № 3. P. 1257–1272. https://doi.org/10.1109/TCOMM.2023.3331532
42. *Oreshin M., Trifonov P.* Polar Subcodes with Improved Weight Spectrum // 2024 IEEE Int. Multi-Conf. on Engineering, Computer and Information Sciences (SIBIRCON). Novosibirsk, Russian Federation. Sept. 30 – Oct. 2, 2024. P. 41–46. https://doi.org/10.1109/SIBIRCON63777.2024.10758477
43. *Sun H., Viterbo E., Liu R.* Analysis of Polarization-adjusted Convolutional Codes (PAC): A Source-Channel Coding Method // Proc. 2021 IEEE Globecom Workshops (GC Wkshps). Madrid, Spain. Dec. 7–11, 2021. P. 1–6. https://doi.org/10.1109/GCWkshps52748.2021.9682079
44. *Rowshan M., Dau S.H., Viterbo E.* On the Formation of Min-Weight Codewords of Polar/PAC Codes and Its Applications // IEEE Trans. Inform. Theory. 2023. V. 69. № 12. P. 7627–7649. https://doi.org/10.1109/TIT.2023.3319015
45. *Kann T., Kudekar S., Bloch M.R.* A Path Metric Based Construction of Polarization-Adjusted Convolutional Codes // Proc. 2024 IEEE Int. Symp. on Information Theory (ISIT 2024). Athens, Greece. July 7–12, 2024. P. 2406–2411. https://doi.org/10.1109/ISIT57864.2024.10619693
46. *Coffey J.T., Kiely A.B.* The Capacity of Coded Systems // IEEE Trans. Inform. Theory. 1997. V. 43. № 1. P. 113–127. https://doi.org/10.1109/18.567656
47. *MacMullan S.J., Collins O.M.* The Capacity of Binary Channels That Use Linear Codes and Decoders // IEEE Trans. Inform. Theory. 1998. V. 44. № 1. P. 197–214. https://doi.org/10.1109/18.651018
48. *ten Brink S.* Convergence Behavior of Iteratively Decoded Parallel Concatenated Codes // IEEE Trans. Commun. 2001. V. 49. № 10. P. 1727–1737. https://doi.org/10.1109/26.957394
49. *Richardson T., Urbanke R.* Modern Coding Theory. Cambridge, UK: Cambridge Univ. Press, 2008.

50. *Trofimiuk G.* A Search Method for Large Polarization Kernels // Proc. 2021 IEEE Int. Symp. on Information Theory (ISIT'2021). Melbourne, Australia. July 12–20, 2021. P. 2084–2089. https://doi.org/10.1109/ISIT45174.2021.9517729

51. *Trofimiuk G.* Shortened Polarization Kernels // Proc. 2021 IEEE Globecom Workshops (GC Wkshps). Madrid, Spain. Dec. 7–11, 2021. P. 1–6. https://doi.org/10.1109/GCWkshps52748.2021.9681982

52. *Land I.* Reliability Information in Channel Decoding: Practical Aspects and Information Theoretical Bounds. Ph.D. Thesis. Faculty of Engineering, Christian-Albrechts-University of Kiel, Germany, 2005. Availale at https://macau.uni-kiel.de/receive/diss_mods_00001414.

53. *Trofimiuk G., Trifonov P.* Efficient Decoding of Polar Codes with Some 16×16 Kernels // Proc. IEEE 2018 Information Theory Workshop (ITW'2018). Guangzhou, China. Nov. 25–29, 2018. P. 11–15. https://doi.org/10.1109/ITW.2018.8613307

Трифонов Петр Владимирович
Трофимюк Григорий Андреевич
Факультет информационных технологий
и программирования, Университет ИТМО,
Санкт-Петербург
pvtrifonov@itmo.ru
gtrofimiuk@itmo.ru

Поступила в редакцию
09.10.2024
После доработки
22.11.2024
Принята к публикации
18.12.2024

ПРОБЛЕМЫ ПЕРЕДАЧИ ИНФОРМАЦИИ

Том 60 2024 Вып. 4

УДК 621.391 : 004.725.5

© 2024 г. **А.А. Федорищева, Д.В. Банков, А.И. Ляхов, Е.М. Хоров**

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ СЕТИ LoRaWAN ПРИ СОВМЕСТНОМ ОБСЛУЖИВАНИИ ПОДТВЕРЖДАЕМОГО И НЕПОДТВЕРЖДАЕМОГО ТИПОВ ТРАФИКА[1]

LoRaWAN является одной из самых популярных энергоэффективных сетей дальнего радиуса действия. Ключевым требованием в таких сетях является низкое энергопотребление. Для его уменьшения в данной статье разработан алгоритм выбора параметров сети. В алгоритме учитывается, что сенсоры могут передавать данные как с подтверждениями, так и без них. Также в алгоритме принимается во внимание ограничение на рабочий цикл.

Ключевые слова: LoRaWAN, LPWAN, математическое моделирование, энергопотребление, рабочий цикл.

DOI: 10.31857/S055529232404003X, **EDN:** ONDBON

§ 1. Введение

С каждым годом Интернет вещей (англ.: Internet of Things, IoT) набирает все большую популярность. Так, согласно [1], в 2021 г. число IoT-устройств превысило 10 млрд, и ожидается, что к 2030 г. число таких устройств возрастет более чем в два раза. Такой стремительный рост Интернета вещей привел к развитию беспроводных технологий с низким энергопотреблением, в частности, технологий энергоэффективных сетей дальнего радиуса действия (англ.: low-power wide-area network, LPWAN). Одной из наиболее популярных и известных во всем мире LPWAN-технологий является LoRaWAN, которая и исследуется в данной статье.

В сценариях развертывания сетей LoRaWAN часто необходимо обеспечивать доставку данных с заданной надежностью, т.е. с ограничением на долю потерянных пакетов (англ.: packet loss ratio, PLR) и с минимальным энергопотреблением конечных устройств (далее – сенсоров). Для обеспечения надежности передачи данных в технологии LoRaWAN может использоваться два режима передачи: с подтверждениями (в дальнейшем этот режим работы упоминается как режим ACK) и с безусловными повторами, но без подтверждений (в дальнейшем упоминается как режим NoACK). С одной стороны, использование режима ACK предпочтительнее, потому что в режиме NoACK сенсоры потенциально тратят больше канальных ресурсов и энергии на повторы, которые могут быть не нужны. С другой стороны, интенсивность подтверждаемого трафика ограничена. Это связано с тем, что сеть LoRaWAN работает в нелицензируемом диапазоне частот, где для обеспечения возможности сосуществования сетей разных операторов или сетей разных технологий вводится ограничение на рабочий цикл устройств (англ.: duty cycle, DC) – долю времени, в течение которого канал занят данным устройством. Это ограничение распространяется и на базовую станцию сети. При высокой интенсивности трафика сенсоров, работающих в режиме ACK, может возникнуть ситуация, когда для отправки подтверждений

[1] Исследование выполнено в рамках Госзадания № FFNU-2022-0035 ИППИ РАН.

на все получаемые кадры базовая станция должна будет нарушить ограничение на рабочий цикл. Чтобы этого не допустить, требуется ограничивать интенсивность подтверждаемого трафика.

Таким образом, интенсивность трафика в режиме ACK не должна превышать некоторого значения, зависящего от ограничения на рабочий цикл, а в режиме NoACK сенсоры потребляют большое количество энергии. Для того чтобы снизить среднее энергопотребление устройств в сети, предлагается некоторой доле устройств назначить режим ACK, а оставшейся доле – NoACK. В данной статье ставится задача нахождения доли устройств, работающих в режиме ACK, и количества попыток передач в режиме NoACK, при которых среднее энергопотребление сенсоров на передачу пакета будет минимальным и будут выполняться ограничения на рабочий цикл и PLR. Подобная задача рассматривалась для сетей LoRaWAN [2–4] и NB-Fi [5], однако в этих работах не приводится алгоритм выбора минимального энергопотребления.

Далее в § 2 описан протокол LoRaWAN, в § 3 описан исследуемый сценарий. В § 4 представлена математическая модель, в § 5 обсуждаются численные результаты, полученные с помощью имитационной модели, и приводится описание алгоритма для снижения энергопотребления, в § 6 представлено заключение.

§ 2. Описание протокола LoRaWAN

Сеть LoRaWAN состоит из сервера, базовых станций и сенсоров и имеет топологию "звезда из звезд". Взаимодействие сенсора и базовой станции происходит через основные и служебный каналы. В основном канале данные могут передаваться как в восходящем направлении, так и в нисходящем. Служебный канал предназначен только для передачи подтверждений. Обычно ограничение на рабочий цикл составляет 1% в основных каналах и 10% в служебных каналах. Для передачи сенсор выбирает один основной канал из F непересекающихся каналов и там осуществляет передачу данных.

Для обеспечения надежной доставки могут использоваться режимы работы с подтверждениями (ACK) и без подтверждений (NoACK). В режиме ACK базовая станция после получения данных отправляет два подтверждения: одно в основном канале, а другое – в служебном. Первое окно приема подтверждения в основном канале открывается через время T_1 (равное 1 с) после завершения отправки данных. Второе окно приема открывается в служебном канале через время T_2 (равное 2 с) после конца отправки данных. При неуспешной попытке передачи данные повторно отправляются через время τ_A,равномерно распределенное в интервале $[a, b]$. Количество попыток передач ограничено значением R_A.

В режиме NoACK сенсор R_N раз передает пакет с одними и теми же данными через случайную отсрочку τ_N, каждый раз выбирая случайный основной канал для передачи. В режиме NoACK случайная отсрочка равномерно распределена в интервале времени $[0; T_{\text{Rep}}]$. Режим NoACK не предполагает отправки подтверждений.

Оба этих режима работы имеют свои плюсы и минусы. Так, режим ACK позволяет не отправлять лишние пакеты с данными, что экономит энергопотребление сенсора и ресурсы канала. Но в то же время при высокой интенсивности трафика нельзя настроить все сенсоры для работы в режиме ACK из-за ограничения на рабочий цикл, поэтому целесообразно передавать часть трафика в режиме NoACK.

§ 3. Сценарий и постановка задачи

Рассмотрим следующий сценарий работы сети LoRaWAN. Пусть M сенсоров равномерно распределены в круге радиуса r, в центре которого находится базовая станция [6]. Сенсоры генерируют пакеты данных одинакового размера в моменты вре-

мени, соответствующие потоку Пуассона [7] с суммарной интенсивностью Λ. Они делятся на две группы по способу передачи кадров. Доля сенсоров x_A передает данные в режиме ACK, где максимальное число попыток передач составляет R_A. Оставшаяся доля сенсоров передает данные в режиме NoACK, где каждый пакет передается R_N раз.

В основных каналах данные и подтверждения передаются на одной сигнально-кодовой конструкции (СКК) [8]. В служебном канале используется самая надежная из доступных СКК (DR0).

На каждом сенсоре присутствует буфер, вмещающий в себя один пакет. Во время генерации пакет записывается в буфер. Если при генерации пакета в буфере уже есть пакет, то старый пакет вытесняется, а на его место помещается новый. Сенсор изымает пакет из буфера и начинает его передачу, если он не передает какой-либо другой пакет или если заканчивается попытка передачи предыдущего пакета.

В данной статье требуется разработать алгоритм для поиска доли сенсоров x_A, работающих в режиме ACK, и количества попыток передач R_N для режима NoACK, минимизирующих среднее энергопотребление сенсоров так, чтобы рабочий цикл базовой станции не превышал ограничение DC^*, а PLR не превышал ограничение PLR^*.

Таким образом, задачу поиска таких значений x_A и R_N можно записать следующим образом:

$$\begin{aligned} &\min_{x_A, R_N} E(\Lambda, x_A, R_N) \\ &\text{при условиях} \quad DC(\Lambda, x_A, R_N) \leqslant DC^*,\ PLR(\Lambda, x_A, R_N) \leqslant PLR^*, \end{aligned} \tag{1}$$

где $E(\Lambda, x_A, R_N)$ – среднее энергопотребление сенсора на успешную передачу пакета, $PLR(\Lambda, x_A, R_N)$ – доля потерянных пакетов и $DC(\Lambda, x_A, R_N)$ – рабочий цикл базовой станции.

§ 4. Математическая модель

Здесь приводится математическая модель, позволяющая оценить долю потерянных пакетов, рабочий цикл базовой станции и энергопотребление сенсоров в сети LoRaWAN. За основу взята математическая модель из работы [4], которая была расширена для режима NoACK. Также был использован подход из [5] с некоторыми изменениями, учитывающими метод доступа к каналу в сети LoRaWAN. Все формулы выводятся в предположении малой интенсивности трафика.

В модели величины, относящиеся к режиму ACK, обозначены индексом A, а величины, относящиеся к режиму NoACK, обозначены индексом N. Индекс N/A означает, что после подстановки N или A получается величина, соответствующая режиму NoACK или ACK.

Далее в п. 4.1 рассчитывается вероятность успешной доставки кадра. В п. 4.2 определяется вероятность того, что сенсор начнет передачу кадра. В п. 4.3 представлены формулы для расчета доли потерянных пакетов. В п. 4.4 оценивается энергопотребление, а в п. 4.5 – рабочий цикл.

4.1. Вероятность успешной доставки кадра. Суммарная интенсивность генерации кадров для передачи с учетом двух режимов работы рассчитывается как

$$\lambda = \Lambda x_A + \Lambda(1 - x_A)R_N. \tag{2}$$

Вероятности успешной первой попытки передачи будут одинаковыми для режимов ACK и NoACK, т.е.

$$P^A_{S,\text{ini}} = P^N_{S,\text{ini}} = P_{S,\text{ini}},$$

где $P_{S,\text{ini}}$ определяется по результатам работы [4, раздел IV-A] с подстановкой суммарной интенсивности (2). Вероятность успешной повторной передачи $P^A_{S,\text{re}}$ для режима ACK определяется, как и в [4, раздел IV-B], с подстановкой интенсивности (2).

Вероятность успешного повтора после неудачной попытки передачи в режиме работы NoACK равна

$$P^N_{S,\text{re}} = \frac{\left(x_A + R_N(1-x_A)\left((1-P^c)\frac{R_N-1}{R_N} + \frac{1}{R_N}\right)\right)P^N_{S,\text{ini}}}{x_A + (1-x_A)R_N}, \tag{3}$$

где P^c – вероятность возникновения повторной коллизии для кадров, передаваемых после случайной отсрочки, которая определятся из работы [4]. Первое и второе слагаемые в скобках описывают повтор после коллизии, которая произошла с пакетом, передаваемым в режиме ACK и NoACK соответственно. После коллизии с пакетом, который передавался в режиме ACK, вероятнее всего больше не будет пересечения с этим пакетом. Это связано с тем, что время, через которое произойдет повторная попытка передачи в режиме ACK больше, чем время, через которое происходит повтор в режиме NoACK. Поэтому вероятность успешной передачи равна $P^N_{S,\text{ini}}$. Коллизия с пакетом, передаваемым в режиме NoACK, может вызвать пересечение их повторных пакетов, поскольку интервалы времен, в течение которых происходит повторная передача, пересекаются. В связи с этим повторной коллизии не будет с вероятностью

$$(1-P^c)\frac{R_N-1}{R_N},$$

где $\frac{R_N-1}{R_N}$ означает, что у пакета, с которым была коллизия, в предыдущий раз была не последняя попытка передачи, т.е. будет по крайней мере еще одна попытка передачи, которая снова вызовет коллизию. Если до текущей повторной отправки была коллизия с пакетом, у которого была последняя попытка передачи, то коллизии с этим пакетом больше не будет, т.е. с вероятностью

$$\frac{1}{R_N}P^N_{S,\text{ini}}$$

будет успешная повторная попытка передачи.

4.2. Вероятность начала обслуживания кадра. Согласно сценарию буфер сенсора вмещает в себя только один кадр. Если новый кадр генерируется во время передачи текущего кадра, то после окончания попытки передачи старый кадр удаляется и начинается обслуживание нового. Если до окончания попытки передачи генерируется несколько новых кадров, то все кадры, кроме самого нового, отбрасываются. Найдем вероятность того, что кадр не будет отброшен до начала его передачи:

$$P^{N/A}_{\text{start}} = \left(1-P^{N/A}_{\text{busy}}\right) + P^{N/A}_{\text{busy}} \times \left(P^{N/A}_{\text{ini}} P_{\text{buf}}\left(T^{N/A}_{\text{ini}}\right) + P^{N/A}_{\text{re}} P_{\text{buf}}\left(T^{N/A}_{\text{re}}\right)\right), \tag{4}$$

где $P^{N/A}_{\text{busy}}$ – вероятность того, что во время генерации сенсор занят передачей пакета, $P^{N/A}_{\text{ini}}$ и $P^{N/A}_{\text{re}}$ – вероятности того, что при этом текущая попытка передачи является первой и повторной соответственно, $P_{\text{buf}}(T)$ – вероятность того, что в течение времени T пакет не удалится из буфера, $T^{N/A}_{\text{ini}}$ и $T^{N/A}_{\text{re}}$ – продолжительности первой и повторной попытки передачи соответственно. Первое слагаемое описывает случай, когда сенсор сгенерировал пакет в тот момент, когда не было передачи другого пакета, т.е. пакет сразу же начал передаваться, как только он был сгенерирован. С вероятностью $P^{N/A}_{\text{busy}}$ сенсор занят передачей пакета, поэтому новый пакет будет

помещен в буфер. В этом случае сенсор передает пакет из буфера, если на сенсоре больше не было сгенерировано пакетов во время передачи пакета. Текущая попытка передачи является первой с вероятностью $P_{\text{ini}}^{N/A}$ и повторной с вероятностью $P_{\text{re}}^{N/A}$. Найдем данные величины.

Вероятность того, что во время генерации сенсор занят передачей другого пакета, можно найти как отношение задержки передачи пакета $D^{N/A}$ к среднему времени генерации $\frac{M}{\Lambda}$, где M – количество сенсоров:

$$P_{\text{busy}}^{N/A} = \min\left\{\frac{\Lambda D^{N/A}}{M}, 1\right\}. \tag{5}$$

Задержка D^N в режиме NoACK определяется аналогично [5]:

$$D^N = T_{\text{Data}} + \mathbb{1}\{R_N > 1\} \times P_{G,\text{ini}}^N \sum_{i=0}^{R_N-2} \left(P_{G,\text{re}}^N\right)^i \left(T_{\text{Data}} + \frac{T_{\text{Rep}}}{2}\right), \tag{6}$$

где $\mathbb{1}\{\ldots\}$ – индикаторная функция, которая равна 1, если условие в скобках выполняется, и равна 0, если условие не выполняется, T_{Data} – длительность кадра с данными, $\frac{T_{\text{Rep}}}{2}$ – средняя отсрочка в режиме NoACK, $P_{G,\text{ini}}^N$ и $P_{G,\text{re}}^N$ – вероятности того, что во время первой и повторной попытки передачи, соответственно, на сенсоре не сгенерируется новый кадр. Эти величины вычисляются так же, как и в [5, п. 5.2].

Вычислим задержку для режима ACK. Если была успешная попытка передачи, то задержка вычисляется как

$$D_S^A = T_{\text{Data}} + P_{\text{Ack1}}\left(T_1 + T_{\text{Ack}}\right) + \left(1 - P_{\text{Ack1}}\right)\left(T_2 + T_{\text{Ack0}}\right), \tag{7}$$

где T_{Ack} и T_{Ack0} – длительности кадров подтверждения в основном и служебном канале соответственно, P_{Ack1} – вероятность доставки первого подтверждения, которая находится как в [4, раздел IV-A] с подстановкой суммарной интенсивности (2). Средняя задержка после неуспешной попытки передачи вычисляется как

$$D_{\text{Re}}^A = T_{\text{Data}} + T_2 + T_{\text{listen0}} + \frac{a+b}{2}, \tag{8}$$

где T_{listen0} – длительность окна приема в служебном канале, $\frac{a+b}{2}$ – среднее время, через которое будет совершена повторная попытка передачи.

Итоговая формула для вычисления задержки имеет следующий вид:

$$D^A = D_S^A + \mathbb{1}\{R_A > 1\} \times (1 - P_{S,\text{ini}}^A) P_{S,\text{re}}^A P_{G,\text{ini}}^A \times$$
$$\times \sum_{i=1}^{R_A-1} i D_{\text{re}}^A \left(\left(1 - P_{S,\text{re}}^A\right) P_{G,\text{re}}^A\right)^{i-1}, \tag{9}$$

где $P_{G,\text{ini}}^A$ и $P_{G,\text{re}}^A$ – вероятности того, что во время первой и повторной попыток передач, соответственно, не будут сгенерированы новые пакеты с данными в режиме ACK. Данные величины вычисляются так же, как и в [4]. Здесь первое слагаемое – задержка при успешной попытке передачи. Второе слагаемое – время, затрачиваемое на повторные попытки передачи. При доставке после i повторных попыток передачи к задержке добавляется iD_{re}^A, при этом вероятность такого события равна произведению вероятности не доставить кадр с первой попытки $(1 - P_{S,\text{ini}}^A)$ и во время $i-1$ повторных попыток $((1-P_{S,\text{re}}^A)^{i-1})$, но доставить на i-й повторной попыт-

ке ($P^A_{S,\mathrm{re}}$), и при этом учитывается, что во время всех повторных попыток передачи кадр не был вытеснен новым кадром ($P^A_{G,\mathrm{ini}}(P^A_{G,\mathrm{re}})^{i-1}$).

Согласно [5] вероятность $P_{\mathrm{buf}}(T)$ того, что в течение времени T пакет не удалится из буфера, рассчитывается как

$$P_{\mathrm{buf}}(T) = \frac{M}{\Lambda \times T}\left(1 - e^{-\frac{\Lambda}{M}T}\right). \tag{10}$$

Продолжительности первой попытки передачи T_{ini} в режимах NoACK и ACK равны $T^N_{\mathrm{ini}} = T_{\mathrm{Data}}$ и $T^A_{\mathrm{ini}} = D^A_S$ соответственно. Длительности повторной попытки передачи T_{re} в режимах NoACK и ACK равны $T^N_{\mathrm{re}} = T_{\mathrm{Data}} + T_{\mathrm{Rep}}$ и $T^A_{\mathrm{re}} = D^A_{\mathrm{Re}}$ соответственно.

Чтобы найти $P^{N/A}_{\mathrm{ini}}$ и $P^{N/A}_{\mathrm{re}}$, нужно посчитать среднее количество попыток передач $R^{N/A}_{\mathrm{av}}$ в режимах NoACK и ACK. Для режима ACK

$$R^A_{\mathrm{av}} = 1 + \mathbb{1}\{R_A > 1\} \times \left(1 - P^A_{S,\mathrm{ini}}\right)P^A_{G,\mathrm{ini}} \sum_{i=0}^{R_A-2} \left(\left(1 - P^A_{S,\mathrm{re}}\right)P^A_{G,\mathrm{re}}\right)^i. \tag{11}$$

Для режима NoACK

$$R^N_{\mathrm{av}} = 1 + \mathbb{1}\{R_N > 1\} \times P^N_{G,\mathrm{ini}} \sum_{i=0}^{R_N-2} \left(P^N_{G,\mathrm{re}}\right)^i. \tag{12}$$

Тогда

$$P^{N/A}_{\mathrm{ini}} = \frac{1}{R^{N/A}_{\mathrm{av}}}$$

– вероятность первой попытки передачи,

$$P^{N/A}_{\mathrm{re}} = \frac{R^{N/A}_{\mathrm{av}} - 1}{R^{N/A}_{\mathrm{av}}}$$

– вероятность повтора.

Подставив $P^{N/A}_{\mathrm{busy}}$, $P^{N/A}_{\mathrm{ini}}$, $P^{N/A}_{\mathrm{re}}$, $T^{N/A}_{\mathrm{ini}}$, $T^{N/A}_{\mathrm{re}}$ в (4), найдем вероятность начала обслуживания кадра $P^{N/A}_{\mathrm{start}}$.

4.3. Доля потерянных пакетов. Средняя по сети доля потерянных пакетов определятся как

$$PLR = PLR_A x_A + PLR_N(1 - x_A), \tag{13}$$

где PLR_A и PLR_N – значения PLR в режимах ACK и NoACK соответственно. Эти величины могут быть найдены как

$$PLR_{N/A} = 1 - P^{N/A}_S P^{N/A}_{\mathrm{start}}. \tag{14}$$

Вероятности успешной попытки передачи в режиме ACK и NoACK вычисляются как

$$P^{N/A}_S = P^{N/A}_{S,\mathrm{ini}} + \mathbb{1}\{R_{N/A} > 1\} \times \left(1 - P^{N/A}_{S,\mathrm{ini}}\right)P^{N/A}_{G,\mathrm{ini}}P^{N/A}_{S,\mathrm{re}} \times$$
$$\times \sum_{i=0}^{R_{N/A}-2} \left(\left(1 - P^{N/A}_{S,\mathrm{re}}\right)P^{N/A}_{G,\mathrm{re}}\right)^i. \tag{15}$$

Первое слагаемое описывает успешную передачу во время первой попытки. Второе слагаемое – вероятность неуспешной первой попытки, но успешной i-й попытки и вероятность того, что пакет не будет вытеснен после каждой попытки из-за генерации нового пакета.

4.4. Энергопотребление. Оценим среднее энергопотребление сенсора за одну успешную доставку. В данной метрике учитывается как энергопотребление для успешных попыток передачи, так и для неуспешных попыток передачи.

Учтем, что энергопотребление для режима ACK и NoACK будет разным, тогда среднее энергопотребление можно найти как

$$E = E_A x_A + E_N(1 - x_A), \tag{16}$$

где E_A, E_N – энергопотребление сенсора для режимов ACK и NoACK соответственно.

Рассмотрим режим ACK. В этом режиме сенсор тратит энергию на отправку пакета с данными, прослушивание канала и получение подтверждения. В LoRaWAN окно приема подтверждения имеет небольшую длительность, поэтому предположим, что подтверждения будут приходить сразу же, как только открывается окно приема подтверждения.

Во время первой попытки передачи сенсор тратит энергию E_{TX} на отправку пакета, и с вероятностью $P^A_{S,\mathrm{ini}}$ попытка передачи успешна. Если получено подтверждение в основном канале, то среднее энергопотребление для приема подтверждения составляет $P^A_{S,\mathrm{ini}} E_{\mathrm{RX}}$, где E_{RX} – энергия, затрачиваемая для приема подтверждения в основном канале. С учетом того, что подтверждение было получено только в основном канале, среднее энергопотребление равно $P^A_{S,\mathrm{ini}} E_{\mathrm{RX}} P_{\mathrm{Ack1}}$, где P_{Ack1} – вероятность успешного приема подтверждения в основном канале, которое определяется в [4]. Если подтверждение в основном канале не приходит, но получено подтверждение в служебном канале, то среднее энергопотребление на получение подтверждения в служебном канале составляет

$$E_{\mathrm{listen}} + P^A_{S,\mathrm{ini}} E_{\mathrm{RX0}},$$

где E_{listen} – энергия, затрачиваемая на прослушивание всего окна приема в основном канале, E_{RX0} – энергия, затрачиваемая для приема подтверждения в служебном канале. С учетом вероятности того, что подтверждение было получено только в служебном канале, среднее энергопотребление для прослушивания основного канала и приема подтверждения в служебном канале составит

$$\big(E_{\mathrm{listen}} + P^A_{S,\mathrm{ini}} E_{\mathrm{RX0}}\big)\big(1 - P_{\mathrm{Ack1}}\big).$$

Таким образом, формула энергопотребления для первой успешной попытки передачи примет следующий вид:

$$E^S_{\mathrm{ini}} = E_{\mathrm{TX}} + P^A_{S,\mathrm{ini}} E_{\mathrm{RX}} P_{\mathrm{Ack1}} + \big(E_{\mathrm{listen}} + P^A_{S,\mathrm{ini}} E_{\mathrm{RX0}}\big)\big(1 - P_{\mathrm{Ack1}}\big). \tag{17}$$

Если первая попытка передачи была неуспешной, но какая-то повторная попытка передачи оказалась успешной, то к среднему энергопотреблению станции добавляется средняя энергия, потребляемая за все повторные попытки передачи, равная

$$E^S_{\mathrm{re}} = \big(1 - P^A_{S,\mathrm{ini}}\big) P^A_{G,\mathrm{ini}} P^A_{S,\mathrm{re}} \sum_{i=1}^{R_A - 1} \Big(i(E_{\mathrm{listen}} + E_{\mathrm{listen0}} + E_{\mathrm{TX}}) + \\ + E_{\mathrm{RX}} P_{\mathrm{Ack1}} + (E_{\mathrm{listen}} + E_{\mathrm{RX}})(1 - P_{\mathrm{Ack1}}) \Big) \Big(\big(1 - P^A_{S,\mathrm{re}}\big) P^A_{G,\mathrm{re}} \Big)^{i-1}, \tag{18}$$

где E_{listen0} – энергия, затрачиваемая сенсором для прослушивания служебного канала.

Если сенсор не получил подтверждение после всех R_A попыток передач, то среднее энергопотребление E^F_{re} на повторные попытки передачи рассчитывается как

$$E^F_{\text{re}} = \big(1 - P^A_{S,\text{ini}}\big) P^A_{G,\text{ini}} \big(1 - P^A_{S,\text{re}}\big)^{R_A - 1} \big(P^A_{G,\text{re}}\big)^{R_A - 1} \times$$
$$\times \Big((R_A - 1) \times E_{\text{TX}} + R_A \times (E_{\text{listen}} + E_{\text{listen0}}) \Big). \tag{19}$$

Если новый пакет был сгенерирован во время неуспешной первой попытки передачи, тогда передаваемый пакет будет вытеснен. В этом случае сенсор будет тратить на прослушивание двух каналов после передачи вытесненного кадра энергию

$$E^G_{\text{ini}} = \big(1 - P^A_{S,\text{ini}}\big)\big(1 - P^A_{G,\text{ini}}\big)\big(E_{\text{listen}} + E_{\text{listen0}}\big). \tag{20}$$

В случае если новый пакет был сгенерирован во время повторной попытки передачи, средняя энергия, затраченная на прослушивание канала и повторные попытки передачи, будет равна

$$E^G_{\text{re}} = \big(1 - P^A_{S,\text{ini}}\big) P^A_{G,\text{ini}} \big(1 - P^A_{G,\text{re}}\big) \mathbb{1}\{R_A > 1\} \times$$
$$\times \sum_{i=0}^{R_A - 2} \big(P^A_{G,\text{re}}\big)^i \big(1 - P_{S,\text{re}}\big)^{i+1} \Big((i+2)(E_{\text{listen}} + E_{\text{listen0}}) + (i+1) E_{\text{TX}} \Big). \tag{21}$$

Сумма всех вышеперечисленных энергий равняется энергопотреблению сенсора, затрачиваемого на один переданный пакет. Поделив сумму на вероятность успеха P^A_S, можно получить среднее энергопотребление сенсора за один успешно переданный пакет в режиме ACK:

$$E_A = \frac{E^S_{\text{ini}} + E^S_{\text{re}} + E^F_{\text{re}} + E^G_{\text{ini}} + E^G_{\text{re}}}{P^A_S}. \tag{22}$$

Перейдем к подсчету энергопотребления в режиме NoACK. В этом режиме энергия тратится только для передачи пакетов с данными.

Повторные попытки передачи происходят, если новый пакет не будет сгенерирован во время передачи. Поделив суммарное энергопотребление в режиме NoACK на вероятность успешной передачи P^N_S, можно получить энергопотребление за один успешно переданный пакет:

$$E_N = \frac{E_{\text{TX}} + \mathbb{1}\{R_N > 1\} E_{\text{TX}} P^N_{G,\text{ini}} \sum\limits_{i=0}^{R_N - 2} \big(P^N_{G,\text{re}}\big)^i}{P^N_S}. \tag{23}$$

4.5. Рабочий цикл. Предположим, что интенсивность восходящего трафика мала, поэтому каждый сенсор поодиночке не нарушает ограничений на рабочий цикл. Однако сенсоров достаточно много, и базовая станция может нарушить ограничение на рабочий цикл в нисходящем канале, поэтому рассмотрим ограничение на рабочий цикл для подтверждений. Интенсивность генерации подтверждаемого трафика определяется как Λx_A. Согласно сценарию, в нисходящем канале передаются только пакеты с подтверждениями, тогда время, в течение которого нисходящий канал занят, определяется длительностью пакета с подтверждением T_{Ack}. Учтем, что подтверждение будет отправлено в случае успешной доставки кадра с данными, т.е. с вероятностью P^A_S. Для рабочего цикла в основном канале нужно учесть, что этот основной канал будет выбран с вероятностью $\frac{1}{F}$. Таким образом, рабочий цикл

Таблица 1

Энергопотребление сенсоров

	Мощность, мВт	Время, с	Энергия, мДж
Передача в основном канале, TX	419,60	0,191	80,14
Прием в основном канале, RX	44,06	0,074	3,26
Прослушивание в основном канале, listen	44,06	0,025	1,10
Передача в служебном канале, TX0	419,60	2,990	1254,60
Прием в служебном канале, RX0	44,06	1,090	46,70
Прослушивание в служебном канале, listen0	44,06	0,401	17,67

для одного основного канала может быть найден как

$$DC_{\text{main}} = \min\left(\frac{\Lambda x_A \times P_S^A \times T_{\text{Ack}}}{F}, 1\right). \tag{24}$$

Для служебного канала аналогично получаем

$$DC_{\text{service}} = \min\Big(\Lambda x_A \times P_S^A \times T_{\text{Ack0}}, 1\Big), \tag{25}$$

где T_{Ack0} – длительность кадра с подтверждением в служебном канале, где данные передаются на самой низкой скорости.

§ 5. Численные результаты

Для валидации математической модели была разработана дискретно-событийная имитационная модель, учитывающая особенности метода доступа к каналу в сети LoRaWAN. Далее приведены результаты для $M = 1000$ сенсоров, равномерно распределенных в круге радиуса $r = 1$ км.

Случайная отсрочка τ_A для режима ACK равномерно распределена в интервале $[a; b]$, и согласно спецификации $a = 1$ с, $b = 3$ с. Случайная отсрочка τ_N для режима NoACK равномерно распределена в интервале времени $[0; T_{\text{Rep}}]$, где $T_{\text{Rep}} = 2$ с. Также согласно спецификации $T_1 = 1$ с, $T_2 = 2$ с. Максимальное число попыток передачи в режиме ACK равно $R_A = 8$.

Согласно [9] ограничение на рабочий цикл базовых станций, работающих в Европе в диапазоне частот 863–870 МГц, составляет 1% в основном канале и 10% в служебном канале. Будем рассматривать ограничение на долю потерянных (т.е. не доставленных по любым причинам) пакетов, равное 10^{-3}.

Все сенсоры передают данные на скорости 3125 бит/с (DR4). Сенсоры передают сигналы с мощностью 14 дБм, затухание сигнала вычислялось с использованием модели Окамуры – Хата [10]. Считаем, что на рассматриваемой скорости передача успешна, если соотношение сигнал-интерференция-шум больше $-7{,}5$ дБ.

Значения мощностей для прослушивания канала, передачи и приема пакета определены в [11] и приведены в табл. 1. Считаем, что для прослушивания канала во время окна приема подтверждения затрачивается та же мощность, что и для приема пакета. Конкретные продолжительности окон приема подтверждения не определены в спецификации, однако сказано, что они должны быть достаточными, чтобы определить преамбулу пакета подтверждения. Будем считать, что длительность прослушивания канала равна длительности преамбулы. Все длительности были вычислены согласно [12]. Кроме того, в табл. 1 рассчитано энергопотребление для передачи, приема кадра, а также прослушивания канала в служебном и основном каналах.

Перейдем к полученным результатам. Сначала рассмотрим рис. 1, на котором представлена зависимость PLR от суммарной интенсивности генерируемого трафика Λ в сети. Горизонтальной линией показана прямая $PLR^* = 10^{-3}$. Очевидно, что

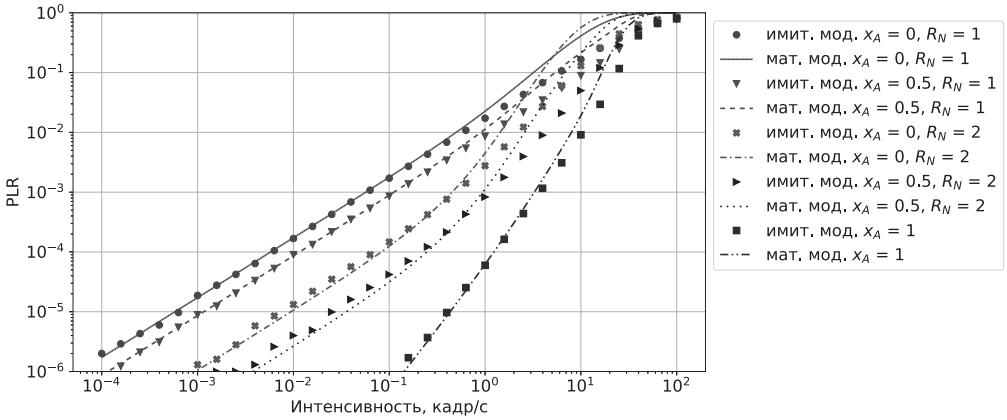


Рис. 1. Зависимость PLR от интенсивности трафика

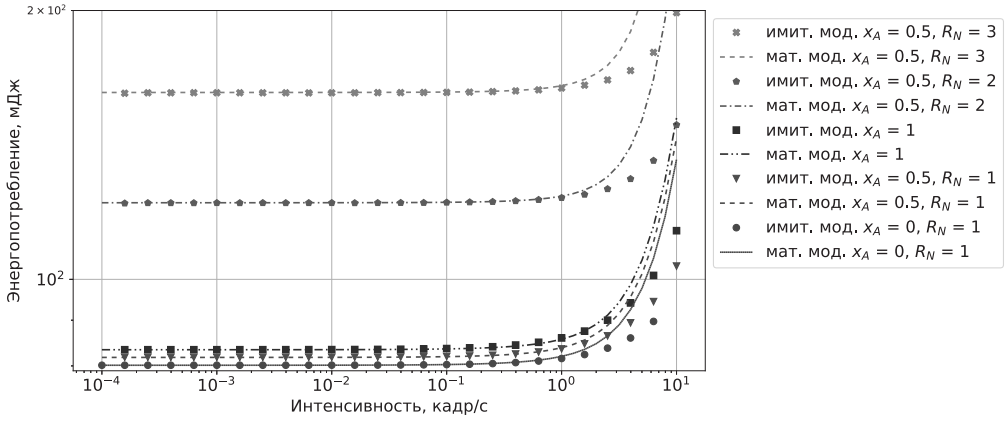


Рис. 2. Зависимость энергопотребления от интенсивности трафика

при $R_N = 1$ и $x_A = 0$, т.е. когда все сенсоры передают в режиме NoACK и делают только одну попытку передачи, наблюдается самое высокое значение для PLR. При увеличении доли x_A PLR уменьшается, так как в режиме ACK у сенсора есть возможность совершить дополнительные попытки передачи в случае неуспеха. При $x_A = 1$ наблюдается самое малое значение PLR из всех представленных. Также отметим, что при $x_A < 1$ увеличение R_N приводит к снижению PLR.

На рис. 2 приведена зависимость энергопотребления сенсоров от суммарной интенсивности генерируемого трафика Λ. Из рисунка видно, что энергопотребление уменьшается при уменьшении x_A, поскольку сенсоры, работающие в режиме ACK, потребляют дополнительную энергию для прослушивания канала и приема пакета с подтверждением. При больших интенсивностях энергопотребление значительно возрастает. Это связано с тем, что вероятность коллизии близка к единице и сенсоры в режиме ACK вынуждены делать повторные попытки передачи, что и приводит к увеличению энергопотребления. Сенсоры в режиме NoACK совершают по одной попытке передачи, которые в большинстве случаев оказываются неудачными, что также приводит к возрастанию среднего энергопотребления.

При малых интенсивностях при $R_N = 2$ сенсоры потребляют почти в два раза больше энергии, чем при $R_N = 1$, поскольку делается в два раза больше попы-

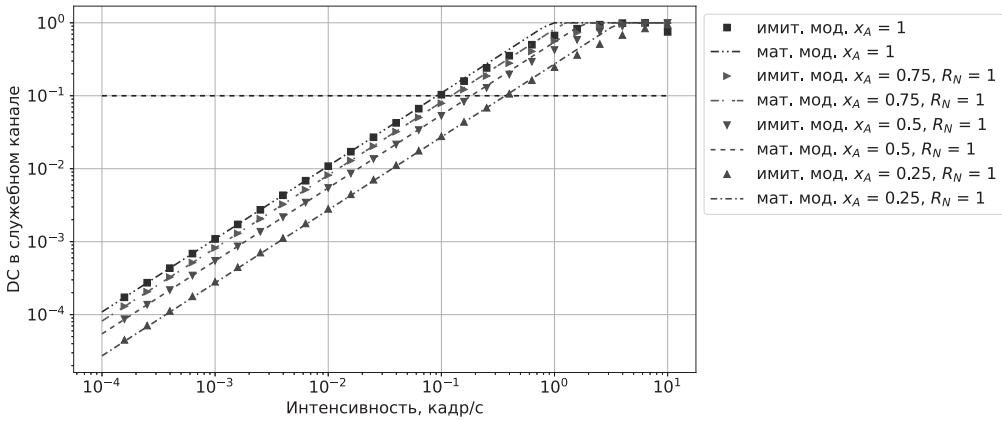


Рис. 3. Зависимость DC в служебном канале от интенсивности трафика

ток передач. Также из рисунка видно, что энергопотребление сенсоров возрастает с увеличением количества попыток передач.

Отметим, что при увеличении интенсивности трафика и R_N возрастает разница в результатах, полученных с помощью математической и имитационной моделей (рис. 2). Данная ошибка связана с тем, что вероятности коллизии в математической модели находятся в предположении малой интенсивности трафика (§ 4), в результате чего пренебрегается коллизиями более чем двух кадров. При большой интенсивности трафика в канале данное допущение вносит заметную ошибку.

Рассмотрим рис. 3, на котором приведена зависимость рабочего цикла базовой станции в служебном канале от суммарной интенсивности генерируемого трафика Λ при $R_N = 1$. Горизонтальной линией показана прямая $DC^* = 10\%$. Очевидно, при увеличении x_A возрастает рабочий цикл. Заметим, что рабочий цикл определяется количеством подтверждений, которые при малых интенсивностях трафика будут все отправлены, поэтому график в этом случае принимает линейный вид. При больших интенсивностях график DC выходит на плато, так как достигается емкость служебного канала.

Заметим, что длительность кадра подтверждения в основном канале значительно ниже, чем в служебном канале. Так, согласно табл. 1 длительность кадра подтверждения для самой высокой скорости в основном канале составляет $T_{\text{Ack}} = 0{,}074\,\text{с}$, а длительность кадра подтверждения в служебном канале составляет $T_{\text{Ack0}} = 1{,}09\,\text{с}$. Из математической модели следует, что при низкой интенсивности трафика рабочий цикл в служебном канале в $\dfrac{T_{\text{Ack0}}F}{T_{\text{Ack}}}$ раз больше, чем в основном канале. Таким образом, рабочий цикл в служебном канале почти в 45 раз больше, чем в основном, и поэтому можно следить за выполнением ограничения рабочего цикла только в служебном канале, так как для основного канала ограничение, равное 1%, будет выполнено, если соблюдается ограничение для служебного.

Подводя итоги полученным результатам, сформулируем алгоритм для выбора параметров x_A и R_N (см. Алгоритм 1) для заданных значений интенсивности трафика Λ и ограничений DC^* и PLR^*.

Если $PLR(\Lambda, x_A = 0, R_N = 1) \leqslant PLR^*$, то минимальное энергопотребление обеспечивается, когда все устройства работают в режиме NoACK и передают один раз, т.е. нужно назначить $x_A = 0$, $R_N = 1$. Если данное условие не выполняется, то нужно проверить, возможно ли выполнение ограничений на PLR при минимальном возможном значении PLR, т.е. когда все устройства передают в режиме ACK

Алгоритм 1 Алгоритм для выбора x_A и R_N

Require: Λ, DC^*, PLR^*
Ensure: x_A, R_N
if $PLR(\Lambda, x_A = 0, R_N = 1) \leqslant PLR^*$ **then**
  $x_A \leftarrow 0, R_N \leftarrow 1$
  return x_A, R_N
else
  if $PLR(\Lambda, x_A = 1) > PLR^*$ **then**
    return Невозможно выполнить ограничение на PLR и DC
  else
    Найти $\widehat{x}_A$ как $PLR(\Lambda, \widehat{x}_A, R_N = 1) = PLR^*$
    if $DC(\Lambda, \widehat{x}_A, R_N = 1) \leqslant DC^*$ **then**
      $x_A \leftarrow \widehat{x}_A, R_N \leftarrow 1$
      return x_A, R_N
    else
      for $\widehat{R}_N = 2$ до $R_N^{\max}$ **do**
        Найти $\widehat{x}_A$ как $\max x_A$: $DC(\Lambda, \widehat{x}_A, \widehat{R}_N) \leqslant DC^*$
        if $PLR(\Lambda, \widehat{x}_A, \widehat{R}_N) \leqslant PLR^*$ **then**
          $x_A \leftarrow \widehat{x}_A, R_N \leftarrow \widehat{R}_N$
          return x_A, R_N
      return Невозможно выполнить ограничение на PLR и DC

($x_A = 1$). Если окажется, что $PLR(\Lambda, x_A = 1) > PLR^*$, то невозможно выполнить ограничение на PLR.

В противном случае для минимизации энергопотребления при $R_N = 1$ необходимо найти значение $\widehat{x}_A$, при котором выполняется равенство

$$PLR(\Lambda, \widehat{x}_A, R_N = 1) = PLR^*,$$

а далее проверить выполнение ограничения

$$DC(\Lambda, \widehat{x}_A, R_N = 1) \leqslant DC^*.$$

Если при найденном $\widehat{x}_A$ и при $R_N = 1$ ограничение на рабочий цикл выполнено, то искомые значения $x_A = \widehat{x}_A$ и $R_N = 1$ для минимизации энергопотребления найдены.

Если ограничение на DC не выполняется, то нужно перейти к большему числу повторов: $R_N = 2$. При таком числе повторов для минимизации энергопотребления необходимо найти $\widehat{x}_A$ как максимальное значение x_A, такое что

$$DC(\Lambda, \widehat{x}_A, \widehat{R}_N) \leqslant DC^*.$$

Если для найденного $\widehat{x}_A$ выполняется ограничение

$$PLR(\Lambda, \widehat{x}_A, R_N = 2) \leqslant PLR^*,$$

то искомые значения $x_A = \widehat{x}_A$ и $R_N = 2$ найдены. Если ограничения не выполняются, то нужно увеличить R_N на единицу и повторить предыдущий шаг алгоритма с поиском x_A и проверкой выполнения ограничения на PLR. Увеличение R_N будет происходить до тех пор, пока не найдем удовлетворяющие ограничениям значения x_A и R_N или пока значение R_N не достигнет максимального значения $R_N^{\max}$. Если R_N достигло значения $R_N^{\max}$ и при этом не выполняется ограничение на PLR, то невозможно одновременно выполнить ограничение на PLR и рабочий цикл.

Заметим, что алгоритм решает задачу при ограничениях на PLR ниже 1%. В то же время большие значения PLR не являются практически интересными.

§ 6. Заключение

В данной статье была исследована сеть LoRaWAN, в которой сенсоры могут работать в режиме с подтверждениями и в режиме без подтверждений, но с безусловными повторами. Была разработана математическая модель, позволяющая оценить долю потерянных пакетов, рабочий цикл и энергопотребление сенсоров при заданной интенсивности трафика, доле устройств, работающих в режиме с подтверждениями, и количестве попыток передач в режиме без подтверждений. Был разработан алгоритм, который при помощи математической модели находит значения параметров сети, минимизирующих энергопотребление устройств при заданных ограничениях на долю потерянных пакетов и рабочий цикл.

В расширенной версии данной статьи [13] приведено более подробное описание протокола. Также в расширенной версии приведен график, сравнивающий энергопотребление алгоритма и энергопотребление базовых конфигураций сети и показывающий эффективность алгоритма.

В дальнейшем планируется реализация предложенного алгоритма в устройствах LoRaWAN и экспериментальное исследование его эффективности в различных сценариях развертывания сетей.

СПИСОК ЛИТЕРАТУРЫ

1. *Jovanovic B.* Internet of Things Statistics for 2024 — Taking Things Apart (online). DataProt: Cybersecurity Product Reviews, Tips & Latest News. https://dataprot.net/statistics/iot-statistics/. Accessed Aug. 2, 2024.
2. *Centenaro M., Vangelista L., Kohno R.* On the Impact of Downlink Feedback on LoRa Performance // Proc. IEEE 28th Annu. Int. Symp. on Personal, Indoor, and Mobile Radio Communications (IEEE PIMRC 2017). Montreal, QC, Canada. Oct. 8–13, 2017. P. 1–6. https://doi.org/10.1109/PIMRC.2017.8292315
3. *Casals, L., Mir B., Vidal R., Gomez C.* Modeling the Energy Performance of LoRaWAN // Sensors. 2017. V. 17. № 10. Paper No. 2364 (30 pp.). https://doi.org/10.3390/s17102364
4. *Bankov D., Khorov E., Lyakhov A.* Mathematical Model of LoRaWAN Channel Access with Capture Effect // Proc. IEEE 28th Annu. Int. Symp. on Personal, Indoor, and Mobile Radio Communications (IEEE PIMRC 2017). Montreal, QC, Canada. Oct. 8–13, 2017. P. 1–5. https://doi.org/10.1109/PIMRC.2017.8292748
5. *Федорищева А.А., Левченко П.А., Банков Д.В.* Снижение энергопотребления устройств при ограничении на рабочий цикл в сетях NB-Fi // Сб. трудов 47-й междисциплинарной школы-конференции ИППИ РАН “Информационные технологии и системы” (ИТиС 2023). Огниково, 17–21 сентября 2023. М: ИППИ РАН, 2023. С. 459–474. https://doi.org/10.53921/itas2023_459
6. *Карамышев А.Ю., Порай Е.Д., Хоров Е.М.* Оценка емкости системы сверхнадежной связи с низкими задержками с помощью аппроксимаций для многосерверных систем массового обслуживания $G/G/s$ // Пробл. передачи информ. 2024. Т. 60. № 2. С. 36–52. https://doi.org/10.31857/S0555292324020049
7. *Лихтциндер Б.Я., Привалов А.Ю.* Обобщение формул для моментов очереди при неординарном пуассоновском потоке для очередей пакетов в системах телекоммуникаций // Пробл. передачи информ. 2023. Т. 59. № 4. С. 32–37. https://doi.org/10.31857/S0555292323040046
8. *Угловский А.Ю., Мельников И.А., Алексеев И.А., Кореев А.А.* Оценка низкого уровня ошибок с помощью выборки по значимости с равномерным распределением // Пробл. передачи информ. 2023. Т. 59. № 4. С. 3–12. https://doi.org/10.31857/S0555292323040010
9. LoRaWAN® Regional Parameters RP002-1.0.4. LoRa Alliance®, Fremont, CA, USA, 2022. Available at https://resources.lora-alliance.org/technical-specifications/rp002-1-0-4-regional-parameters.

10. *Hata M.* Empirical Formula for Propagation Loss in Land Mobile Radio Services // IEEE Trans. Veh. Technol. 1980. V. 29. № 3. P. 317–325. https://doi.org/10.1109/T-VT.1980.23859
11. *To T.-H., Duda A.* Simulation of LoRa in NS-3: Improving LoRa Performance with CSMA // Proc. 2018 IEEE Int. Conf. on Communications (ICC 2018). Kansas City, MO, USA. May 20–24, 2018. P. 1–7. doi:10.1109/ICC.2018.8422800
12. SX1276-7-8-9 Datasheet. LoRa Connect™ 137 MHz to 1020 MHz Long Range Low Power Transceiver. Semtech Corp., 2016. Available at https://www.semtech.com/products/wireless-rf/lora-connect/sx1276#documentation.
13. Fedorishcheva A.A., Bankov D.V., Lyakhov A.I., Khorov E.M. Reducing Energy Consumption in LoRaWAN Networks with Duty Cycle Limitation // Probl. Inf. Transm. 2025. V. 61. № 1 (to appear).

Федорищева Анастасия Анатольевна
Банков Дмитрий Викторович
Ляхов Андрей Игоревич
Хоров Евгений Михайлович
Институт проблем передачи информации
им. А.А. Харкевича Российской академии наук, Москва
fedorishcheva@wireless.iitp.ru
bankov@wireless.iitp.ru
lyakhov@iitp.ru
khorov@wireless.iitp.ru

Поступила в редакцию
16.10.2024
После доработки
07.11.2024
Принята к публикации
13.12.2024

ПРОБЛЕМЫ ПЕРЕДАЧИ ИНФОРМАЦИИ

Том 60 2024 Вып. 4

УДК 621.391 : 004.725.5

© 2024 г. А.В. Ритерман, Д.В. Банков, А.И. Ляхов, Е.М. Хоров

ОБ ЭФФЕКТИВНОСТИ МЕТОДА ДОСТУПА К КАНАЛУ С ВЫТЕСНЕНИЕМ В СЕТЯХ Wi-Fi 8[1]

Для обеспечения надежной доставки пакетов с низкой задержкой, требуемой приложениями реального времени (англ.: real-time applications, RTAs), разрабатывается метод доступа к каналу с вытеснением, который будет определен в дополнении к стандарту Wi-Fi 8. В статье проведено исследование эффективности данного метода доступа в сети с одной RTA-станцией. Для этого разрабатывается аналитическая модель сети Wi-Fi 8, использующей метод доступа к каналу с вытеснением, с помощью которой находятся параметры метода доступа к каналу, при которых обеспечивается низкая задержка и высокая надежность доставки RTA-трафика, а эффективность использования канала станциями, передающими менее приоритетные кадры, максимальна.

Ключевые слова: приложения реального времени, доступ к каналу с вытеснением, Wi-Fi 8, IEEE 802.11bn.

DOI: 10.31857/S0555292324040041, **EDN:** QPNGPH

§ 1. Введение

Поддержка приложений реального времени (англ.: real time applications, RTA), таких как приложения виртуальной и дополненной реальности (VR/AR) и различные индустриальные приложения, является важной задачей для современных сетей Wi-Fi. Обычно RTA-трафик требует обеспечивать низкую задержку доставки пакетов (порядка 1–10 мс) с высокой надежностью (вероятность своевременной доставки пакетов порядка 99,999%) [1]. На данный момент сети Wi-Fi не всегда могут удовлетворить требования к качеству обслуживания RTA-трафика. Один из источников такой проблемы состоит в невозможности передачи каких-либо данных в то время, когда канал занят другой передачей, которая может длиться несколько миллисекунд.

Для решения данной проблемы в новом дополнении к стандарту сетей Wi-Fi, IEEE 802.11bn [2, 3], предлагается добавить механизм вытеснения (англ.: preemption). При его использовании станции Wi-Fi, получив доступ к каналу, разделяют передачу неприоритетных данных на фрагменты, между которыми предусмотрены интервалы времени, когда другие станции могут захватить канал для передачи приоритетных данных.

Данный механизм новый для сетей Wi-Fi, и в литературе практически нет исследований его эффективности. На данный момент существует исследование, в котором предложены и описаны три схемы приоритетного доступа к каналу, адаптированные для механизма вытеснения [4]. В первых двух схемах RTA-станции взводят

[1] Исследование выполнено в ИППИ РАН за счет гранта Российского научного фонда № 24-19-00816, https://rscf.ru/project/24-19-00816/.

случайные счетчики отсрочки, прежде чем передать данные с помощью механизма вытеснения. Отличие этих схем состоит в том, когда именно уменьшаются значения этих счетчиков. В первой схеме счетчик уменьшается при детектировании кадра, позволяющего использовать механизм вытеснения. Во второй схеме счетчик уменьшается на границах слотов, аналогично тому, как это происходит в стандартном методе доступа EDCA (англ.: enhanced distrtibuted channel access). В третьей схеме для каждой RTA-станции вводится вероятность получения доступа к каналу в момент времени, когда можно начать передачу данных с использованием механизма вытеснения. В работе [4] нет данных о том, позволяют ли рассмотренные подходы удовлетворить требования к качеству обслуживания RTA-трафика.

В статье рассматривается сценарий, в котором в сети есть точка доступа, передающая низкоприоритетные данные подключенным к ней станциям, и RTA-станция, передающая срочные кадры точке доступа. В статье исследуются зависимости распределения задержки доставки срочных кадров и эффективности использования канала от параметров механизма вытеснения и интенсивности RTA-трафика. Под задержкой подразумевается время, прошедшее с момента генерации RTA-кадра до получения подтверждения о его доставке, а под эффективностью использования канала – доля времени, в течение которого точка доступа передает полезные данные, прямо пропорциональная пропускной способности сети. Исследуется, какие факторы сильнее всего влияют на задержку доставки RTA-кадров. Разработана аналитическая модель метода доступа с вытеснением для случая одной станции, передающей RTA-кадры в сети.

Дальнейшее изложение построено следующим образом. В § 2 содержится дополнительная информация о механизме вытеснения. В § 3 приводятся сценарий и постановка задачи. В § 4 описывается математическая модель сети Wi-Fi с механизмом вытеснения. В § 5 приведены полученные численные результаты. В § 6 представлено заключение статьи.

§ 2. Объект исследования

В современных сетях Wi-Fi основным методом доступа к каналу является механизм случайного доступа EDCA, краткое и упрощенное описание которого приведено ниже.

Перед получением доступа к каналу станции Wi-Fi должны прослушать канал и определить, свободен ли он. Если новый кадр на передачу поступает в пустую очередь и канал свободен, то станция сразу начинает передачу, в противном случае она взводит счетчик отсрочки. Счетчик отсрочки инициализируется целым числом, которое равновероятно выбирается из полуинтервала $[0, W_r)$. Параметр W_r называется конкурентным окном, его значение зависит от количества неудачных попыток передачи текущего кадра r следующим образом:

$$W_r = \begin{cases} W_{\min}, & r = 0, \\ \min(2W_{r-1}, W_{\max}), & r > 0, \end{cases} \tag{1}$$

где $W_{\min}, W_{\max}$ – границы конкурентного окна.

Пока канал свободен, счетчик отсрочки уменьшается на единицу после каждого пустого временного слота длительностью σ. Когда канал занят, счетчик отсрочки замораживается. Как только канал оказывается свободным в течение межкадрового интервала AIFS (англ.: arbitration inter-frame space), станция возобновляет отсчет и уменьшает значение счетчика отсрочки.

Когда значение счетчика отсрочки достигает нуля, станция совершает попытку передачи кадра данных. Попытка считается успешной, если станция-отправитель получит кадр подтверждения ACK от станции-получателя через короткий межкад-

ровый интервал SIFS (англ.: short inter-frame spacing). Если станция-отправитель в течение интервала времени AckTimeout после окончания передачи не детектирует начало подтверждения, попытка считается неудачной. В случае неудачи станция увеличивает счетчик r и продолжает попытку передачи кадра данных. При успешной передаче счетчик r обнуляется, и станция переходит к передаче следующих данных при их наличии.

Станция получает доступ к каналу на время TXOP (англ.: transmission opportunity). При этом станция может передать как несколько кадров, так и один. Однако время, в течение которого станция занимает канал, включая кадр подтверждения, не должно превышать предельное значение длительности TXOP (TXOP limit).

Механизм случайного доступа EDCA определяет четыре категории доступа (англ.: access categories, ACs), для каждой из которых установлены свои значения $W_{\min}$, $W_{\max}$, AIFS и TXOP limit. Такое разделение на категории можно использовать для того, чтобы станции, передающие RTA-кадры, могли получать доступ к каналу раньше, чем станции, передающие неприоритетные данные. В частности, чтобы минимизировать вероятность коллизии между станциями, передающими RTA-кадры, и станциями, передающими кадры другого типа, можно установить параметры так, чтобы значение AIFS + $W_{\max}$ для RTA-кадров было меньше значения AIFS для остальных кадров. В то же время остается проблема, связанная с тем, что пока канал занят передачей одной станции, другая не может передавать данные и должна дождаться освобождения канала, что может привести к длительным задержкам. С одной стороны, можно ограничивать максимальную длительность передачи для неприоритетного трафика так, чтобы учесть максимальную задержку на ожидание конца передачи [5], однако такой подход приведет к снижению эффективности использования канала при передаче неприоритетного трафика, связанному с накладными расходами на получение доступа к каналу и на передачу служебных данных. Этого можно избежать, используя механизм вытеснения, который не требует ограничения максимальной длительности передачи для неприоритетного трафика, и в то же время позволяет станциям, передающим RTA-кадры, получать доступ к каналу, не дожидаясь окончания длительной передачи.

Дополнение к стандарту IEEE 802.11bn еще только разрабатывается, и подробного описания механизма вытеснения в нем нет. Поэтому приведем упрощенное описание данного механизма, сохранив наиболее важные его свойства. Будем рассматривать передачу неприоритетных данных от точки доступа, предполагая, что в ее начале происходит обмен кадрами RTS/CTS (англ.: ready to send/clear to send), чтобы избежать длительной коллизии. Для этого точка доступа начинает передачу кадром RTS, в ответ на которую станция-получатель отправляет CTS, и после получения CTS точка доступа начинает передачу данных. Данные разделяются на несколько коротких фрагментов (длиной T), между которыми предусмотрены промежутки времени длительностью XIFS, которые в данной статье равны PIFS (англ.: PCF inter-frame space, PIFS = SIFS + σ), когда канал свободен. При отсутствии приоритетного трафика точка доступа совершает передачу до конца TXOP так, чтобы в конце TXOP получить кадр с блочным подтверждением (англ.: Block ACK, BACK), подтверждающим доставку каждого фрагмента. Причем если в TXOP не укладывается целое количество фрагментов длиной T и BACK, то TXOP соответствующим образом заканчивается позже (при этом длительность TXOP L изначально устанавливается такой, чтобы при "удлинении" TXOP не превысил TXOP limit). При наличии RTA-кадров появляется возможность передать их с использованием механизма вытеснения. Для этого станция через SIFS после окончания фрагмента передает RTA-кадр, на который точка доступа передает подтверждение. Через PIFS после передачи подтверждения точка доступа продолжает передачу фрагментов. Если произошло вытеснение, передачу следующего RTA-кадра станция может выполнить только после очередного фрагмента от точки доступа. Другими словами,

RTA-станция не может с помощью механизма вытеснения передать два RTA-кадра подряд. В случае коллизии при вытеснении точка доступа теряет TXOP, и RTA-станции, чьи передачи попали в коллизию, разрешают ее методом EDCA.

Точка доступа объявляет станциям о доступе с использованием механизма вытеснения и об интервалах времени, когда можно сделать вытеснение, в заголовке первого фрагмента. Для снижения накладных расходов последующие фрагменты имеют более короткие заголовки, которые лишь содержат небольшую преамбулу, необходимую для возобновления синхронизации приемника. При этом в случае успешного вытеснения следующий фрагмент снова содержит полный заголовок, поскольку принимающая станция за время передачи RTA-данных теряет синхронизацию.

Не очевидно, как следует выбирать значение параметра механизма вытеснения T. Например, с одной стороны, значение T следует устанавливать как можно большим, чтобы минимизировать накладные расходы, связанные с межкадровыми интервалами и заголовками. С другой стороны, значение T определяет максимальное время, которое RTA-станции нужно ждать до получения возможности вытеснения. Возникает вопрос, как выбирать параметр механизма вытеснения T, удовлетворяющий требования к качеству обслуживания RTA-трафика, при которых эффективность использования канала при передаче неприоритетных данных была бы максимальна.

§ 3. Сценарий и постановка задачи

Будем рассматривать сеть Wi-Fi, состоящую из точки доступа, передающей насыщенные потоки с неприоритетными данными, станций, принимающих данные от точки доступа, и одной станции, генерирующей ненасыщенный RTA-трафик (называемой RTA-станцией). У точки доступа всегда есть данные на передачу, в то время как у RTA-станции после успешной передачи новые данные появляются через временной промежуток, распределенный экспоненциально с параметром λ [6]. Все станции находятся в зоне уверенного приема кадров друг от друга. Будем считать, что в случае одновременной передачи кадров от разных станций происходит коллизия, в результате которой ни один кадр не доставляется успешно. При этом используется достаточно надежная сигнально-кодовая конструкция [7], такая что можно пренебречь влиянием случайного шума на передачи, и в отсутствие коллизий передача данных всегда успешна. При этом для RTA-трафика требуется, чтобы Q^*-квантиль задержки доставки пакетов не превышал ограничения D^*.

Устройства используют механизм вытеснения. Значения AIFS и конкурентных окон для трафика от точки доступа и RTA-станций установлены так, чтобы после освобождения канала RTA-станция всегда получала доступ к каналу раньше точки доступа.

Определим эффективность использования канала как долю времени, в течение которого точка доступа передает части кадров с полезными данными. Эта величина прямо пропорциональна пропускной способности сети. В статье ставится следующая задача: найти длительность фрагмента T, обеспечивающую требуемую задержку и надежность доставки RTA-кадров, при которых эффективность использования канала точкой доступа S максимальна. Данную задачу можно сформулировать как следующую задачу оптимизации:

$$\begin{aligned} &\max_T S(T) \\ &\text{при условии} \quad \mathbf{P}(D(T) \leqslant D^*) \geqslant Q^*, \end{aligned} \tag{2}$$

где D – задержка доставки RTA-кадров, D^* – ограничение на эту задержку, Q^* – минимально допустимая вероятность своевременной доставки RTA-кадров.

§ 4. Математическая модель

Для решения поставленной задачи разработана аналитическая модель, описание которой приводится ниже. Данная модель построена на основе аналитических моделей, представленных в [5, 8].

Пусть интенсивность трафика от RTA-станции намного меньше интенсивности трафика от точки доступа. Тогда можно рассматривать процесс передачи RTA-станции как некоторое возмущение поверх стационарного процесса передачи точки доступа.

В описанной ниже математической модели параметры $W_{\min}^{\mathrm{AP}}$, $W_{\max}^{\mathrm{AP}}$ и $\mathrm{AIFS}_{\mathrm{AP}}$ обозначают границы конкурентного окна и AIFS для точки доступа, а параметры $W_{\min}^{\mathrm{RTA}}$, $W_{\max}^{\mathrm{RTA}}$ и $\mathrm{AIFS}_{\mathrm{RTA}}$ – для RTA-станции.

4.1. Передача точки доступа. Опишем процесс передачи точки доступа в отсутствие RTA-станции. Время в канале, аналогично [9], делится на виртуальные слоты – интервалы времени между последовательным уменьшением счетчика отсрочки. Процесс передачи точки доступа можно разделить на интервалы (далее – *интервалы обслуживания*), состоящие из отсчета отсрочки и передачи кадра. Рассмотрим один такой интервал обслуживания.

Точка доступа перед передачей отсчитывает отсрочку, среднее значение которой составляет

$$\bar{b} = \frac{W_{\min}^{\mathrm{AP}} - 1}{2}.$$

После этого точка доступа передает в течение L_{ext} – длительности TXOP, удлиненной так, чтобы передать целое число фрагментов и BACK:

$$L_{\mathrm{ext}} = T_{\mathrm{first}} + k \cdot T_{\mathrm{mid}} + T_{\mathrm{last}}, \tag{3}$$

где T_{first} – длительность интервала (далее называемого *первым интервалом*) времени между началом передачи от точки доступа и первой возможностью прерывания, T_{mid} – длительность интервала (далее – *средний интервал*) времени между моментами, когда возможно прерывание, T_{last} – длительность интервала (далее – *последний интервал*) времени между концом последнего из средних интервалов и концом TXOP, k – максимальное количество интервалов длиной T_{mid}, которые могут уложиться в L_{ext} так, чтобы в начале успел произойти обмен кадрами RTS/CTS, а в конце точка доступа получила BACK. Длительности интервалов равняются

$$T_{\mathrm{first}} = T_{\mathrm{RTS}} + \mathrm{SIFS} + T_{\mathrm{CTS}} + \mathrm{SIFS} + T + \mathrm{SIFS},$$
$$T_{\mathrm{mid}} = \sigma + T + \mathrm{SIFS},$$
$$T_{\mathrm{last}} = \sigma + T + \mathrm{SIFS} + \mathrm{BACK},$$

где T_{RTS} и T_{CTS} – длительности кадров RTS и CTS соответственно. Значение k находится из системы

$$\begin{cases} k = \left\lceil \dfrac{L - T_{\mathrm{first}} - T_{\mathrm{last}}}{T_{\mathrm{mid}}} \right\rceil, & L - T_{\mathrm{first}} - T_{\mathrm{last}} > 0, \\ k = 0, & L - T_{\mathrm{first}} - T_{\mathrm{last}} \leqslant 0. \end{cases} \tag{4}$$

После передачи точка доступа ждет $\mathrm{AIFS}_{\mathrm{AP}}$. Таким образом, средняя длительность интервала обслуживания равна

$$L_{\mathrm{period}} = \bar{b} \cdot \sigma + L_{\mathrm{ext}} + \mathrm{AIFS}_{\mathrm{AP}}. \tag{5}$$

4.2. Распределение задержки. Рассмотрим передачу RTA-кадра. Распределение задержки RTA-кадра зависит от состояния канала в момент его генерации. Можно

выделить следующие возможные события: генерация кадра в пустом слоте, в первом интервале, в одном из средних и в последнем. Вероятности этих событий обозначим через p_{idle}, p_{first}, p_{mid}, p_{last} соответственно. Функции распределения задержки, соответствующие этим событиям, обозначим через $F_{\text{idle}}(t)$, $F_{\text{first}}(t)$, $F_{\text{mid}}(t)$, $F_{\text{last}}(t)$.

Итоговая функция распределения задержки имеет вид

$$F(t) = p_{\text{idle}} \cdot F_{\text{idle}}(t) + p_{\text{first}} \cdot F_{\text{first}}(t) + p_{\text{mid}} \cdot F_{\text{mid}}(t) + p_{\text{last}} \cdot F_{\text{last}}(t). \tag{6}$$

Рассмотрим каждый из четырех случаев подробнее.

Генерация в пустом слоте. Согласно сценарию значение AIFS для RTA-станции меньше, чем для точки доступа, поэтому после освобождения канала RTA-станция начинает отсчет отсрочки на

$$\Delta_{\text{AIFS}} = \frac{\text{AIFS}_{\text{AP}} - \text{AIFS}_{\text{RTA}}}{\sigma}$$

пустых слотов раньше, чем точка доступа. Учитывая это, вероятность p_{idle} может быть найдена как доля времени, когда канал свободен:

$$p_{\text{idle}} = \frac{(\bar{b} + \Delta_{\text{AIFS}}) \cdot \sigma}{L_{\text{period}}}. \tag{7}$$

Рассмотрим произвольный пустой слот с точки зрения RTA-станции. Вероятность того, что точка доступа начнет передавать на границе рассматриваемого слота, равна среднему количеству передач за интервал обслуживания, деленному на среднее количество слотов в интервале обслуживания:

$$\tau = \frac{1}{\bar{b} + \Delta_{\text{AIFS}} + 1}. \tag{8}$$

При генерации RTA-кадра во время пустого слота RTA-станция начинает передачу кадра на границе слота, и возможно несколько случаев:

- Если точка доступа не передает в следующем слоте (с вероятностью $1 - \tau$), происходит успешная передача RTA-станции. Соответствующую функцию распределения задержки обозначим через $F^s_{\text{idle}}(t)$;
- Если точка доступа передает в следующем слоте (с вероятностью τ), происходит коллизия между кадрами RTA-станции и точки доступа. Соответствующую функцию распределения задержки обозначим через $F^c_{\text{idle}}(t)$.

Результирующая функция распределения задержки $F_{\text{idle}}(t)$ имеет вид

$$F_{\text{idle}}(t) = (1 - \tau) \cdot F^s_{\text{idle}}(t) + \tau \cdot F^c_{\text{idle}}(t). \tag{9}$$

Найдем входящие в нее вероятности и функции распределения задержек.

Успешная передача RTA-станции. Задержка передачи кадра в случае успешной передачи складывается из времени ожидания от момента генерации кадра до начала следующего слота и последующей передачи кадра. Чтобы найти ее функцию распределения, рассмотрим задачу в общем случае, когда RTA-кадр генерируется в течение интервала времени длительностью g, а после окончания интервала передается за время s. Найдем вероятность того, что задержка RTA-кадра, которую обозначим через D, будет не больше чем t. Пусть x – время от начала интервала g до момента генерации RTA-кадра, причем x – экспоненциально распределенная на отрезке $[0, g]$ случайная величина.

Искомая вероятность равна

$$\mathbf{P}(D < t \,|\, x < g) = \mathbf{P}(g - x + s < t \,|\, x < g) = \frac{\mathbf{P}(g + s - t < x < g)}{\mathbf{P}(x < g)}.$$

Введем функцию $\Phi(t, g, s)$ для упрощения выкладок:

$$\Phi(t, g, s) = \frac{\mathbf{P}(g + s - t < x < g)}{\mathbf{P}(x < g)} = \begin{cases} 0, & t < s, \\ \dfrac{e^{-\lambda(g+s-t)} - e^{-\lambda g}}{1 - e^{-\lambda g}}, & s \leqslant t < g + s, \\ 1, & t \geqslant g + s. \end{cases} \tag{10}$$

Используя (10), функцию распределения задержки $F^s_{\text{idle}}(t)$ можно записать как

$$F^s_{\text{idle}}(t) = \Phi(t, \sigma, T_r), \tag{11}$$

где

$$T_r = \text{DATA} + \text{SIFS} + \text{ACK}$$

– время успешной передачи RTA-станции (DATA – длительность кадра данных RTA-станции, ACK – длительность подтверждения).

Коллизия между RTA-станцией и точкой доступа. С вероятностью τ при попытке передачи RTA-кадра возникает коллизия с точкой доступа. При коллизии с точкой доступа RTA-станция взведет счетчик отсрочки b, отсчет которого начнется после интервала времени, равного длительности коллизии

$$T_c = \max\{\text{RTS}, \text{DATA}\} + \text{ACKTIMEOUT} + \text{AIFS}_{\text{RTA}},$$

где ACKTIMEOUT – время, в течение которого станция ожидает получения подтверждения. Затем через время $b \cdot \sigma$ RTA-станция совершит повторную попытку передачи. Функция распределения задержки при конкретном значении b будет равна

$$\Phi(t, \sigma, T_c + b \cdot \sigma + T_r).$$

Согласно сценарию у RTA-станции установлены приоритетные значения AIFS и конкурентных окон (таблица 1), поэтому повторная коллизия с точкой доступа невозможна. Учитывая то, что после коллизии счетчик отсрочки выбирается равновероятно из полуинтервала $[0, W_1^{\text{RTA}})$, где

$$W_1^{\text{RTA}} = \min\{2W_{\min}^{\text{RTA}}, W_{\max}^{\text{RTA}}\},$$

получим функцию распределения задержки $F^c_{\text{idle}}(t)$ в случае коллизии RTA-станции с точкой доступа:

$$F^c_{\text{idle}}(t) = \frac{1}{W_1^{\text{RTA}}} \sum_{b=0}^{W_1^{\text{RTA}}-1} \Phi(t, \sigma, T_c + b \cdot \sigma + T_r). \tag{12}$$

Генерация в первом и средних интервалах Вероятности генерации RTA-кадра в первом интервале и в одном из средних интервалов могут быть приближенно найдены как

$$p_{\text{first}} = \frac{T_{\text{first}}}{L_{\text{period}}}, \tag{13}$$

$$p_{\text{mid}} = \frac{k \cdot T_{\text{mid}}}{L_{\text{period}}}. \tag{14}$$

Кадр, сгенерированный в одном из таких интервалов, будет успешно передан с использованием механизма вытеснения при первой такой возможности. Задержка при успешной передаче складывается из времени ожидания, отсчитанного от момента генерации кадра до конца интервала, т.е. до возможности сделать вытеснение, и длительности T_r. Длительности T_{first} и T_{mid} различны, поэтому функция распределения задержки при генерации RTA-кадра в первом интервале находится как

$$F_{\text{first}}(t) = \Phi(t, T_{\text{first}}, T_r), \tag{15}$$

а при генерации RTA-кадра в одном из средних интервалов – как

$$F_{\text{mid}}(t) = \Phi(t, T_{\text{mid}}, T_r) \tag{16}$$

соответственно.

Генерация в последнем интервале. Вероятность генерации RTA-кадра в последнем интервале равна

$$p_{\text{last}} = \frac{T_{\text{last}}}{L_{\text{period}}}. \tag{17}$$

При генерации кадра в последнем слоте RTA-станция взводит счетчик отсрочки, который равновероятно выбирается из полуинтервала $[0, W_{\min}^{\text{RTA}})$, ждет окончания TXOP, после чего через время AIFS_{RTA} начинает отсчет отсрочки, который достигнет значения 0 спустя время $b \cdot \sigma$, и затем успешно передает RTA-кадр.

Задержка передачи RTA-кадра будет складываться из времени, прошедшего от момента генерации кадра до конца TXOP, AIFS_{RTA}, времени отсчета отсрочки и T_r. Согласно сценарию у RTA-станции установлены приоритетные значения AIFS и конкурентных окон, поэтому коллизия с точкой доступа невозможна. По аналогии с формулой (12) функция распределения задержки при конкретном значении счетчика отсрочки будет равна

$$\Phi(t, T_{\text{last}} + \text{AIFS}_{\text{RTA}}, b \cdot \sigma + T_r).$$

Учитывая, что счетчик отсрочки выбирается равновероятно из полуинтервала $[0, W_{\min}^{\text{RTA}})$, получим функцию распределения задержки $F_{\text{last}}(t)$:

$$F_{\text{last}}(t) = \frac{1}{W_{\min}^{\text{RTA}}} \sum_{b=0}^{W_{\min}^{\text{RTA}}-1} \Phi(t, T_{\text{last}} + \text{AIFS}_{\text{RTA}}, b \cdot \sigma + T_r), \tag{18}$$

4.3. Эффективность использования канала. Оценим эффективность использования канала точкой доступа.

Эффективность использования канала при отсутствии передачи RTA-кадра за интервал обслуживания обозначим через S_0. За это время (L_{period}) точка доступа передаст $k + 2$ фрагмента длиной T: по одному в первом и последнем интервалах, k фрагментов в средних интервалах. Учтем, что каждый фрагмент имеет заголовок, поэтому эффективность использования канала будет равна

$$S_0 = \frac{T \cdot (k + 2) - t_0(k)}{L_{\text{period}}}, \tag{19}$$

где $t_0(k)$ – суммарная длина всех заголовков в этом случае.

Учтем наличие передач RTA-кадров. Найдем среднее количество сгенерированных за интервал обслуживания RTA-кадров ($\bar{n}$) как отношение длительности ин-

тервала к сумме средней задержки передачи RTA-кадра ($\bar{D}$) и среднего времени ожидания перед генерацией нового RTA-кадра:

$$\bar{n} = \frac{L_{\text{period}}}{\bar{D} + \dfrac{1}{\lambda}}. \tag{20}$$

Величину $\bar{D}$ можно найти, используя распределение задержки $F(t)$, найденное в (6):

$$\bar{D} = \int_0^\infty x\, dF(x). \tag{21}$$

Если RTA-кадр сгенерирован во время передачи точки доступа длительностью L_{ext}, то он будет передан с использованием механизма вытеснения. В противном случае, при генерации во время интервала длительностью

$$\text{AIFS}_{\text{AP}} + \bar{b} \cdot \sigma$$

(т.е. $L_{\text{period}} - L_{\text{ext}}$), когда канал свободен, кадр будет передан без использования механизма вытеснения. Для достижения высокой пропускной способности точке доступа целесообразно устанавливать как можно большую длительность передачи. При этом будет выполняться

$$L_{\text{ext}} \gg \text{AIFS}_{\text{AP}} + \bar{b} \cdot \sigma,$$

и большинство RTA-кадров будут передаваться с использованием механизма вытеснения, поэтому оценим эффективность использования канала следующим образом:

$$S \approx S_0 \cdot \left(1 - \frac{\bar{n} \cdot (T_r + \text{PIFS})}{L_{\text{period}}}\right) = S_0 \cdot \left(1 - \frac{T_r + \text{PIFS}}{\dfrac{1}{\lambda} + \bar{D}}\right), \tag{22}$$

где эффективность S_0 в случае передачи точки доступа без RTA-станции умножается на масштабирующий множитель, равный доле времени, в течение которого точка доступа передает свои кадры. Эта доля времени оценивается с учетом того, что за интервал обслуживания L_{period} RTA-станция генерирует в среднем $\bar{n}$ кадров, для передачи каждого из которых отнимает у точки доступа $T_r + \text{PIFS}$ канального времени.

§ 5. Численные результаты

В данном параграфе приводятся численные результаты исследования эффективности механизма вытеснения. Сначала в п. 5.1 показаны результаты оценки точности модели. Далее в п. 5.2 показано влияние параметра T механизма вытеснения на значение Q-квантиля задержки и эффективности использования канала. В п. 5.3 найдены оптимальные параметры механизма вытеснения.

В табл. 1 приведены значения временных интервалов и параметров EDCA, используемых при получении численных результатов.

5.1. Оценка точности модели. Для оценки точности аналитической модели была разработана имитационная модель. В отличие от аналитической модели имитационная модель не использует предположения о стационарности вероятности передачи точки доступа, а подробно моделирует процесс отсчета отсрочки. Также в имитационной модели не предполагается, что процесс передачи RTA-станции не влияет на передачи точки доступа, и учитываются изменения конкурентного окна точки

Таблица 1

Значения временных интервалов и параметров EDCA

RTS = 44 мкс	PIFS = SIFS + σ	$W_{\min}^{\text{RTA}} = 4$	$L = 4$ мс
CTS = 44 мкс	$\text{AIFS}_{\text{RTA}} = \text{SIFS} + 2\sigma$	$W_{\max}^{\text{RTA}} = 8$	DATA = 300 мкс
SIFS = 16 мкс	$\text{AIFS}_{\text{AP}} = \text{SIFS} + 10\sigma$	$W_{\min}^{\text{AP}} = 16$	
$\sigma = 9$ мкс	ACKTIMEOUT = 45 мкс	$W_{\max}^{\text{AP}} = 1024$	

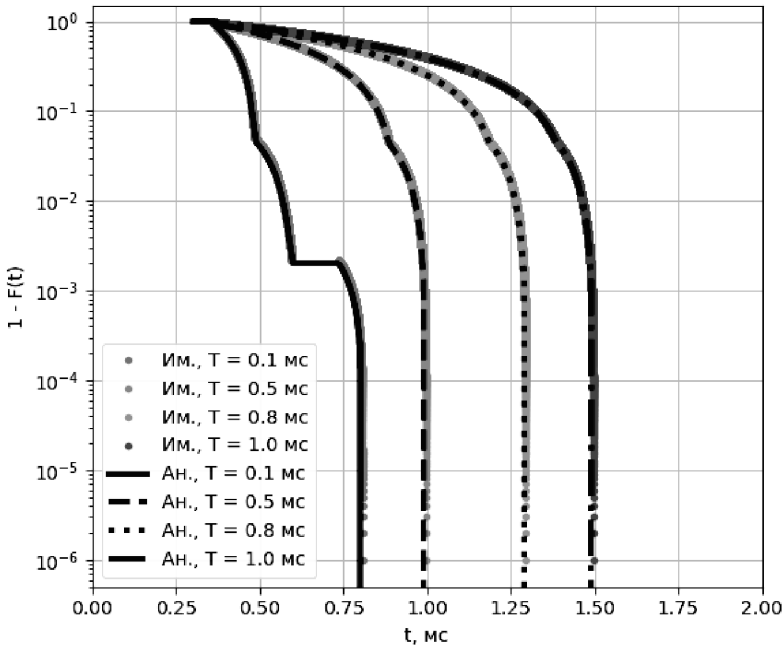


Рис. 1. Дополненная функция распределения задержки, $\lambda = 50\,\text{с}^{-1}$

доступа при коллизиях. На рис. 1 и 2 представлены графики зависимости дополненной функции распределения задержки от времени и эффективности использования канала от интенсивности RTA-трафика. Как видно из графиков, результаты аналитической модели (кривые “Ан.”) отличаются от результатов имитационной (кривые “Им.”) не более чем на 2% для дополненной функции распределения задержки, и не более чем на 1% для эффективности использования канала при малой интенсивности RTA-трафика. При большей интенсивности RTA-трафика значение эффективности для математической и имитационной моделей незначительно возрастает из-за роста вероятностей возникновения коллизии с точкой доступа и передачи RTA-кадра без использования механизма вытеснения, однако оно по-прежнему меньше 1%. Разработанная аналитическая модель позволяет с высокой точностью оценивать квантили задержки порядка 0,99999, поэтому ее можно использовать для выбора параметров механизма вытеснения для RTA-трафика. На графиках дополненной функции распределения задержки от времени видны особые точки, в которых меняется характер зависимости. По абсциссе они соответствуют максимальной задержке доставки RTA-кадра, сгенерированного в пустом слоте. Большие задержки возможны при передаче RTA-кадра с вытеснением или при коллизии. На рис. 1 для $T = 0{,}1$ мс виден горизонтальный участок, наличие которого обусловлено тем, что для такой длительности фрагмента максимальная задержка при передаче с вы-

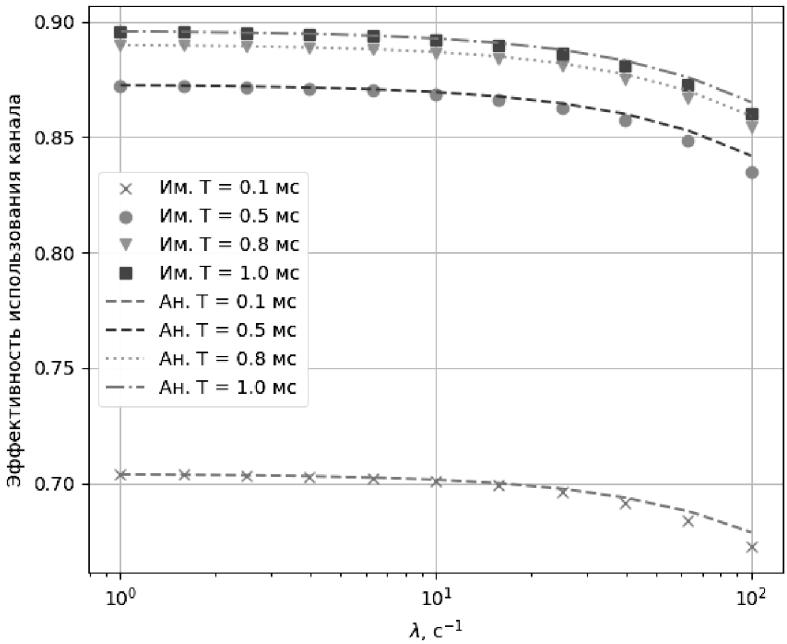


Рис. 2. Зависимость эффективности использования канала от интенсивности трафика

теснением оказывается меньше, чем минимальная задержка в случае коллизии. Таким образом, при малых значениях T квантиль задержки порядка 0,999–0,99999 определяется вероятностью коллизии RTA-станции с точкой доступа, а при больших T – вероятностью генерации RTA-кадра в первом интервале. Из зависимости эффективности использования канала от интенсивности видно, что эффективность использования канала падает с увеличением λ и растет с увеличением T.

5.2. Влияние параметров механизма вытеснения. Рассмотрим, как выбор параметров механизма вытеснения влияет на значения 0,99999-квантиля задержки и эффективности использования канала. Значение 0,99999-квантиля задержки практически не изменяется от интенсивности трафика и совпадает с показанным на рис. 1. На рис. 2 показана зависимость эффективности использования канала от интенсивности трафика. Эффективность возрастает с ростом T, что объясняется снижением накладных расходов на передачу. Таким образом, для максимизации эффективности использования канала точкой доступа и выполнения ограничения на квантиль задержки доставки RTA-кадров нужно установить T наибольшим, но так, чтобы при нем выполнялось ограничение на квантиль задержки.

5.3. Выбор параметра T. Рассмотрим, как выбирать значение параметра T. На рис. 3 представлены зависимости Q-квантиля задержки от длительности фрагмента T. Данные зависимости показывают, что при значениях T, превышающих некоторое значение T^*, квантили задержки близки к максимальной задержке при передаче с использованием механизма вытеснения RTA-кадра, сгенерированного в первом интервале, равной

$$D_{\text{first}}^{\max} = RTS + \text{SIFS} + \text{CTS} + \text{SIFS} + T + \text{SIFS} + T_r + \text{AIFS}_{\text{RTA}}, \tag{23}$$

причем квантили порядка 99,9–99,99% практически совпадают с данной величиной, а квантиль 99% ниже максимальной задержки примерно на 0,05 мс.

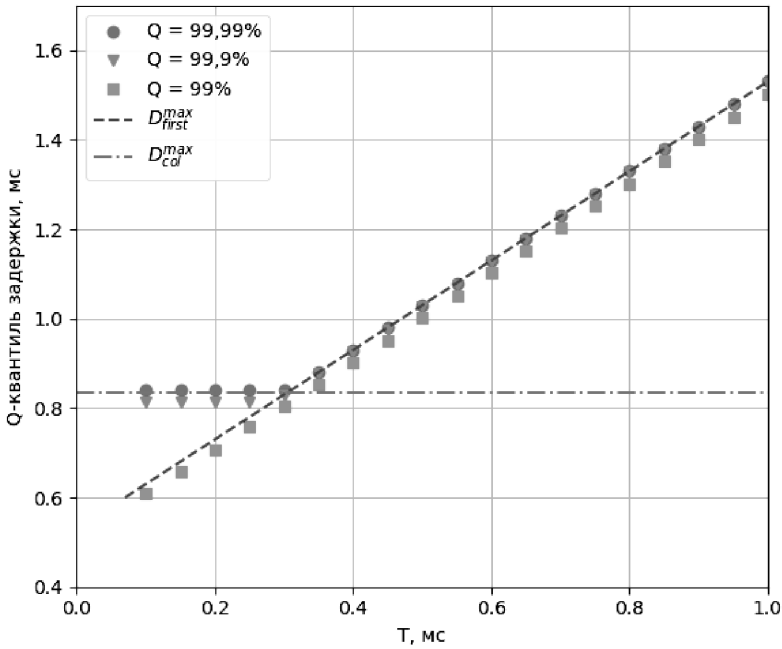


Рис. 3. Зависимость Q-квантилей от длительности фрагмента

При $T < T^*$ квантиль задержки порядка 99% также близок к $D_{\text{first}}^{\max}$, а квантили задержки порядка 0,999–0,9999 определяются максимальной задержкой, соответствующей случаю коллизии кадров RTA-станции и точки доступа, равной

$$D_{\text{col}}^{\max} = \sigma + T_c + \left(W_1^{\text{RTA}} - 1\right) \cdot \sigma + T_r + \text{AIFS}_{\text{RTA}}. \tag{24}$$

Длину фрагмента T^*, начиная с которого меняется зависимость, можно найти из уравнения

$$D_{\text{first}}^{\max} = D_{\text{col}}^{\max}.$$

В рассматриваемом случае $T^* \approx 0{,}3$ мс.

Таким образом, при $D^* > D_{\text{col}}^{\max}$ и $Q \geqslant 99\%$ параметр T можно выбирать, исходя из уравнения

$$D_{\text{first}}^{\max}(T) = D^*.$$

В противном случае, если $D^* < D_{\text{col}}^{\max}$, нужно сравнить $1 - Q$ с вероятностью коллизии передач RTA-станции и точки доступа. Если

$$p_{\text{idle}} \cdot \tau < 1 - Q,$$

параметр T можно выбрать исходя из того же уравнения. Если

$$p_{\text{idle}} \cdot \tau \geqslant 1 - Q,$$

то квантиль задержки близок к значению $D_{\text{col}}^{\max}$, и меньшую задержку обеспечить нельзя.

§ 6. Заключение

В данной статье исследован механизм вытеснения, который планируется использовать для обслуживания приложений реального времени (RTA) в сетях Wi-Fi 8. При его использовании передачи неприоритетных данных разделяются на фрагменты, между которыми предусмотрены интервалы времени, когда можно передать приоритетные данные, даже если доступ к каналу был получен другим устройством.

В статье представлена аналитическая модель передачи данных при использовании метода доступа к каналу с вытеснением. Даны рекомендации для выбора параметров метода доступа для доставки приоритетных данных с требуемым квантилем задержки доставки RTA-кадров и максимальной эффективностью использования канала точкой доступа.

В расширенной версии данной статьи [10] также проведено сравнение метода доступа с вытеснением и стандартного метода доступа к каналу EDCA и показано, что при оптимальных параметрах обоих методов метод доступа с вытеснением обеспечивает существенно большую пропускную способность сети.

В дальнейшем планируется расширить разработанную модель для сети с неидеальным каналом и с большим количеством RTA-станций.

СПИСОК ЛИТЕРАТУРЫ

1. *Карамышев А.Ю., Порай Е.Д., Хоров Е.М.* Оценка емкости системы сверхнадежной связи с низкими задержками с помощью аппроксимаций для многосерверных систем массового обслуживания $G/G/s$ // Пробл. передачи информ. 2024. Т. 60. № 2. С. 36–52. https://doi.org/10.31857/S0555292324020049
2. *Fang J., Akhmetov D., Park M., Cariou L., Stacey R.* Preemption for Low Latency Application. IEEE 802.11-23/0092r0. Mar. 13, 2023. https://mentor.ieee.org/802.11/dcn/23/11-23-0092-00-0uhr-preemption.pptx.
3. *Ruy K., Chu L., Wang H., Cao R., Zhang H.* Low Latency Support in UHR. IEEE 802.11-23/0018r1. Feb. 5, 2023. https://mentor.ieee.org/802.11/dcn/23/11-23-0018-01-0uhr-low-latency-support-in-uhr.pptx.
4. *Moon J., Kim R.Y.* Preemptive Channel Access Scheme for Next Generation Wi-Fi // Proc. 2024 IEEE Int. Conf. on Big Data and Smart Computing (IEEE BigComp 2024). Bangkok, Thailand. Feb. 18–21, 2024. P. 131–135. https://doi.org/10.1109/BigComp60711.2024.00029
5. *Bankov D., Chemrov K., Khorov E.* Tuning Channel Access to Enable Real-Time Applications in Wi-Fi 7 // 12th Int. Congr. on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT 2020). Brno, Czech Republic. Oct. 5–7, 2020. P. 20–25. https://doi.org/10.1109/ICUMT51630.2020.9222409
6. *Лихтциндер Б.Я., Привалов А.Ю.* Обобщение формул для моментов очереди при неординарном пуассоновском потоке для очередей пакетов в системах телекоммуникаций // Пробл. передачи информ. 2023. Т. 59. № 4. С. 32–37. https://doi.org/10.31857/S0555292323040046
7. *Угловский А.Ю., Мельников И.А., Алексеев И.А., Кореев А.А.* Оценка низкого уровня ошибок с помощью выборки по значимости с равномерным распределением // Пробл. передачи информ. 2023. Т. 59. № 4. С. 3–12. https://doi.org/10.31857/S0555292323040010
8. Bankov D.V., Khorov E.M., Lyakhov A.I., Sandal M.L. Approach to Real-Time Communications in Wi-Fi Networks // J. Commun. Technol. Electron. 2019. V. 64. P. 880–889. https://doi.org/10.1134/S1064226919080205
9. *Vishnevsky V., Lyakhov A.* 802.11 LANs: Saturation Throughput in the Presence of Noise // Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications. Proc. 2nd Int. IFIP-TC6 Network-

ing Conf. (NETWORKING 2002). Pisa, Italy. May 19–24, 2002. Lect. Notes Comp. Sci. V. 2345. Berlin: Springer, 2002. P. 1008–1019. https://doi.org/10.1007/3-540-47906-6_82

10. *Riterman A.V., Bankov D.V., Lyakhov A.I., Khorov E.M.* Modeling of Channel Access with Preemption in Wi-Fi 8 Networks // Probl. Inf. Transm. 2024. V. 60. № 4 (to appear). https://doi.org/10.1134/S0032946024040045

Ритерман Алиса Вадимовна
Банков Дмитрий Викторович
Ляхов Андрей Игоревич
Хоров Евгений Михайлович
Институт проблем передачи информации
им. А.А. Харкевича Российской академии наук, Москва
riterman@wireless.iitp.ru
bankov@iitp.ru
lyakhov@iitp.ru
khorov@wireless.iitp.ru

Поступила в редакцию
19.11.2024
После доработки
19.11.2024
Принята к публикации
10.12.2024

ПРОБЛЕМЫ ПЕРЕДАЧИ ИНФОРМАЦИИ

Том 60 **2024** **Вып. 4**

УДК 616-073.756.8 : 519.6

© 2024 г. **Д.В. Полевой, Д.Д. Казимиров, М.В. Чукалина, Д.П. Николаев**

ТРАНСПОНИРОВАНИЕ СУММИРУЮЩИХ АЛГОРИТМОВ С СОХРАНЕНИЕМ ВЫЧИСЛИТЕЛЬНОЙ СЛОЖНОСТИ ПРИ ПОМОЩИ ГРАФОВОГО ПРЕДСТАВЛЕНИЯ ВЫЧИСЛЕНИЙ

Представлен новый метод транспонирования суммирующих алгоритмов с использованием их графового представления, обеспечивающий большую гибкость по сравнению с предыдущими подходами, основанными на явном матричном представлении соответствующего суммирующего оператора. Применение нашего метода продемонстрировано на примере транспонирования нескольких алгоритмов быстрого преобразования Хафа. Важно отметить, что наш подход сохраняет асимптотическую вычислительную сложность исходного алгоритма. Последнее свойство очень важно для приложений в компьютерной томографии.

Ключевые слова: суммирующие алгоритмы, транспонированный оператор, быстрое преобразование Хафа, паттерны, компьютерная томография, оператор прямого проецирования, оператор обратного проецирования.

DOI: 10.31857/S0555292324040053, **EDN:** RGCBHA

§ 1. Введение

В широком спектре прикладных задач хорошо известны быстрые алгоритмы вычисления суммирующих операторов. Однако быстрое вычисление транспонированных (или сопряженных) операторов часто оказывается не менее, а может быть, и более важным. Примерами операторов, для которых требуется транспонирование, являются преобразование Хафа и оператор прямого проецирования, занимающий центральное место в компьютерной томографии (КТ). В следующем параграфе мы подробнее рассмотрим эти примеры.

Ранее в литературе был предложен метод транспонирования быстрых суммирующих алгоритмов [1]. Этот метод сохраняет асимптотическую вычислительную сложность исходного алгоритма, позволяя вычислять транспонированный оператор столь же эффективно, что и прямой. Предложенный универсальный метод транспонирования основан на разложении матрицы прямого оператора в произведение более простых матриц. Однако получение такого разложения матрицы часто является отдельной, достаточно сложной задачей, которую необходимо решить до применения метода транспонирования. Согласно ранее предложенному методу транспонирование суммирующего алгоритма предлагается выполнять только после разложения матрицы оператора. В данной статье мы представляем альтернативный метод транспонирования суммирующих алгоритмов, который не требует явной матричной записи суммирующего оператора. Вместо этого мы предлагаем интерпретацию на основе ориентированных ациклических графов (directed acyclic graph, или DAG), что позволяет облегчить процесс “ручного” транспонирования.

§ 2. Прямые и транспонированные суммирующие операторы и алгоритмы: примеры

2.1. Быстрое преобразование Хафа и быстрое транспонированное преобразование Хафа. Известным примером суммирующего оператора является преобразование Хафа (ПХ), которое также называют дискретным преобразованием Радона (ДПР). Преобразование Хафа – мощный метод в области обработки изображений и компьютерного зрения. Обычно его рассматривают как метод робастной оценки параметров одной или нескольких прямых на дискретном изображении путем подсчета количества точек, лежащих на каждой прямой из некоторого множества параметризованных прямых. Метод назван в честь Пола Хафа, который впервые представил его в 1959 году как средство идентификации прямолинейных треков в экспериментах с пузырьковой камерой [2]. По своей сути, ПХ накапливает "голоса" вдоль дискретных параметризованных прямых. Накопленное значение каждой прямой указывает на вероятность того, что она пдействительно присутствует на изображении, при этом бóльшие значения указывают на бóльшую вероятность присутствия прямой. Хотя ПХ чаще всего используется для обнаружения прямых линий или отрезков на изображениях [3–6], его применение выходит далеко за эти рамки [7]. Со временем ПХ успешно применялось в различных областях, включая бинаризацию изображений [8], сегментирование [9, 10], компьютерную томографию [1, 11] и др.

Быстрые алгоритмы для вычисления преобразования Хафа, обычно называемые алгоритмами быстрого преобразования Хафа (БПХ, или FHT), получили широкое развитие. Среди них можно отметить алгоритм Брейди – Ёна [12], ставший де-факто стандартным, и алгоритм $FHT2DT$ [13, 14]. Эти алгоритмы позволяют быстро вычислить преобразование Хафа для двумерного полутонового изображения размера $w \times h$, снижая вычислительную сложность с $\Theta(w^2 h)$, которая требуется при наивном подходе – при суммировании значений пикселей вдоль каждой дискретной прямой на изображении, до $\Theta(wh \log_2 w)$. Последнее обстоятельство позволило использовать БПХ в системах с жесткими аппаратными ограничениями и требованием работы в реальном времени [15–19].

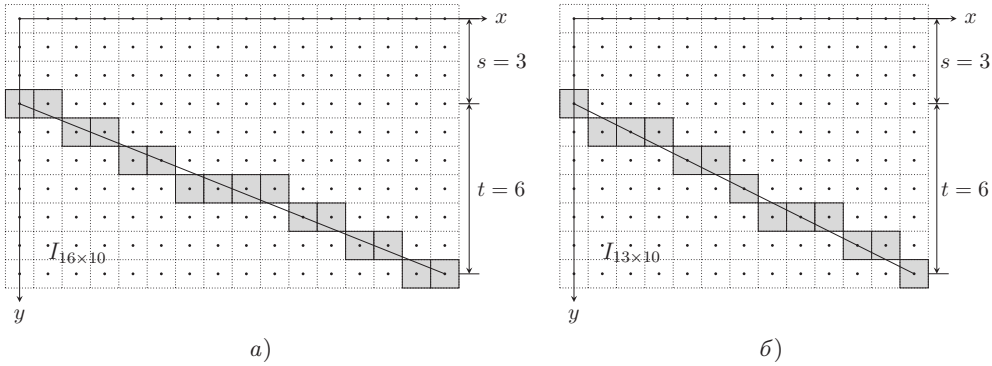
Упомянутые алгоритмы БПХ [12–14] аппроксимируют непрерывные прямые на изображении с помощью паттернов – в частности, дискретно-непрерывных прямых, совпадающих с исходными идеальными прямыми в граничных точках (см. рис. 1). В данном контексте под дискретной непрерывностью понимается отсутствие скачков в паттерне с величиной более 2 пикселей. Следует обратить внимание на то, что разные алгоритмы БПХ используют различные наборы паттернов для аппроксимации прямых линий в области изображения: в алгоритме Брейди – Ёна используются так называемые диадические паттерны длины, равной степени двойки, а в $FHT2DT$ – паттерны произвольной длины, совпадающие с диадическими только при ширине изображения, равной степени двойки.

Без ограничения общности мы будем рассматривать неубывающие прямые с наклоном в диапазоне $[0, 1]$ в системе координат изображения, называемые преимущественно горизонтальными прямыми. Преимущественно вертикальные прямые определяются аналогичным образом и получаются из преимущественно горизонтальных прямых путем отражения относительно вертикальной оси. Для описания БПХ-паттернов обычно используется st-параметризация. Пусть

$$p = p(t, s) = \{(x,\, p(t, s)(x)) \mid x \in \mathbb{Z}_{0,w-1}\}$$

представляет собой паттерн с параметрами

$$(t, s) \in \mathbb{Z}_{0,w-1} \times \mathbb{Z}_{0,h-1},$$



а) *б*)

Рис. 1. Примеры паттернов, аппроксимирующих прямые линии в области изображения: *а*) Диадический паттерн $p_{BY}(6,3)$, используемый в алгоритме Брейди – Ёна. Здесь он служит для аппроксимации прямой $l(6,3)$ в пределах $\boldsymbol{I}_{16\times10}$. По своей конструкции диадические паттерны имеют длину, равную степени двойки; *б*) $FHT2DT$-паттерн $p_{DT}(6,3)$, аппроксимирующий прямую $l(6,3)$ в пределах $\boldsymbol{I}_{13\times10}$. Этот тип паттернов введен в алгоритме $FHT2DT$. Паттерны $FHT2DT$ могут быть произвольной длины

который, согласно конструкции конкретного алгоритма БПХ, аппроксимирует преимущественно горизонтальную прямую

$$l = l(t,s) = \left\{ \left(x, l(t,s)(x) = \left(s + \frac{t}{w-1} x \right) \bmod h \right) \;\middle|\; x \in \mathbb{Z}_{0,w-1} \right\}$$

в области изображения $\boldsymbol{I} = \boldsymbol{I}_{w\times h}$, где $w \times h$ обозначает размер изображения. По определению st-параметризации $s = p(0)$, а $p(w-1) = (t+s) \bmod h$ (рис. 1).

При заданном наборе паттернов

$$\mathcal{P} = \{p(t,s) \mid (t,s) \in \mathbb{Z}_{0,w-1} \times \mathbb{Z}_{0,h-1}\},$$

зависящем от конструкции конкретного БПХ-алгоритма, БПХ $\mathcal{H}$ линейно отображает изображение $\boldsymbol{I} = \boldsymbol{I}_{w\times h} \in \mathcal{I}_{w\times h}$ в изображение $\mathcal{H}\boldsymbol{I} \in \mathcal{I}_{w\times h}$, что через st-параметризацию выражается следующим образом:

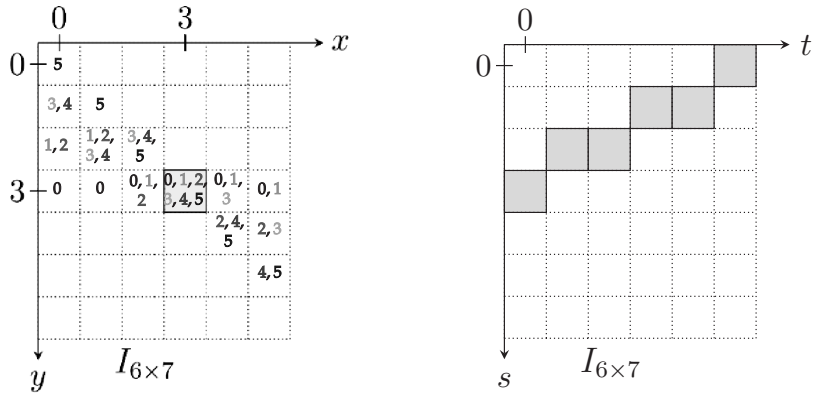
$$\mathcal{H}\colon \mathcal{I}_{w\times h} \to \mathcal{I}_{w\times h}, \quad \mathcal{H}\boldsymbol{I}(t,s) = \sum_{(x,y)\in p(t,s)} \boldsymbol{I}(x,y).$$

Здесь $\boldsymbol{I}(x,y)$ – значение пикселя с координатами $(x,y) \in \mathbb{Z}_{0,w-1} \times \mathbb{Z}_{0,h-1}$ изображения $\boldsymbol{I} \in \mathcal{I}_{w\times h}$. Через $\mathcal{I}_{w\times h}$ обозначено евклидово пространство всех двумерных одноканальных изображений размера $w \times h$ с естественным скалярным произведением $(\cdot\,,\cdot)_{\mathcal{I}_{w\times h}}$: для двух изображений $\boldsymbol{I}_1, \boldsymbol{I}_2 \in \mathcal{I}_{w\times h}$ имеем

$$(\boldsymbol{I}_1, \boldsymbol{I}_2)_{\mathcal{I}_{w\times h}} = \sum_{i\in\mathbb{Z}_{0,w-1}} \sum_{j\in\mathbb{Z}_{0,h-1}} \boldsymbol{I}_1(i,j) \cdot \boldsymbol{I}_2(i,j).$$

Полужирный шрифт $\boldsymbol{I}$ используется здесь для того, чтобы подчеркнуть, что изображение I рассматривается как элемент $\boldsymbol{I}$ евклидова векторного пространства $\mathcal{I}_{w\times h}$. Начиная с этого момента, мы будем обозначать одним символом как линейный оператор, так и его матрицу в стандартном базисе

$$\{\boldsymbol{e}_{ij} \mid (i,j) \in \mathbb{Z}_{0,w-1} \times \mathbb{Z}_{0,h-1}\}, \quad \boldsymbol{e}_{ij}(k,l) = \delta_i^k \cdot \delta_j^l,$$



а) *б)*

Рис. 2. Построение транспонированного $FHT2DT$ паттерна $p^*_{DT}(3,3)$: *а)* Паттерн $FHT2DT$, содержащий пиксель с координатами $(x,y)=(3,3)$. Пиксель со значением t в нем указывает на то, что через этот пиксель проходит паттерн с наклоном t. Значение параметра s паттерна определяется значением ординаты его первого ($x=0$) пикселя; *б)* Транспонированный $FHT2DT$-паттерн $p^*_{DT}(3,3)$ как множество значений параметров (t,s) паттернов $FHT2DT$, проходящих через пиксель с координатами $(x,y)=(3,3)$

.

где δ_i^j – символ Кронекера. Например, $\mathcal{H}$ будет обозначать как БПХ, так и его матрицу в базисе $\{\boldsymbol{e}_{ij} \mid (i,j) \in \mathbb{Z}_{0,w-1} \times \mathbb{Z}_{0,h-1}\}$. Так как матрица Грама базиса $\{\boldsymbol{e}_{ij} \mid (i,j) \in \mathbb{Z}_{0,w-1} \times \mathbb{Z}_{0,h-1}\}$ является единичной, то из этого следует, что сопряженный, или транспонированный, оператор

$$\mathcal{H}^T \colon \mathcal{I}^{w\times h} \to \mathcal{I}^{w\times h}$$

имеет матрицу $\mathcal{H}^T$ в том же базисе $\{\boldsymbol{e}_{ij} \mid (i,j) \in \mathbb{Z}_{0,w-1} \times \mathbb{Z}_{0,h-1}\}$.

В то время как прямое БПХ $\mathcal{H}$ выполняет суммирование значений пикселей изображения $\boldsymbol{I}$ вдоль некоторого набора паттернов

$$\mathcal{P} = \{p(t,s) \mid (t,s) \in \mathbb{Z}_{0,w-1} \times \mathbb{Z}_{0,h-1}\},$$

транспонированное БПХ $\mathcal{H}^T$ выполняет суммирование значений пикселей изображения $\boldsymbol{I}$ вдоль некоторого набора транспонированных (сопряженных) паттернов

$$\mathcal{P}^T = \{p^T(x,y) \mid (x,y) \in \mathbb{Z}_{0,w-1} \times \mathbb{Z}_{0,h-1}\}.$$

Транспонированный паттерн $p^T(x,y)$ с параметрами $(x,y) \in \mathbb{Z}_{0,w-1} \times \mathbb{Z}_{0,h-1}$ определяется как

$$p^T(x,y) = \{(t,s) \mid (x,y) \in p(t,s),\ p(t,s) \in \mathcal{P}\}.$$

Иными словами, транспонированный паттерн $p^T(x,y)$ состоит из пар значений параметров (t,s) всех паттернов, проходящих через точку с координатами (x,y) (рис. 2). Действие транспонированного БПХ $\mathcal{H}^T$ можно записать следующим образом:

$$\mathcal{H}^T \colon \mathcal{I}_{w\times h} \to \mathcal{I}_{w\times h}, \quad \mathcal{H}^T\boldsymbol{I}(x,y) = \sum_{(t,s)\in p^T(x,y)} \boldsymbol{I}(t,s).$$

Быстрое и точное вычисление транспонированного ПХ крайне важно в ряде прикладных областей, в частности, оно является первостепенной необходимостью в задаче рентгеновской компьютерной томографии.

Хорошо известно, что транспонированный диадический паттерн также является диадическим [20,21]. Это означает, что оригинальный алгоритм Брейди – Ёна можно непосредственно использовать для вычисления сумм по транспонированным диадическим паттернам. Таким образом, алгоритм Брейди – Ёна дает самосопряженный метод для вычисления БПХ. Однако недостатком алгоритма Брейди – Ёна при вычислении БПХ является то, что он применим только к изображениям с шириной, равной степени двойки. На практике чаще всего встречаются изображения произвольной ширины, не обязательно равной степени двойки. Для таких изображений алгоритм Брейди – Ёна неприменим, что делает особенно актуальной потребность в транспонированных БПХ-алгоритмах, применимых к изображениям произвольной ширины, таких как алгоритм $FHT2DT$ [13, 14].

При этом матрица оператора, реализуемого алгоритмом $FHT2DT$, имеет нетривиальную структуру, что затрудняет применение метода транспонирования [1]. Это подчеркивает необходимость разработки сохраняющих сложность методов транспонирования для суммирующих алгоритмов, не зависящих от конкретного вида матрицы оператора. В настоящей статье предлагается такой метод.

2.2. Операторы прямого и обратного проецирования в КТ. В качестве еще одной особо важной задачи, ключевую роль в которой играет суммирующий оператор, в частности оператор БПХ, можно назвать задачу КТ. Наиболее вычислительно трудоемкой частью многих методов томографической реконструкции [22–24] является вычисление операторов прямого и обратного проецирования. Прямое проецирование описывается выражением

$$\boldsymbol{W}\boldsymbol{m} = \boldsymbol{g}, \tag{1}$$

где $\boldsymbol{g}$ – вектор линеаризованных лучевых сумм, $\boldsymbol{m}$ – вектор реконструированных значений, $\boldsymbol{W}$ – проекционная матрица размера $N \cdot C \times N^2$, N – число ячеек в линейке детектора, C – число проекций. Элементы проекционной матрицы $0 \leqslant w_{i,j} \leqslant 1$ определяют вклад пикселя в лучевую сумму, а сама матрица $\boldsymbol{W}$ описывает измерительную схему. Матрица $\boldsymbol{W}^T$ задает обратное проецирование [22].

Вычисление операторов проецирования используется для КТ-реконструкции на основе метода свертки и обратной проекции (filtered back projection, FBP) [25, 26], при реализации итеративных алгебраических подходов [21, 27, 28] и во многих подходах с использованием нейронных сетей [29–31]. Один из подходов быстрого приближенного вычисления операторов опирается на иерархическое разложение дискретного приближения линейных интегралов [12,32–37]. В дискретном пространстве семейства пересекающихся прямых могут иметь общие подпоследовательности пикселей (паттерны), поэтому требуемое для вычисления операторов число операций уменьшается за счет повторного использования частичных сумм, соответствующих пересечениям нескольких паттернов. Возникающие при использовании дискретного приближения ошибки аппроксимации ограничены и хорошо изучены [34, 37–39].

В рамках вышеупомянутого подхода иерархического разложения линейных интегралов паттерны как общие подмножества близко расположенных прямых в дискретном пространстве непосредственно относятся к определенным ранее паттернам БПХ. Таким образом, БПХ представляет собой существенное ядро целого ряда быстрых схем вычислений для операторов прямого проецирования в КТ. В то же время оператор обратного проецирования можно вычислить, используя транспонированный оператор прямого проецирования [1], что означает, что транспонированное БПХ служит эффективным средством для вычисления оператора обратного проецирования. Таким образом, схемы быстрого вычисления операторов прямого и обратного проецирования могут быть основаны на алгоритмах, эффективно вычисляющих пару операторов – прямое и транспонированное БПХ.

Схему быстрых приближенных вычислений для (1) можно представить в виде линейного оператора

$$\boldsymbol{B}\colon \mathbb{R}^n \to \mathbb{R}^m. \tag{2}$$

Матрица $\boldsymbol{B}$ этого оператора является булевой матрицей вида

$$\boldsymbol{B} \stackrel{\text{def}}{=} (\boldsymbol{B}(y,x) = b_{y,x} \in \{0,1\} :\ y \in \mathbb{Z}_{0,H-1}, x \in \mathbb{Z}_{0,W-1}),$$

где y обозначает номер строки в матрице $\boldsymbol{B}$, x – номер столбца в матрице $\boldsymbol{B}$, через $H \equiv H(\boldsymbol{B})$ обозначается количество строк матрицы, а через $W \equiv W(\boldsymbol{B})$ – количество столбцов матрицы, $H(\boldsymbol{B}), W(\boldsymbol{B}) \in \mathbb{Z}_{1,\infty}$.

Для оператора $\boldsymbol{B}$ вида (2) оператором обратного проецирования $\boldsymbol{B}^T$, задаваемым матрицей $\boldsymbol{B}^T$, будем называть транспонированный оператор

$$\boldsymbol{B}^T\colon \mathbb{R}^m \to \mathbb{R}^n. \tag{3}$$

Пусть $\mathrm{dec}^p(\cdot)$ – предварительно установленное разложение булевой матрицы в произведение p булевых матриц:

$$\mathrm{dec}^p(\boldsymbol{B}) \stackrel{\text{def}}{=} \prod_{i=p}^{1} \mathrm{dec}_i^p(\boldsymbol{B}) = \prod_{i=p}^{1} \boldsymbol{B}_i = \boldsymbol{B}_p \boldsymbol{B}_{p-1} \ldots \boldsymbol{B}_1 = \boldsymbol{B}, \tag{4}$$

где $\mathrm{dec}_i^p(\boldsymbol{B}) \stackrel{\text{def}}{=} \boldsymbol{B}_i$ – i-я компонента разложения. Мы предполагаем, что $\mathrm{dec}^1(\boldsymbol{B}) \equiv \boldsymbol{B}$, и без ограничения общности далее будем считать, что матрица $\boldsymbol{B}$ оператора $\boldsymbol{B}$ и все матрицы $\mathrm{dec}_i^p(\boldsymbol{B})$ в разложении не содержат нулевых строк и столбцов. Для каждого разложения $\mathrm{dec}^p(\boldsymbol{B})$ матрицы $\boldsymbol{B}$, представляющей прямой оператор $\boldsymbol{B}$, существует соответствующее разложение $\mathrm{dec}^p(\boldsymbol{B}^T)$ матрицы $\boldsymbol{B}^T$ для транспонированного оператора $\boldsymbol{B}^T$. Оно вычисляется через транспонирование произведения матриц:

$$\mathrm{dec}^p(\boldsymbol{B}^T) = \boldsymbol{B}^T = (\boldsymbol{B}_p \boldsymbol{B}_{p-1} \ldots \boldsymbol{B}_1)^T = \boldsymbol{B}_1^T \ldots \boldsymbol{B}_{p-1}^T \boldsymbol{B}_p^T = \prod_{i=p}^{1} \boldsymbol{B}_{p-i+1}^T. \tag{5}$$

Алгоритм вычисления оператора проецирования соответствует умножению на булевы матрицы. Как было отмечено выше, общий метод построения быстрого алгоритма вычисления линейного оператора обратного проецирования (3) при известном быстром алгоритме вычисления линейного оператора прямого проецирования (2) был предложен в работе [1] и опирается на соотношения (4) и (5). Согласно этому методу алгоритмы вычисления прямого и транспонированного операторов, а именно операторов прямого и обратного проецирования, имеют асимптотическую сложность (по числу сумм) одного порядка. Однако представление вычисления операторов в виде цепочек умножения булевых матриц, как требуется в методе из [1], усложняет его применение к известному быстрому алгоритму, выраженному в виде псевдокода или программы на каком-либо языке программирования. Матрица оператора суммирования, неявно реализуемая суммирующими алгоритмами, часто определяется рекурсивно (подобно рекурсивному построению паттернов в алгоритмах Брейди – Ёна и $FHT2DT$) и имеет размер, зависящий от размера входных данных, который в приложениях может быть достаточно большим для “ручного” анализа.

Далее в данной статье этот недостаток устраняется за счет изложения метода транспонирования суммирующих алгоритмов в терминах эквивалентной алгоритму вычислительной сети.

§ 3. Общие определения и методология

Назовем (суммирующей) сетью ориентированный ациклический граф (directed acyclic graph, DAG)

$$D \stackrel{\text{def}}{=} (V, A),$$

где V – множество вершин (значений), A – множество дуг (ориентированных ребер), описывающих порядок суммирования.

Далее используются следующие обозначения:

$\text{prev}(v) \stackrel{\text{def}}{=} \{u \in V : \exists(u, v) \in A\}$ – множество предков (входов) вершины $v \in V$,

$\text{next}(v) \stackrel{\text{def}}{=} \{u \in V : \exists(v, u) \in A\}$ – множество потомков (выходов) вершины $v \in V$,

$\text{inp}(D) \stackrel{\text{def}}{=} \{v \in V : \text{prev}(v) = \varnothing\}$ – вход сети, множество входных вершин сети D,

$\text{out}(D) \stackrel{\text{def}}{=} \{v \in V : \text{next}(v) = \varnothing\}$ – выход сети, множество выходных вершин сети D.

При заданных значениях входа $\text{inp}(D)$ значения в остальных вершинах такой сети вычисляются по формуле

$$v = \sum_{u \in \text{prev}(v)} u, \quad \forall v \in V \setminus \text{inp}(D).$$

3.1. Суммирующая сеть, эквивалентная матричному разложению в произведение булевых матриц. Покажем, как построить вычислительную суммирующую сеть, соответствующую алгоритму вычисления оператора проецирования, задаваемого разложением произведения булевых матриц вида (4). Для этого введем следующие обозначения:

$$\boldsymbol{V}^{(0)} = \boldsymbol{m},$$
$$\boldsymbol{V}^{(p)} = \boldsymbol{B}(\boldsymbol{m}),$$
$$\boldsymbol{V}^{(i)} = \boldsymbol{B}_i \boldsymbol{V}^{(i-1)}.$$

Вычисление оператора записывается в виде

$$\boldsymbol{V}^{(p)} = \boldsymbol{B}_p \boldsymbol{B}_{p-1} \dots \boldsymbol{B}_1 \boldsymbol{V}^{(0)} = \prod_{i=p}^{1} \text{dec}_i^p(\boldsymbol{B}) \boldsymbol{V}^{(0)}. \tag{6}$$

Далее будем использовать обозначение $V^{(i)}$ для множества вершин сети, составленного из элементов вектора $\boldsymbol{V}^{(i)}$. Алгоритму вычисления выражения (6) соответствует суммирующая сеть вида

$$D_{\boldsymbol{B}} = (V_{\boldsymbol{B}}, A_{\boldsymbol{B}}),$$
$$V_{\boldsymbol{B}} = \bigcup_{i=0}^{p} V^{(i)} : \quad \forall i \neq j, \quad V^{(i)}, V^{(j)} \in V \ \rightarrow \ V^{(i)} \cap V^{(j)} = \varnothing,$$
$$A_{\boldsymbol{B}} = \bigcup_{i,x,y} \left(v_x^{(i-1)}, v_y^{(i)}\right) : \quad \boldsymbol{B}_i(y, x) = 1,$$
$$\text{inp}(D_{\boldsymbol{B}}) = V^{(0)},$$
$$\text{out}(D_{\boldsymbol{B}}) = V^{(p)}.$$

Существование в сети дуги $a = (v_x^{(i-1)}, v_y^{(i)})$ означает, что значение вершины $v_x^{(i-1)} \in V^{(i-1)}$ входит в сумму при расчете значения вершины $v_y^{(i)} \in V^{(i)}$, поэтому

$$\boldsymbol{B}_i(y,x) = 1 \;\Leftrightarrow\; \exists (v_x^{(i-1)}, v_y^{(i)}) \in A_{\boldsymbol{B}}.$$

3.2. Эквипотенциальная суммирующая сеть. Покажем, что суммирущая сеть вычислительно эквивалентна цепочке умножений на булевы матрицы. Для этого определим расстояние $\mathrm{dist}\colon V_{\boldsymbol{B}} \to \mathbb{Z}_{0,\infty}$ в сети $D_{\boldsymbol{B}}$ от произвольной вершины $v \in V_{\boldsymbol{B}}$ до входа сети $\mathrm{inp}(D_{\boldsymbol{B}})$ по формуле

$$\mathrm{dist}(v) \stackrel{\mathrm{def}}{=} \begin{cases} 0, & \forall v \in \mathrm{inp}(D_{\boldsymbol{B}}), \\ \max\limits_{u \in \mathrm{prev}(v)} (\mathrm{dist}(u)) + 1, & \forall v \in V_{\boldsymbol{B}} \setminus \mathrm{inp}(D_{\boldsymbol{B}}). \end{cases}$$

Назовем i-слоем сети $D_{\boldsymbol{B}}$ и будем обозначать через $V^{(i)}$ эквидистантное подмножество вершин с заданным расстоянием $0 \leqslant i \leqslant \mathrm{depth}(D_{\boldsymbol{B}})$ до входа

$$V^{(i)} \stackrel{\mathrm{def}}{=} \{v \in V_{\boldsymbol{B}} :\ \mathrm{dist}(v) = i\},$$

где $\mathrm{depth}(D) = \max\limits_{v \in V_{\boldsymbol{B}}} \mathrm{dist}(v)$ – глубина сети (максимальная длина пути в сети). По определению

$$V_{\boldsymbol{B}} = \bigcup_{j=0}^{\mathrm{depth}(D_{\boldsymbol{B}})} V^{(j)}, \quad \forall i \neq k \;\to\; V^{(i)} \cap V^{(k)} = \varnothing.$$

Определим длину дуги $\mathrm{len}\colon A_{\boldsymbol{B}} \to \mathbb{N}$, $\forall a = (u,v) \in A_{\boldsymbol{B}}$, как

$$\mathrm{len}(a) = \mathrm{len}(u,v) \stackrel{\mathrm{def}}{=} \mathrm{dist}(v) - \mathrm{dist}(u).$$

Назовем сеть эквипотенциальной, если выполняется условие

$$\forall a \in A_{\boldsymbol{B}} \;\to\; \mathrm{len}(a) = 1. \tag{7}$$

В эквипотенциальной сети каждый последующий слой зависит только от предыдущего. При некотором заданном на $V_{\boldsymbol{B}}$ линейном порядке каждому слою $V^{(i)}$ соответствует единственный вектор $\boldsymbol{V}^{(i)}$, а зависимость между слоями можно описать в векторно-матричном виде как

$$\forall\, 1 \leqslant j \leqslant \mathrm{depth}(D_{\boldsymbol{B}}) \to \boldsymbol{V}^{(j)} = \boldsymbol{B}_j \boldsymbol{V}^{(j-1)} = \boldsymbol{B}_j \boldsymbol{B}_{j-1} \ldots \boldsymbol{B}_1 \boldsymbol{V}^{(0)},$$

где $\boldsymbol{B}_j$ – булева матрица. Таким образом, при выполнении условия (7) вычисление в сети описывается произведением булевых матриц вида (4).

3.3. Нормализация суммирующей сети. Покажем, что если условие эквипотенциальности (7) нарушено, то сеть D может быть нормализована, т.е. приведена к вычислительно эквивалентному эквипотенциальному виду. Пример основных шагов процесса нормализации показан на рис. 3, *а*)–*в*).

Вырожденной назовем вершину $v \in V$ суммирущей сети D, которая не участвует в сложениях (рис. 3*а*) и 3*б*)), не влияет на результат вычисления сети и не изменяет количество суммирований в сети. Для вырожденной вершины выполняется соотношение

$$\dim(\mathrm{prev}(v)) = \dim(\mathrm{next}(v)) = 1.$$

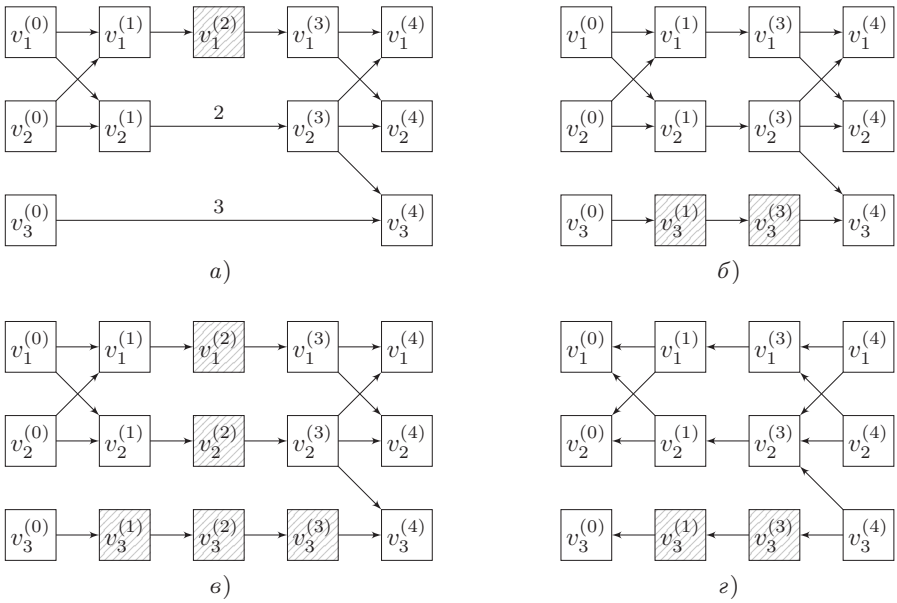


Рис. 3. Пример суммирующей сети; длины дуг подписаны только для значений, отличных от 1: *а*) исходная сеть с нарушением эквипотенциальности и вырожденной вершиной $v_1^{(2)}$; *б*) эквипотенциальная сеть с вырожденным слоем $V^{(2)}$; *в*) эквипотенциальная сеть после удаления вырожденных вершин $v_3^{(1)}$ и $v_3^{(3)}$; *г*) обратная эквипотенциальная суммирующая сеть с вырожденными вершинами $v_3^{(1)}$ и $v_3^{(3)}$

Слой сети, полностью состоящий из вырожденных вершин, как показано на рис. 3*б*), назовем вырожденным слоем. Удаление вырожденного слоя из эквипотенциальной сети сохраняет ее эквипотенциальность и не изменяет результатов вычисления.

Сформулируем способ приведения суммирущей сети к нормализованной эквипотенциальной форме. Пусть $\exists(u,v) \in A$: $1 < \ell = \operatorname{len}(u,v)$, тогда нормализация осуществляется добавлением в сеть $\ell - 1$ вырожденных вершин $\{e_i\}_{i=1}^{\ell-1}$ и заменой дуги (u,v) на последовательность дуг $(u,e_1),(e_1,e_2),\ldots,(e_{\ell-1},v)$. Таким способом можно заменить каждую дугу, нарушающую условие (7), на последовательность дуг, удалить все вырожденные слои и получить эквипотенциальную суммирующую сеть, вычислительно эквивалентную исходной (см. рис. 3*в*)). В дальнейшем все суммирующие сети будем считать эквипотенциальными.

3.4. Метод транспонирования суммирующего алгоритма. Покажем, как в терминах обращения суммирующих сетей представляется транспонирование алгоритма вычисления оператора проецирования. Пусть алгоритму вычисления прямого оператора соответствует сеть $D_{\boldsymbol{B}}$. Рассмотрим обратную сеть $D_{\boldsymbol{B}^{\mathrm{T}}}$, которая получается из прямой сети сменой направления всех дуг на противоположное:

$$\operatorname{inv}(a) = \operatorname{inv}\bigl(v_x^{(i-1)}, v_y^{(i)}\bigr) = \bigl(v_y^{(i)}, v_x^{(i-1)}\bigr).$$

Рассмотрим связь двух соседних слоев исходной сети. Для каждого ненулевого элемента матрицы $\boldsymbol{B}_i(k,j) = 1$ вершина $\boldsymbol{V}_j^{(i-1)}$ входит в сумму для вычисления $\boldsymbol{V}_k^{(i)}$, поэтому смена направления дуг в сети соответствует транспонированию матрицы перехода между слоями. При смене направления всех дуг слои суммируются в об-

ратном порядке. Таким образом, для обратной сети справедливо равенство

$$\boldsymbol{V}^{(i-1)} = \boldsymbol{B}_i^T \boldsymbol{V}^{(i)}$$

и $\forall i \in \mathbb{Z}_{0,p-1}$ можно вычислить

$$\boldsymbol{V}^{(i)} = \boldsymbol{B}_{i+1}^T \boldsymbol{V}^{(i+1)} = \boldsymbol{B}_{i+1}^T \boldsymbol{B}_{i+2}^T \dots \boldsymbol{B}_p^T \boldsymbol{V}^{(p)}.$$

Учитывая разложение (5), полная обратная сеть соответствует алгоритму вычисления транспонированного оператора

$$\boldsymbol{V}^{(0)} = \boldsymbol{B}_1^T \boldsymbol{B}_2^T \dots \boldsymbol{B}_p^T \boldsymbol{V}^{(p)} = \boldsymbol{B}^T \boldsymbol{V}^{(p)}.$$

Таким образом, если прямая сеть описывает вычисление оператора $\boldsymbol{B}$, то обратная сеть описывает вычисление оператора $\boldsymbol{B}^T$.

Общий метод транспонирования суммирующего алгоритма вычисления оператора $\boldsymbol{B}$ можно представить в виде следующих шагов:

1. представить суммирующий алгоритм вычисления оператора $\boldsymbol{B}$ в виде нормализованной суммирующей сети $D_{\boldsymbol{B}}$;
2. построить обратную суммирующую сеть $D_{\boldsymbol{B}^T}$ путем изменения направления всех дуг сети $D_{\boldsymbol{B}}$ на противоположные;
3. представить суммирующую сеть $D_{\boldsymbol{B}^T}$ в виде алгоритма вычисления оператора $\boldsymbol{B}^{\boldsymbol{T}}$.

Далее мы рассмотрим примеры практического применения обобщенного метода транспонирования суммирующих алгоритмов применительно к задачам вычисления операторов обратного проецирования в методах томографической реконструкции. На основе метода транспонирования с использованием ориентированных ациклических графов, предложенного в этой статье, также будет получено транспонирование алгоритма $FHT2DT$.

§ 4. Результаты

4.1. Транспонирование оператора прямого проецирования для задачи реконструкции в двумерной малоракурсной КТ с параллельно-лучевой схемой. Недостатком матричного представления в работе [1] является нетривиальность сопоставления алгоритма с их матричными факторизациями. В ней представлены примеры транспонирования алгоритмов, но при этом не показывается, какие части алгоритмов с какими матричными умножениями соотносятся, и кроме того, термин “распространение” (spreading) используется без формального определения.

Далее рассмотрим, как графовую интерпретацию можно применять для транспонирования алгоритма 2 вычисления оператора прямого проецирования для задачи двумерной малоракурсной КТ с параллельно-лучевой схемой из [1]. Здесь термин “малоракурсная” означает, что лучевые интегралы вычисляются по разреженному набору направлений, т.е. на вход алгоритма реконструкции подаются проекции, соответствующие разреженному набору направлений распространения рентгеновских лучей. Этот алгоритм для входного изображения I_{2^n} размера $2^n \times 2^n$ вычисляет выходной вектор сумм $\boldsymbol{s} = (s_j)_{j=0}^{q-1}$ для q дискретных прямых с параметрами (s,t), определяемых как множество $L = \{(x_j, a_j)\}_{j=0}^{q-1}$.

Алгоритм 2 Алгоритм для прямого БПХ с терминацией и досчетом

1: **procedure** $dirPFHT(L, I_{2^n}, n, k)$
2: $R_0(x, y, 0) \leftarrow I_{2^n}(x, y) \quad \forall x \in \mathbb{Z}_{0,2^n-1}, y \in \mathbb{Z}_{0,2^n-1}$
3: $R_0(x, y, 0) \leftarrow 0 \quad \forall x \in \mathbb{Z}_{2^n,2^{n+1}-1}, y \in \mathbb{Z}_{0,2^n-1}$
4: $R_k \leftarrow dirPFHTk(R_k, n, k)$
5: $s_j \leftarrow dirSUMk(x_j, a_j, R_k, n, k) \quad \forall j \in \mathbb{Z}_{0,q-1}, (x_j, a_j) \in L$
6: **return** s

Для понимания структуры суммирования элементов следует подробнее рассмотреть строки 4 и 5. Алгоритм 3 *dirPFHTk* вычисляет и сохраняет суммы для входного тензора R_0 в виде тензора R_k, производя суммирование по подпаттернам длины 2^k.

Алгоритм 3 Прямое БПХ с терминацией

1: **procedure** $dirPFHTk(R_0, n, k)$
2: **for** $i = 1$ to k **do**
3: **for** $a = 0$ to $2^i - 1$ **do**
4: **for** $y = 0$ to $2^n - 2^i$ step 2^i **do**
5: **for** $x = 0$ to $2^{n+1} - 1$ **do**
6: $x_2 \leftarrow (x - \lceil a/2 \rceil) \bmod 2^{n+1}$
7: $y_2 \leftarrow y + 2^{i-1}$
8: $R_i(x, y, a) \leftarrow R_{i-1}(x, y, \lfloor a/2 \rfloor) + R_{i-1}(x_2, y_2, \lfloor a/2 \rfloor)$
9: **return** R_k

Алгоритм 4 *dirSUMk* досчитывает сумму по паттерну с заданными параметрами (x, a), где $x \in \mathbb{Z}_{0,2^n-1}, a \in \mathbb{Z}_{0,2^n-1}$, для тензора R_k с суммами по подпаттернам длины 2^k через рекурсивный вызов функции *recSUMk*.

Алгоритм 4 Досчет для прямого оператора проецирования

1: **procedure** $dirSUMk(x, a, R_k, n, k)$
2: **return** $recSUMk(x, 0, a, n, R_k, n, k)$

Алгоритм 5 *recSUMk* рекурсивно суммирует предварительно вычисленные частичные суммы по паттернам длины 2^i, $i \in \mathbb{Z}_{k,n}$, в R_k вдоль дискретной прямой, заданной значениями (x, a).

Алгоритм 5 Досчет

1: **procedure** $recSUMk(x, y, a, i, R_k, n, k)$
2: **if** $i = k$ **then**
3: $s \leftarrow R_k(x, y, a)$
4: **else**
5: $x_2 \leftarrow (x - \lceil a/2 \rceil) \bmod 2^{n+1}$
6: $y_2 \leftarrow y + 2^{i-1}$
7: $s \leftarrow recSUMk(x, y, \lfloor a/2 \rfloor, i - 1, R_k, n, k) + recSUMk(x_2, y_2, \lfloor a/2 \rfloor, i - 1, R_k, n, k)$
8: **return** s

В алгоритме 2 прямое суммирование идет последовательно двумя этапами, что соответствует естественному разделению вычислительной сети на два последовательных сегмента, для первого из которых один некоторый слой является выходным, а для второго сегмента – входным. Первый сегмент сети соответствует вызову *dirPFHTk* в строке 4, второй сегмент сети соответствует всем вызовам *dirSUMk* в строке 5. При транспонировании алгоритма направления дуг сети меняются в каждом из сегментов, а сами сегменты вычисляются в обратном порядке. Транспонированный алгоритм 6 называется *revPFHT* (reversed Partial Fast Hough Transform).

Алгоритм 6 Транспонированный алгоритм *dirPFHT*

1: **procedure** $revPFHT(L, s, n, k)$
2: $R_k(x, y) \leftarrow 0 \quad \forall\, x \in \mathbb{Z}_{0,2^{n+1}-1}, y \in \mathbb{Z}_{0,2^n-1}$
3: $R_k \leftarrow revSUMk(s_j, x_j, a_j, R_k, n, k)\ \forall j \in \mathbb{Z}_{1,q}$
4: $R_0 \leftarrow revPFHTk(R_k, n, k)$
5: $I_{2^n}(x, y) \leftarrow R_0(x, y)\ \forall\, x \in \mathbb{Z}_{0,2^n-1}, y \in \mathbb{Z}_{0,2^n-1}$
6: **return** I_{2^n}

Он вычисляет действие транспонированного оператора на вектор лучевых сумм $\boldsymbol{s} = (s_i)_{i=0}^{q-1}$ и набор дискретных прямых $L = \{(x_i, a_i)\}_{i=0}^{q-1}$ в заданном диапазоне направлений. Для транспонирования первого сегмента сети в строке 2 алгоритма 6 используем алгоритм 7, запускающий вызов рекурсивной функции *recSPRk* алгоритма 8.

Алгоритм 7 Транспонированный алгоритм досчета *dirSUMk*

1: **procedure** $revSUMk(s, x, a, R_k, n, k)$
2: $R_k \leftarrow recSPRk(s, x, 0, a, n, R_k, n, k)$
3: **return** R_k

Положения значений частичных сумм в R_i определяются рекурсивным спуском в алгоритме 8, а "накопление суммы вдоль паттерна" после транспонирования соответствует накоплению в элементах R_k значений из $\boldsymbol{s}$ в соответствии с тем, из каких частичных сумм вдоль прямых складывались значения s_i.

Алгоритм 8 Алгоритм рекурсивного распространения для вычисления транспонированного оператора

1: **procedure** $recSPRk(s, x, y, a, i, R_k, n, k)$
2: **if** $i = k$ **then**
3: $R_k(x, y + a) \leftarrow R_k(x, y + a) + s$
4: **else**
5: $R_k \leftarrow recSPRk(s, x, y, \lfloor a/2 \rfloor, i - 1, R_k, n, k)$
6: $R_k \leftarrow recSPRk(s, x - \lceil a/2 \rceil, y + 2^{i-1}, \lfloor a/2 \rfloor, i - 1, R_k, n, k)$
7: **return** R_k

Второй сегмент сети соответствует прямому суммированию в алгоритме 3, которое выполняется в строке 8. Чтобы получить сеть с обратным порядком сложения, организуем счетчик циклов i в строке 2 в обратном порядке, а также "обратное" суммирование, что соответствует строкам 9 и 10 алгоритма 9.

Алгоритм 9 Окончание вычисления полностью транспонированного оператора для матрицы R_k

1: **procedure** $revPFHTk(R_k, n, k)$
2: **for** $i = k$ to 1 step -1 **do**
3: $R_{i-1}(x, y, 0) \leftarrow 0 \quad \forall\, x \in \mathbb{Z}_{0,2^{n+1}-1}, y \in \mathbb{Z}_{0,2^n-1}$
4: **for** $a = 0$ to $2^i - 1$ **do**
5: **for** $y = 0$ to $2^n - 2^i$ step 2^i **do**
6: **for** $x = 0$ to $2^{n+1} - 1$ **do**
7: $x_2 \leftarrow (x - \lceil a/2 \rceil) \bmod 2^{n+1}$
8: $y_2 \leftarrow y + 2^{i-1}$
9: $R_{i-1}(x, y, \lfloor a/2 \rfloor) \leftarrow R_{i-1}(x, y, \lfloor a/2 \rfloor) + R_i(x, y, a)$
10: $R_{i-1}(x_2, y_2, \lfloor a/2 \rfloor) \leftarrow R_{i-1}(x_2, y_2, \lfloor a/2 \rfloor) + R_i(x, y, a)$
11: **return** R_k

Измерения временной сложности алгоритмов, представленных в данном пункте, для различных значений входных параметров можно найти в [1].

4.2. Транспонирование алгоритма *FHT2DT*. Недавно в [13, 14] был предложен новый быстрый алгоритм $FHT2DT$ для вычисления преобразования Хафа. Алгоритм $FHT2DT$ отличается возможностью обработки изображений произвольного размера, тогда как, например, де-факто стандартный алгоритм Брейди – Ёна позволяет быстро вычислить преобразование Хафа для изображений с шириной, строго равной степени двойки. При этом, как доказано в том же исследовании [13, 14], алгоритм $FHT2DT$ является более точным по сравнению с другими известными ранее и описанными в литературе алгоритмами быстрого преобразования Хафа, работающими для случая произвольного размера изображения.

Алгоритм $FHT2DT$ устроен следующим образом (см. алгоритм 10, [13,14]). Изображение $I = I_{w\times h}$ произвольной ширины w и высоты h разбивается на два подызображения – левое I_L и правое I_R (алгоритм 10, строки 4–8). Левое подызображение I_L имеет ширину

$$w_L = 2^{\lceil \log_2 w \rceil - 1},$$

т.е. равно максимальной степени двойки, меньшей ширины исходного изображения; второе подызображение I_R имеет ширину, равную $w_R = w - w_L$. Высота обоих подызображений равна h. Для левого и правого подызображений I_L и I_R вычисляется (рекурсивно) преобразование Хафа $FHT2DT$ и получаются Хаф-образы J_L и J_R (алгоритм 10, строки 9–10). Затем Хаф-образы J_L и J_R подызображений I_L и I_R объединяются (алгоритм 10, строки 11–18) и образуют изображение J, которое и является результатом работы алгоритма $FHT2DT$.

Алгоритм 10 Алгоритм $FHT2DT$ ($dirFHT2DT$) вычисления БПХ для изображения произвольного размера

```
 1: procedure FHT2DT(w, h, I = I_{w×h})
 2:     if w > 1 then
 3:         p ← ⌈log_2 w⌉ − 1
 4:         w_L ← 2^p
 5:         w_R ← w − w_L
 6:         I_L ← I(0 : w_L, :)                          ▷ I_L – изображение
 7:         I_R ← I(w_L : w, :)                          ▷ I_R – изображение
 8:         J_L ← FHT2DT(w_L, h, I_L)                    ▷ J_L – изображение
 9:         J_R ← FHT2DT(w_R, h, I_R)                    ▷ J_R – изображение
10:         J ← Create_Zeroed_Image(w, h)                ▷ J – изображение
11:         k_L ← (w_L − 1)/(w − 1)
12:         k_R ← (w_R − 1)/(w − 1)
13:         for t ← 0 to w − 1 do
14:             t_L ← [t k_L]
15:             t_R ← [t k_R]
16:             s ← (t − t_R) mod h
17:             J(t, :) ← J_L(t_L, :) + Concat(J_R(t_R, s : h), J_R(t_R, 0 : s))
18:     else
19:         J ← I
20:     return J = J_{w×h}
```

В алгоритме 10 используется несколько вспомогательных функций. Функция $Create\_Zeroed\_Image(w, h)$ возвращает изображение размера $w \times h$. Кроме того, функция $Concat(I_1, I_2)$ возвращает изображение, являющееся конкатенацией изображений I_1 и I_2 по горизонтальной оси. Мы также будем придерживаться slice-нотации, т.е. $n_1 : n_2$ обозначает диапазон от n_1 до n_2 (включая n_1, не включая n_2).

Отсутствие индекса n_1 или n_2 указывает на полный диапазон с соответствующей стороны. Отсутствие обоих индексов указывает на полный диапазон.

Метод транспонирования суммирующих алгоритмов, предложенный в данной статье, мы применили к алгоритму $FHT2DT$ ($dirFHT2DT$). Алгоритм 11 описывает транспонированный алгоритм $revFHT2DT$. Он позволяет быстро вычислить транспонированное (сопряженное) преобразование Хафа, поскольку наш метод транспонирования суммирующих алгоритмов сохраняет асимптотическую вычислительную сложность прямого алгоритма $FHT2DT$ ($dirFHT2DT$).

Алгоритм 11 Алгоритм $revFHT2DT$ вычисления транспонированного БПХ для изображения произвольного размера

```
1:  procedure revFHT2DT(w, h, J = J_{w×h})
2:      if w > 1 then
3:          p ← ⌈log_2 w⌉ − 1
4:          w_L ← 2^p
5:          w_R ← w − w_L
6:          k_L ← (w_L − 1)/(w − 1)
7:          k_R ← (w_R − 1)/(w − 1)
8:          J_L ← Create_Zeroed_Image(w_L, h)
                                  ▷ Для распространения по левым подпаттернам
9:          J_R ← Create_Zeroed_Image(w_R, h)
                                  ▷ Для распространения по правым подпаттернам
10:         for t ← 0 to w − 1 do
11:             t_L ← [t k_L] mod w_L
12:             t_R ← [t k_R] mod w_R
13:             s ← (t_R − t) mod h
14:             J_L(t_L, :) ← J_L(t_L, :) + J(t, :)   ▷ Распространение по подпаттернам
15:             J_R(t_R, :) ← J_R(t_R, :) + Concat(J(t, s : h), J(t, 0 : s))
16:         I_L ← revFHT2DT(w_L, h, J_L)
                          ▷ Вычисление изображения Хафа после распространения
17:         I_R ← revFHT2DT(w_R, h, J_R)
                          ▷ Вычисление изображения Хафа после распространения
18:         I ← Concat(I_L, I_R)                  ▷ Конкатенация изображений Хафа
19:     else
20:         I ← J
21:     return I = I_{w×h}
```

Для транспонирования алгоритма $FHT2DT$, следуя обращенным ребрам вычислительного графа для алгоритма $FHT2DT$, мы распространяем (в смысле, описанном в предыдущем пункте) значение $J(t, s)$ каждого пикселя (t, s) входного изображения $J = J_{w\times h}$ на левую (длины w_L) и правую (длины w_R) части паттерна $FHT2DT$ (которые называются подпаттернами) с параметрами (t, s). Это реализовано в строках 4–16 (алгоритм 11). В результате получаются два изображения J_L и J_R шириной w_L и w_R со значениями пикселей входного изображения J, распространенными по левому (J_L) и правому (J_R) подпаттернам. Далее для изображений J_L и J_R вычисляются (рекурсивно) образы I_L и I_R транспонированного преобразования $FHT2DT$ (алгоритм 11, строки 17–18). Выходом транспонированного алгоритма $revFHT2DT$ является конкатенация изображений I_L и I_R по их высотному измерению (алгоритм 11, строка 19).

§ 5. Обсуждение

Была проведена проверка корректности разработанного нами и представленного в предыдущем параграфе алгоритма $revPFHT$ для вычисления оператора об-

ратного проецирования: результаты, полученные при применении к изображениям произвольного размера, согласуются с результатами соответствующего алгоритма из оригинальной работы [1]. Алгоритм $revFHT2DT$ также был протестирован на широкой выборке изображений. Для этого его результаты сравнивалась с результатами работы эталонного алгоритма, который, следуя определению транспонированного БПХ, наивно строит транспонированные паттерны $FHT2DT$ и циклически суммирует значения внутри них. Тестирование не выявило никаких расхождений.

Обсуждая суть предлагаемого подхода, основанного на использовании DAG, мы добавим, что создание формализмов для представления вычислительного процесса в виде ориентированного графа (data flow graph, DFG) продолжается уже около 50 лет, причем DFG является одним из важных внутренних промежуточных представлений [40]. В общем случае DFG не содержит всей информации о программе, но вычислительный граф для операторов проецирования можно сгенерировать без управляющих инструкций, поэтому порядок вычислений определяется только топологией сети DFG.

Графовые представления алгоритмов остаются в центре внимания исследователей и активно используются для распараллеливания и оптимизации [41–43], чтобы максимально использовать масштабируемые гетерогенные архитектуры. Оптимизацию можно выполнять эффективно и автоматически на основе DFG для алгоритмов вычисления операторов как прямого, так и обратного проецирования. Мы также надеемся, что предложенное представление вычислительного графа для алгоритмов вычисления оператора проецирования и автоматизация генерации вычислительной сети оператора обратного проецирования будут способствовать развитию нейросетевых методов с использованием алгоритмов БПХ [44, 45], в том числе в задаче КТ [46–50]. Кроме того, мы считаем, что предложенный подход к транспонированию послужит важным инструментом для изучения самосопряженных БПХ-алгоритмов, обобщающих алгоритм Брейди – Ёна для изображений произвольного размера.

§ 6. Заключение

В данной статье предложен общий метод транспонирования суммирующего алгоритма на основе его DAG-представления. Другими словами, метод позволяет, имея алгоритм вычисления действия прямого суммирующего оператора в виде DAG, эффективно получить результат действия транспонированного оператора. Последнее, как доказано в статье, достигается технически несложной инверсией дуг нормализованного эквипотенциального DAG. Таким образом, предложенный в данной статье метод транспонирования на основе DAG имеет существенное преимущество, поскольку не требует описания входного алгоритма в виде конкретного перемножения булевых матриц. Мы подчеркиваем, что итоговый транспонированный алгоритм, полученный в результате применения предложенного метода транспонирования, обладает той же асимптотической вычислительной сложностью, что и исходный прямой алгоритм (см. [1]).

В статье приведены инструкции по применению общего метода, включающие этап задания порядка вычислений для DAG-представления алгоритма, нормализации вычислительной сети и последующей смены ориентации дуг графа. При этом изложенный метод, в соответствии с инструкциями, был применен в качестве подробного примера к задаче вычисления оператора обратного проецирования в задаче реконструкции в двумерной малоракурсной КТ с параллельно-лучевой схемой. Кроме того, предложенный метод транспонирования был применен к новому алгоритму $FHT2DT$ для быстрого и точного вычисления ПХ. Алгоритм транспонирования $revFHT2DT$ обеспечивает новый вычислительно эффективный подход к вычислению транспонированного ПХ для изображений произвольного размера,

устраняя ограничение транспонированного алгоритма Брейди – Ёна, который применим только к изображениям с шириной, равной степени двойки.

СПИСОК ЛИТЕРАТУРЫ

1. *Polevoy D., Gilmanov M., Kazimirov D., Chukalina M., Ingacheva A., Kulagin P., Nikolaev D.* Tomographic Reconstruction: General Approach to Fast Back-Projection Algorithms // Mathematics. 2023. V. 11. № 23. Paper No. 4759 (37 pp.). https://doi.org/10.3390/math11234759
2. *Hough P.V.C.* Machine Analysis of Bubble Chamber Pictures // Proc. 2nd Int. Conf. on High-Energy Accelerators and Instrumentation (HEACC 1959). CERN, Geneva, Switzerland. Sept. 14–19, 1959. P. 554–558.
3. *Illingworth J., Kittler J.* A Survey of the Hough Transform // Comput. Vision Graph. Image Process. 1988. V. 44. № 1. P. 87–116. https://doi.org/10.1016/S0734-189X(88)80033-1
4. *Chaloeivoot T., Phiphobmongkol S.* Building Detection from Terrestrial Images // J. Image Graph. 2016. V. 4. № 1. P. 46–50.
5. *Rahmdel P.S., Comley R., Shi D., McElduff S.* A Review of Hough Transform and Line Segment Detection Approaches // Proc. 10th Int. Conf. on Computer Vision Theory and Applications (VISAPP 2015). Berlin, Germany. Mar. 11–14, 2015. V. 2. P. 411–418. https://doi.org/10.5220/0005268904110418
6. *Aggarwal N., Karl W.* Line Detection in Images through Regularized Hough Transform // IEEE Trans. Image Process. 2006. V. 15. № 3. P. 582–591. https://doi.org/10.1109/TIP.2005.863021
7. *Mukhopadhyay P., Chaudhuri B.B.* A Survey of Hough Transform // Pattern Recognit. 2015. V. 48. № 3. P. 993–1010. https://doi.org/10.1016/j.patcog.2014.08.027
8. *Алиев М.А., Николаев Д.П., Сараев А.А.* Построение быстрых вычислительных схем настройки алгоритма бинаризации Ниблэка // Тр. ИСА РАН. 2014. Т. 64. № 3. С. 25–34.
9. *Ozturk H., Saricam I.T.* Core Segmentation and Fracture Path Detection Using Shadows // J. Image Graph. 2018. V. 6. № 1. P. 69–73.
10. *Saha S., Basu S., Nasipuri M., Basu D.* A Hough Transform Based Technique for Text Segmentation // J. Comput. 2010. V. 2. № 2. P. 134–141.
11. *Yazdi M., Mohammadi M.* Metal Artifact Reduction in Dental Computed Tomography Images Based on Sinogram Segmentation Using Curvelet Transform Followed by Hough Transform // J. Med. Signals Sens. 2017. V. 7. № 3. P. 145–152.
12. *Brady M.L., Yong W.* Fast Parallel Discrete Approximation Algorithms for the Radon Transform // Proc. 4th Ann. ACM Symp. on Parallel Algorithms and Architectures (SPAA'92). San Diego, California, USA. June 29 – July 1, 1992. P. 91–99. https://doi.org/10.1145/140901.140911
13. *Kazimirov D., Nikolaev D., Rybakova E., Terekhin A.* Generalization of Brady–Yong Algorithm for Fast Hough Transform to Arbitrary Image Size, https://arxiv.org/abs/2411.07351 [cs.CV], 2024.
14. *Kazimirov D., Nikolaev D., Rybakova E., Terekhin A.* Generalization of Brady–Yong Algorithm for Fast Hough Transform to Arbitrary Image Size // Proc. 5th Symp. on Pattern Recognition and Applications (SPRA 2024). Istanbul, Turkey. Nov. 11–13, 2024 (to appear).
15. *Jahan R, Suman P., Singh D.K.* Lane Detection Using Canny Edge Detection and Hough Transform on Raspberry Pi // Int. J. Adv. Res. Comput. Sci. 2018. V. 9. № 2. P. 85–89.
16. *Thongpan N., Rattanasiriwongwut M., Ketcham M.* Lane Detection Using Embedded System // Int. J. Comput. Internet Manag. 2020. V. 28. № 2. P. 46–51.
17. *Panfilova E., Shipitko O.S., Kunina I.* Fast Hough Transform-Based Road Markings Detection For Autonomous Vehicle // 13th Int. Conf. on Machine Vision (ICMV 2020). Rome, Italy. Nov. 2–6, 2020. Proc. SPIE. V. 11605. P. 671–680. https://doi.org/10.1117/12.2587615
18. *Котов А.А., Коноваленко И.А., Николаев Д.П.* Прослеживание объектов в видеопотоке, оптимизированное с помощью быстрого преобразования Хафа // ИтиВС. 2015. № 1. С. 56–68.

19. *van den Braak G.-J., Nugteren C., Mesman B., Corporaal H.* Fast Hough Transform on GPUs: Exploration of Algorithm Trade-Offs // Advanced Concepts For Intelligent Vision Systems: Proc. 13th Int. Conf. ACIVS 2011. Ghent, Belgium. Aug. 22–25, 2011. Lect. Notes Comput. Sci. V. 6915. Berlin: Springer, 2011. P. 611–622. https://doi.org/10.1007/978-3-642-23687-7_55

20. *Brady M.L.* A Fast Discrete Approximation Algorithm for the Radon Transform // SIAM J. Comput. 1998. V. 27. № 1. P. 107–119. https://doi.org/10.1137/S0097539793256673

21. *Prun V.E., Nikolaev D.P., Buzmakov A.V., Chukalina M.V., Asadchikov V.E.* Effective Regularized Algebraic Reconstruction Technique for Computed Tomography // Crystallogr. Rep. 2013. V. 58. № 7. P. 1063–1066. https://doi.org/10.1134/S1063774513070158

22. *Buzug T.M.* Computed Tomography: From Photon Statistics to Modern Cone-Beam CT. Berlin: Springer, 2008. https://doi.org/10.1007/978-3-540-39408-2

23. *Withers P.J., Bouman C., Carmignato S., Cnudde V., Grimaldi D., Hagen C.K., Maire E., Manley M., Du Plessis A., Stock S.R.* X-Ray Computed Tomography // Nat. Rev. Methods Primers. 2021. V. 1. № 1. Article No. 18. https://doi.org/10.1038/s43586-021-00015-4

24. *Arlazarov V.L., Nikolaev D.P., Arlazarov V.V., Chukalina M.V.* X-Ray Tomography: The Way from Layer-by-Layer Radiography to Computed Tomography // Компьютерная оптика. 2021. Т. 45. № 6. С. 897–906. https://doi.org/10.18287/2412-6179-CO-898

25. *Lewitt R.M.* Reconstruction Algorithms: Transform Methods // Proc. IEEE. 1983. V. 71. № 3. P. 390–408. https://doi.org/10.1109/PROC.1983.12597

26. *Dolmatova A., Chukalina M., Nikolaev D.* Accelerated FBP for Computed Tomography Image Reconstruction // Proc. 2020 IEEE Int. Conf. on Image Processing (ICIP 2020). Abu Dhabi, United Arab Emirates. Virtual Conf. Oct. 25–28, 2020. P. 3030–3034. https://doi.org/10.1109/ICIP40778.2020.9191044

27. *Mileto A., Guimaraes L.S., McCollough C.H., Fletcher J.G., Yu. L.* State of the Art in Abdominal CT: The Limits of Iterative Reconstruction Algorithms // Radiology. 2019. V. 293. № 3. P. 491–503. https://doi.org/10.1148/radiol.2019191422

28. *Kasai R., Yamaguchi Y., Kojima T., Abou Al-Ola O.M., Yoshinaga T.* Noise-Robust Image Reconstruction Based on Minimizing Extended Class of Power-Divergence Measures // Entropy. V. 23. № 8. 2021. Paper No. 1005 (16 pp.). https://doi.org/10.3390/e23081005

29. *Kerr J.P., Bartlett E.B.* Neural Network Reconstruction of Single-Photon Emission Computed Tomography Images // J. Digit. Imaging. 1995. V. 8. № 3. P. 116–126. https://doi.org/10.1007/BF03168085

30. *Adler J., Öktem O.* Learned Primal-Dual Reconstruction // IEEE Trans. Med. Imaging. 2018. V. 37. № 6. P. 1322–1332. https://doi.org/10.1109/TMI.2018.2799231

31. *Yamaev A.V., Chukalina M.V., Nikolaev D.P., Kochiev L.G., Chulichkov A.I.* Neural Network Regularization in the Problem of Few View Computed Tomography // Компьютерная оптика. 2022. Т. 46. № 3. С. 422–428. https://doi.org/10.18287/2412-6179-CO-1035

32. *Götz W.A., Druckmüller H.J.* A Fast Digital Radon Transform—An Efficient Means for Evaluating the Hough Transform // Pattern Recognit. 1995. V. 28. № 12. P. 1985–1992. https://doi.org/10.1016/0031-3203(95)00057-7

33. *Wu T.-K., Brady M.L.* A Fast Approximation Algorithm for 3D Image Reconstruction // Proc. 1998 Int. Computer Symp. Workshop on Image Processing and Character Recognition. Tainan, Taiwan. Dec. 17–19, 1998. P. 213–220.

34. *Ершов Е.И., Терехин А.П., Николаев Д.П.* Обобщение быстрого преобразования Хафа для трехмерных изображений // Информационные процессы. 2017. Т. 17. № 4. С. 294–308.

35. *Aliev M., Ershov E.I., Nikolaev D.P.* On the Use of FHT, Its Modification for Practical Applications and the Structure of Hough Image // 11th Int. Conf. on Machine Vision (ICMV 2018). Munich, Germany. Nov. 1–3, 2018. Proc. SPIE. V. 11041. P. 1–9. https://doi.org/10.1117/12.2522803

36. *Bulatov K.B., Chukalina M.V., Nikolaev D.P.* Fast X-Ray Sum Calculation Algorithm for Computed tomography problem // Вестник ЮурГУ ММП. 2020. Т. 13. № 1. С. 95–106. https://doi.org/10.14529/mmp200107

37. *Nikolaev D., Ershov E., Kroshnin A., Limonova E., Mukovozov A., Faradzhev I.* On a Fast Hough/Radon Transform as a Compact Summation Scheme over Digital Straight Line Segments // Mathematics. 2023. V. 15. № 15. Papre No. 3336 (22 pp.). https://doi.org/10.3390/math11153336

38. *Карпенко С.М., Ершов Е.И.* Исследование свойств диадического паттерна быстрого преобразования Хафа // Пробл. передачи информ. 2021. Т. 57. № 3. С. 102–111. https://doi.org/10.31857/S0555292321030074

39. *Ershov E., Terekhin A., Nikolaev D., Postnikov V., Karpenko S.* Fast Hough Transform Analysis: Pattern Deviation from Line Segment // 8th Int. Conf. on Machine Vision (ICMV 2015). Barcelona, Spain. Nov. 19–21, 2015. Proc. SPIE. V. 9875. P. 42–46. https://doi.org/10.1117/12.2228852

40. *Stanier J., Watson D.* Intermediate Representations in Imperative Compilers: A Survey // ACM Comput. Surv. (CSUR). 2013. V. 45. № 3. Article No. 26. P. 1–27. https://doi.org/10.1145/2480741.2480743

41. *Gandarillas V., Joshy A.J., Sperry M.Z., Ivanov A.K., Hwang J.T.*, A Graph-based Methodology for Constructing Computational Models that Automates Adjoint-based Sensitivity Analysis // Struct. Multidiscip. Optim. 2024. V. 67. № 5. P. 76. https://doi.org/10.1007/s00158-024-03792-0

42. *Shingde N., Blattner T., Bardakoff A., Keyrouz W., Berzins M.* An Illustration of Extending Hedgehog to Multi-Node GPU Architectures Using GEMM // SN Comput. Sci. 2024. V. 5. № 5. Article No. 654. https://doi.org/10.1007/s42979-024-02917-y

43. *Bardakoff A.* Analysis and Execution of a Data-Flow Graph Explicit Model Using Static Metaprogramming. Ph.D. Thesis. Université Clermont Auvergne, Clermont-Ferrand, France, 2021. Available at https://theses.hal.science/tel-03813645v1.

44. *Sheshkus A., Ingacheva A., Arlazarov V., Nikolaev D.* HoughNet: Neural Network Architecture for Vanishing Points Detection // Proc. 15th IAPP Int. Conf. on Document Analysis and Recognition (ICDAR 2019). Sept. 20–25, 2019. Sydney, NSW, Australia. P. 844–849. https://doi.org/10.1109/ICDAR.2019.00140

45. *Sheshkus A., Nikolaev D.P., Arlazarov V.L.* Houghencoder: Neural Network Architecture for Document Image Semantic Segmentation // Proc. 2020 IEEE Int. Conf. on Image Processing (ICIP 2020). Abu Dhabi, United Arab Emirates. Virtual Conf. Oct. 25–28, 2020. P. 1946–1950. https://doi.org/10.1109/ICIP40778.2020.9191182

46. *Yamaev A., Chukalina M., Nikolaev D., Sheshkus A. Chulichkov A.* Lightweight Denoising Filtering Neural Network for FBP Algorithm // 13th Int. Conf. on Machine Vision (ICMV 2020). Rome, Italy. Nov. 2–6, 2020. Proc. SPIE. V. 11605. P. 158–167. https://doi.org/10.1117/12.2587185

47. *Ge R., He Y., Xia C., Sun H., Zhang Y., Hu D., Chen S., Chen Y., Li S., Zhang D.* DDPNet: A Novel Dual-Domain Parallel Network for Low-Dose CT Reconstruction // Medical Image Computing and Computer Assisted Intervention – MICCAI 2022: Proc. 25th Int. Conf. Singapore. Sept. 18–22, 2022. Part VI. Lect. Notes Comput. Sci. V. 6915. Cham: Springer, 2022. P. 748–757. https://doi.org/10.1007/978-3-031-16446-0_71

48. *Niu C., Li M., Guo X., Wang G.* Self-supervised Dual-Domain Network for Low-Dose CT Denoising // Developments in X-Ray Tomography XIV. San Diego, California, United States. Aug. 22–24, 2022. Proc. SPIE. V. 12242. P. 85–91. https://doi.org/10.1117/12.2633197

49. *Smolin A., Yamaev A., Ingacheva A., Shevtsova T., Polevoy D., Chukalina M., Nikolaev D., Arlazarov V.* Reprojection-based Numerical Measure of Robustness for CT Reconstruction Neural Networks Algorithms // Mathematics. 2022. V. 10. № 22. Paper No. 4210 (17 pp.). https://doi.org/10.3390/math10224210

50. *Kojima T., Yoshinaga T.* Iterative Image Reconstruction Algorithm with Parameter Estimation by Neural Network for Computed Tomography // Algorithms. 2023. V. 16. № 1. Paper No. 60 (18 pp.). https://doi.org/10.3390/a16010060

Полевой Дмитрий Валерьевич
Институт проблем передачи информации
им. А.А. Харкевича РАН, Москва
Федеральный исследовательский центр
"Информатика и управление" РАН, Москва
ООО "Смарт Энджинс Сервис", Москва
dvpsun@gmail.com
Казимиров Данил Дмитриевич
Институт проблем передачи информации
им. А.А. Харкевича РАН, Москва
ООО "Смарт Энджинс Сервис", Москва
d.kazimirov@smartengines.com
Чукалина Марина Валерьевна
Николаев Дмитрий Петрович
Федеральный исследовательский центр
"Информатика и управление" РАН, Москва
ООО "Смарт Энджинс Сервис", Москва
m.chukalina@smartengines.com
d.p.nikolaev@smartengines.com

Поступила в редакцию
18.10.2024
После доработки
06.12.2024
Принята к публикации
25.12.2024

ПРОБЛЕМЫ ПЕРЕДАЧИ ИНФОРМАЦИИ

Том 60 **2024** **Вып. 4**

УДК 510.51 : 004.932 : 519.6 : 519.171

© 2024 г. **Д.Д. Казимиров, Д.П. Николаев, Е.О. Рыбакова, А.П. Терехин**

БЫСТРЫЙ АЛГОРИТМ ВЫЧИСЛЕНИЯ ПРЕОБРАЗОВАНИЯ ХАФА ДЛЯ ИЗОБРАЖЕНИЙ ПРОИЗВОЛЬНОГО РАЗМЕРА С ПЕРЕИСПОЛЬЗОВАНИЕМ ВЫДЕЛЕННОЙ ПАМЯТИ

In-place алгоритмы эффективно используют память, уже выделенную для входных данных, ограничиваясь лишь незначительным дополнительным объемом памяти для промежуточных вычислений. Для изображений ширины, равной степени двойки, известен in-place алгоритм, являющийся вариацией стандартного алгоритма Брейди – Ёна для вычисления преобразования Хафа. Однако этот алгоритм неприменим к изображениям с произвольной шириной, наиболее часто встречающимся на практике. Напротив, out-of-place алгоритм $FHT2DS$ может обрабатывать изображения различных размеров. В настоящей статье представлен in-place вариант алгоритма $FHT2DS$, названный $FHT2IDS$. Мы показываем, что алгоритм $FHT2IDS$ дает такие же результаты, как и алгоритм $FHT2DS$, но использует значительно меньше памяти на каждом шаге рекурсии. В частности, на каждом шаге рекурсии алгоритм $FHT2IDS$ требует массива размера не более $w+h$ (где w и h – ширина и высота изображения), в то время как алгоритм $FHT2DS$ требует массива размера wh. Экспериментальные результаты показывают, что алгоритм $FHT2IDS$, реализованный на C/C++, работает на 26% быстрее своего out-of-place аналога, алгоритма $FHT2DS$. Алгоритм $FHT2IDS$ также доступен на Python через открытый исходный код библиотеки *adrt*.

Ключевые слова: преобразование Хафа, быстрое преобразование Хафа, приближенное дискретное преобразование Радона, in-place, вычислительный граф.

DOI: 10.31857/S0555292324040065, **EDN:** VKDONT

§ 1. Введение

Преобразование Хафа (ПХ) широко используется в обработке изображений и машинном зрении. Оно обычно рассматривается как инструмент для робастной оценки параметров одной или нескольких прямых на дискретном изображении путем определения числа точек, лежащих на каждой из заранее заданных прямых. Этот метод был предложен Полом Хафом в 1959 году как способ обнаружения прямолинейных траекторий в экспериментах с пузырьковыми камерами [1]. Основной принцип преобразования Хафа заключается в накоплении “голосов” вдоль дискретизированных прямых в заданной параметризации и присвоении накопленного значения каждой прямой. Это накопленное значение увеличивается с вероятностью присутствия соответствующей прямой на изображении.

Исторически преобразование Хафа (ПХ) в первую очередь было известно благодаря своему применению для обнаружения контрастных прямых или их сегментов на изображениях [2–5]. Однако с течением времени область его применения значительно расширилась. Преобразование Хафа используется в различных областях, включая бинаризацию изображений [6] и сегментацию [7,8], а также для таких задач,

как автоматическое определение параметров аберрации оптической системы [9] и робастная ортогональная линейная регрессия на низкоразмерных гистограммах [10].

Преобразование Хафа требует использования быстрых вычислительных алгоритмов (алгоритмы быстрого преобразования Хафа, или БПХ). В 1992 году М. Брейди и В. Ён разработали метод для приближенного дискретного преобразования Радона [11], предназначенный для квадратных изображений размера $2^q \times 2^q, q \in \mathbb{N}$. В аннотации и заключении своей работы авторы взаимозаменяемо используют термины "преобразование Хафа" и "преобразование Радона". Их подход основывается на динамическом программировании, что позволяет исключить избыточные вычисления сумм для ранее обработанных сегментов при определении суммы вдоль соответствующих прямых. Это приводит к эффективному вычислению преобразования Хафа для изображения размера $n \times n$ за $\mathcal{O}(n^2 \log n)$ операций сложения. Оценка ортотропной ошибки аппроксимации прямых диадическими дискретизациями, используемыми в методе Брейди – Ёна, была подробно изучена в литературе [12]. Метод Брейди – Ёна стал стандартным подходом для практических применений преобразования Хафа, так как он предоставляет вычислительно эффективное решение, сохраняя при этом основную идею оригинальной техники [13–18].

Широкая применимость преобразования Хафа на практике накладывает строгие требования к производительности и объему памяти на алгоритмы БПХ, особенно во встроенных системах с ограниченными вычислительными ресурсами [13–16, 19, 20] или, как правило, в менее мощных периферийных устройствах [17, 21–23]. Эти требования к производительности и памяти становятся особенно важными в парадигме интернета вещей, в которой вычисления выполняются локально на процессорах, а не на серверных машинах [21, 24, 25].

На практике наиболее часто встречаются изображения с шириной, не являющейся степенью двойки. Стандартный алгоритм Брейди – Ёна неприменим к таким изображениям. В литературе были предложены обобщения алгоритма Брейди – Ёна для изображений произвольного размера [26–28]. Однако эти подходы все еще не могут быть внедрены в системы с жесткими аппаратными ограничениями и требованиями к реальному времени. Предложенные алгоритмы неэффективно используют память и не соответствуют строгим требованиям по скорости и объему выделяемой памяти [21, 24, 25, 29, 30]. Избыточная дополнительная память, требуемая алгоритмами для их корректного исполнения, остается преградой для применения этих алгоритмов к большим изображениям на портативных устройствах, что подчеркивает критическую необходимость разработки быстрых in-place алгоритмов для вычисления БПХ.

In-place алгоритм по определению не использует дополнительную память для обработки входных данных, но может требовать небольшого объема памяти, который может быть как постоянным $\mathcal{O}(1)$, так и непостоянным, например, $\mathcal{O}(\log n)$, для работы с входными данными размера n [31, 32]. Иногда допускается использование памяти, меньшей, чем $\mathcal{O}(n)$. In-place алгоритмы часто работают быстрее, чем их out-of-place аналоги [32], поскольку они оптимизируют использование памяти [33, 34], снижают накладные расходы на копирование данных [32, 33] и повышают эффективность работы с кэшем [32, 35]. Изменяя данные непосредственно в исходном месте в памяти, in-place алгоритмы уменьшают необходимость в дополнительном выделении памяти, что может привести к снижению количества операций с памятью и улучшению локальности данных в кэше. Последнее, в свою очередь, снижает задержки доступа и ускоряет выполнение алгоритма [32, 35]. In-place подход переиспользования исходно выделенной памяти для промежуточных вычислений особенно выгоден в условиях ограниченности ресурсов, когда минимизация объема используемой памяти имеет решающее значение. Out-of-place алгоритмы могут обращаться к памяти чрезмерное количество раз и могут привести к существенно-

му замедлению выполнения алгоритма из-за обмена данными между оперативной памятью и диском [36]. Кроме того, in-place алгоритмы избегают сложностей, связанных с управлением множественными копиями данных, что снижает накладные расходы на выделение памяти, инициализацию и очистку [32]. Среди известных примеров in-place алгоритмов можно выделить алгоритм Кули – Тьюки для вычисления одномерного быстрого преобразования Фурье (БПФ) [37], а также in-place версию алгоритма Брейди – Ёна для БПХ для изображений с ширинами, являющимися степенями двойки [38].

В статье [27] исследуются два алгоритма БПХ для изображений произвольного размера – алгоритмы $FHT2DS$ и $FHT2DT$. Показано, что с точки зрения количества выполняемых суммирований алгоритм $FHT2DS$, первоначально предложенный в работе [26], является более быстрым. Использование in-place модификации алгоритма $FHT2DS$ может привести к дополнительным улучшениям как в вычислительной скорости, так и в объеме вспомогательной памяти, что видится шагом на пути к его широкому использованию во встраиваемых устройствах или системах в интернете вещей. В настоящей статье предлагается алгоритм $FHT2IDS$ – in-place модификация алгоритма $FHT2DS$. Представлены теоретические и экспериментальные исследования свойств данного алгоритма.

Статья имеет следующую структуру. Параграф 2 посвящен описанию алгоритма $FHT2DS$ и перечислению его ключевых характеристик. В § 3 вводится графовое представление алгоритма $FHT2DS$, в § 4 рассматривается in-place свойство алгоритмов с точки зрения теории графов. Доказательства важных свойств вычислительного графа алгоритма $FHT2DS$, основанные на понятиях и определениях из предыдущих параграфов, изложены в § 5. Доказанные свойства позволяют описать in-place алгоритм $FHT2IDS$ в § 6. Далее, в § 7 приводятся экспериментальные оценки производительности предложенного алгоритма. Наконец, в § 8 представлено обсуждение результатов, с выделением открытых вопросов и проблем, связанных с in-place алгоритмами, подобными $FHT2IDS$, которые требуют дальнейшего решения. Заключение представлено в § 9.

§ 2. Описание алгоритма $FHT2DS$ и его характеристики

Алгоритм $FHT2DS$ рекурсивно делит входное изображение

$$I = I_{w\times h}\colon \mathbb{Z}_w \times \mathbb{Z}_h \to \mathbb{A}$$

размера $w \times h$ на левое и правое подызображения I_L и I_R размеров $w_L \times h$ и $w_R \times h$ соответственно, где $w_L = \lfloor w/2 \rfloor$ и $w_R = w - \lfloor w/2 \rfloor$ [26–28]. Здесь $\mathbb{A}$ обозначает произвольную абелеву группу. На каждом шаге рекурсии вычисляется преобразование Хафа для подызображений I_L и I_R, – результаты вычислений обозначены J_L и J_R. Затем Хаф-образы J_L и J_R объединяются (сливаются) для формирования полного преобразования Хафа изображения I. Вектор $J(t,:)$ вычисляется как поэлементная сумма векторов $J_L(t_L,:)$ и $Concat(J_R(t_R, s:h), J_R(t_R, 0:s))$, где

$$s = (t - t_R) \bmod h, \quad t_L = [k_L t], \quad t_R = [k_R t],$$
$$k_L = (w_L - 1)/(w - 1), \quad k_R = (w_R - 1)/(w - 1).$$

Здесь для обращения к поддиапазонам векторов используется нотация среза, т.е. $n_1 : n_2$ обозначает диапазон от n_1 до n_2 (включая n_1, не включая n_2). Отсутствие обоих индексов указывает на полный диапазон. Функция $[\cdot]$ округляет вещественное число x до ближайшего целого числа $[x]$, при этом $[m + 1/2] = m$ для $m \in \mathbb{Z}$. В рамках алгоритмов, представленных в статье, Хаф-образ J использует так называемую st-параметризацию [26]: значение пикселя $J(t, s)$ равняется аппроксимации

интеграла входного одноканального изображения вдоль прямой

$$y(x) = s + \frac{t}{w-1}x, \quad (t,s) \in \mathbb{Z}_w \times \mathbb{Z}_h,$$

в декартовой системе координат Oxy, где оси Ox и Oy направлены вдоль ширины и высоты изображения соответственно [12].

Псевдокод алгоритма $FHT2DS$ представлен ниже [26–28]. В данной статье всякий алгоритм понимается как совокупность правил, которым необходимо следовать при выполнении вычислений, а псевдокод служит средством спецификации этого набора упорядоченных правил. Алгоритм однозначно определяет вычислительный граф, однако один и тот же вычислительный граф может соответствовать множеству различных алгоритмов, воплощающих одну и ту же идею, приводящих к одинаковому результату, но работающих с входными данными различными способами и следуя разным последовательностям правил. После того как алгоритм A представлен, его сложность по объему вспомогательной памяти, обозначенная через $M_A(n)$, где n – размер входных данных, определяется как минимально возможный размер массива, который должен быть выделен при выполнении алгоритма для получения корректного результата (в частности, результаты новых вычислений не должны перезаписывать массив, который все еще используется в вычислениях). При этом предполагается, что как только массив перестает использоваться в вычислениях (например, после завершения обработки рекурсивного вызова), он очищается, и память, необходимая для его хранения, освобождается. В данной статье свойства алгоритма в общем случае отличаются от свойств его конкретных реализаций, которые не рассматриваются. Так, например, при обсуждении сложности по объему вспомогательной памяти реализации, в отличие от сложности рекурсивного алгоритма, следует учитывать память, необходимую для хранения стека рекурсивных вызовов.

В алгоритме 12 функция $CreateZeroedImage(w,h)$ инициализирует изображение нулями. Функция $Concat(v_1,v_2)$ выполняет конкатенацию двух векторов v_1 и v_2.

Алгоритм 12 Алгоритм $FHT2DS$ для вычисления преобразования Хафа для изображений произвольного размера [26–28]

Input: $w > 0$, $h > 0$, изображение $I = I_{w\times h}$
Output: Хаф-образ $J = J_{w\times h}$
if $w > 1$ **then**
    $w_L \leftarrow \lfloor w/2 \rfloor$
    $w_R \leftarrow w - w_L$
    $I_L \leftarrow I(0 : w_L, :)$ ▷ I_L – область изображения I, память не выделяется
    $I_R \leftarrow I(w_L : w, :)$ ▷ I_R – область изображения I, память не выделяется
    $J_L \leftarrow FHT2DS(w_L, h, I_L)$ ▷ J_L является изображением
    $J_R \leftarrow FHT2DS(w_R, h, I_R)$ ▷ J_R является изображением
    $J \leftarrow CreateZeroedImage(w,h)$ ▷ для изображения J выделяется массив размера wh
    $k_L \leftarrow (w_L - 1)/(w-1)$
    $k_R \leftarrow (w_R - 1)/(w-1)$
    for $t \leftarrow 0$ **to** $w-1$ **do**
        $t_L \leftarrow [t\,k_L]$
        $t_R \leftarrow [t\,k_R]$
        $s \leftarrow (t - t_R) \bmod h$
        $J(t,:) \leftarrow J_L(t_L,:) + Concat(J_R(t_R, s:h), J_R(t_R, 0:s))$
else
    $J \leftarrow I$
return $J = J_{w\times h}$

Деля на каждом шаге рекурсии обрабатываемое изображение на левое и правое подызображения алгоритм $FHT2DS$ следует схеме "разделяй и властвуй" – это обеспечивает быстрое вычисление преобразования Хафа. На самом деле, было доказано, что алгоритм $FHT2DS$ демонстрирует следующую вычислительную сложность $T_{DS}(w,h)$ [27, 28], которая рассчитывается как количество выполненных операций в группе $\mathbb{A}$:

$$T_{DS}(w,h) = (\lfloor \log w \rfloor + 2)wh - 2^{\lfloor \log w \rfloor + 1}h \leqslant \frac{5\log_3 2}{3}\, wh\log w < 1{,}052\, wh\log w,$$

где $\dfrac{5\log_3 2}{3}$ – это точная константа в асимптотике вида $\mathrm{const}\cdot wh\log w$, т.е.

$$\sup_{w,h} \frac{T_{DS}(w,h)}{wh\log w} = \frac{5\log_3 2}{3},$$

$\log_2 x \equiv \log x$. Более того, имеет место асимптотическая эквивалентность [27, 28]:

$$T_{DS}(w,h) \sim wh\log w$$

при $w \to \infty$, т.е.

$$\lim_{w\to\infty} \frac{T_{DS}(w,h)}{wh\log w} = 1$$

для любой зависимости $h = h(w)$. Поскольку асимптотическая вычислительная сложность алгоритма Брейди–Ёна $T_{BY}(w,h) = wh\log w$, при $w = 2^q$, $q \in \mathbb{N}$, является оптимальной и не может быть улучшена [39], алгоритм $FHT2DS$ представляет собой обобщение алгоритма Брейди–Ёна для изображений с шириной, не являющейся степенью двойки, и это обобщение невозможно модифицировать с точки зрения асимптотической вычислительной сложности.

На каждом шаге рекурсии алгоритм $FHT2DS$ требует аллоцирования массива размера wh для вычисления Хаф-образа J с помощью J_L и J_R, вместо повторного использования массивов J_L и J_R. Иными словами, при каждом рекурсивном вызове алгоритм $FHT2DS$ требует выделения массива размера, равного размеру Хаф-образа, который вычисляется в данном вызове. Это приводит к тому, что сложность по вспомогательной памяти $M_{DS}(w,h)$ алгоритма $FHT2DS$ при обработке изображения размера $w \times h$ составляет $M_{DS}(w,h) = \mathcal{O}(wh\log w)$. Данный факт представляет собой препятствие для вычисления преобразования Хафа с использованием алгоритма $FHT2DS$ на устройствах с ограниченной памятью.

Наша цель – разработать in-place модификацию алгоритма $FHT2DS$ под названием $FHT2IDS$, которая, сохраняя рекурсивную схему алгоритма $FHT2DS$, требует выделения массива размера не более $w + h$ на каждом шаге рекурсии и при этом сохраняет вычислительную сложность $T_{IDS}(w,h) = T_{DS}(w,h)$ алгоритма $FHT2DS$.

§ 3. Вычислительный граф алгоритма $FHT2DS$

На каждом шаге рекурсии мы ассоциируем процедуру суммирования вектор-столбцов двух отдельных Хаф-образов, описанную в строках 13–17 алгоритма 12 и условно именуемую merge-этапом (merge-операцией или merge-процедурой в рамках одного рекурсивного вызова) алгоритма $FHT2DS$, с диаграммой потока данных (вычислительным графом), которая представлена ориентированным ациклическим графом

$$\Gamma = (V_\Gamma, E_\Gamma).$$

Множество вершин

$$V_\Gamma = \{L(i)\}_{i\in\mathbb{Z}_{w_L}} \cup \{R(i)\}_{i\in\mathbb{Z}_{w_R}} \cup \{C(i)\}_{i\in\mathbb{Z}_w}$$

состоит из вершин $L(i)$ (*левая* вершина) и $R(i)$ (*правая* вершина), хранящих векторы $J_L(i,:)$ и $J_R(i,:)$, а также вершин $C(t)$, в которых сохраняется объединенный вектор, полученный из $J_L(t_L,:)$ и $J_R(t_R,:)$,

$$t_L = [k_L t], \quad t_R = [k_R t], \quad k_L = (w_L - 1)/(w-1), \quad k_R = (w_R - 1)/(w-1),$$
$$w_L = \lfloor w/2 \rfloor, \quad w_R = w - \lfloor w/2 \rfloor.$$

Операции сложения векторов-столбцов производятся согласно направлениям ребер

$$E_\Gamma = \{(L(t_L), C(t))\}_{t\in\mathbb{Z}_w} \cup \{(R(t_R), C(t))\}_{t\in\mathbb{Z}_w}, \tag{1}$$
$$t_L = t_L(t) = [k_L t], \quad t_R = t_R(t) = [k_R t], \tag{2}$$
$$k_L = (w_L - 1)/(w-1), \quad k_R = (w_R - 1)/(w-1), \tag{3}$$
$$w_L = \lfloor w/2 \rfloor, \quad w_R = w - \lfloor w/2 \rfloor. \tag{4}$$

Отметим, что Γ является двудольным графом, поскольку все ребра $e \in E_\Gamma$ начинаются в $\{L(i)\}_{i\in\mathbb{Z}_{w_L}} \cup \{R(i)\}_{i\in\mathbb{Z}_{w_R}}$ и заканчиваются в $\{C(i)\}_{i\in\mathbb{Z}_w}$. Примеры описанного графа Γ, который управляет порядком исполнения merge-процедур в алгоритме $FHT2DS$, приведены на рис. 1.

Уже можно отметить, что вычислительный граф алгоритма $FHT2DS$ для четных значений ширины w напоминает вычислительный граф алгоритма Кули – Тьюки для БПФ по основанию 2 с прореживанием по времени (radix-2 decimation-in-time fast Fourier transform FFT, или radix-2 DIT FFT) [37], который используется для быстрого вычисления одномерного преобразования Фурье. Вычислительный граф алгоритма $FHT2DS$ для четных w представляет собой объединение непересекающихся подграфов, называемых бабочками, что аналогично структуре вычислительного графа для БПФ с основанием 2. Далее в § 6 будет дано точное определение графа типа бабочки, а также объяснение того, что вычислительный граф алгоритма $FHT2DS$ для четных значений w изоморфен вычислительному графу алгоритма Кули – Тьюки для БПФ с основанием 2. Это объяснение основывается на теоремах, представленных в § 5.

§ 4. Графовая интерпретация in-place свойства для алгоритма $FHT2DS$

Алгоритм $FHT2DS$ на каждой итерации предварительно выделяет память для массива J размера $w \times h$, в котором будут храниться суммы векторов $J_L(t_L,:)$ и $J_R(t_R,:)$ после исполнения merge-процедуры. Вместо предварительного выделения массива J мы можем хранить суммы векторов непосредственно в уже аллоцированных вершинах $\{L(i)\}_{i\in\mathbb{Z}_{w_L}} \cup \{R(i)\}_{i\in\mathbb{Z}_{w_R}}$, которые далее уже не будут задействованы при вычислении сумм остальных векторов. Такой подход потребует меньших затрат дополнительной рабочей памяти при условии, что нам известен порядок векторов, хранимых в $\{L(i)\}_{i\in\mathbb{Z}_{w_L}} \cup \{R(i)\}_{i\in\mathbb{Z}_{w_R}}$, а также порядок вершин $C(i)$, в которые последовательно будем записывать результаты суммирования векторов. Правильный порядок вычисления вершин $C(i)$ является принципиально важным для разработки модификации in-place алгоритма $FHT2DS$.

Математически задачу правильного обхода вершин $C(t)$ в in-place алгоритме можно сформулировать следующим образом. Рассмотрим биекцию

$$K\colon \{C(i)\}_{i\in\mathbb{Z}_w} \to \{L(i)\}_{i\in\mathbb{Z}_{w_L}} \cup \{R(i)\}_{i\in\mathbb{Z}_{w_R}}$$

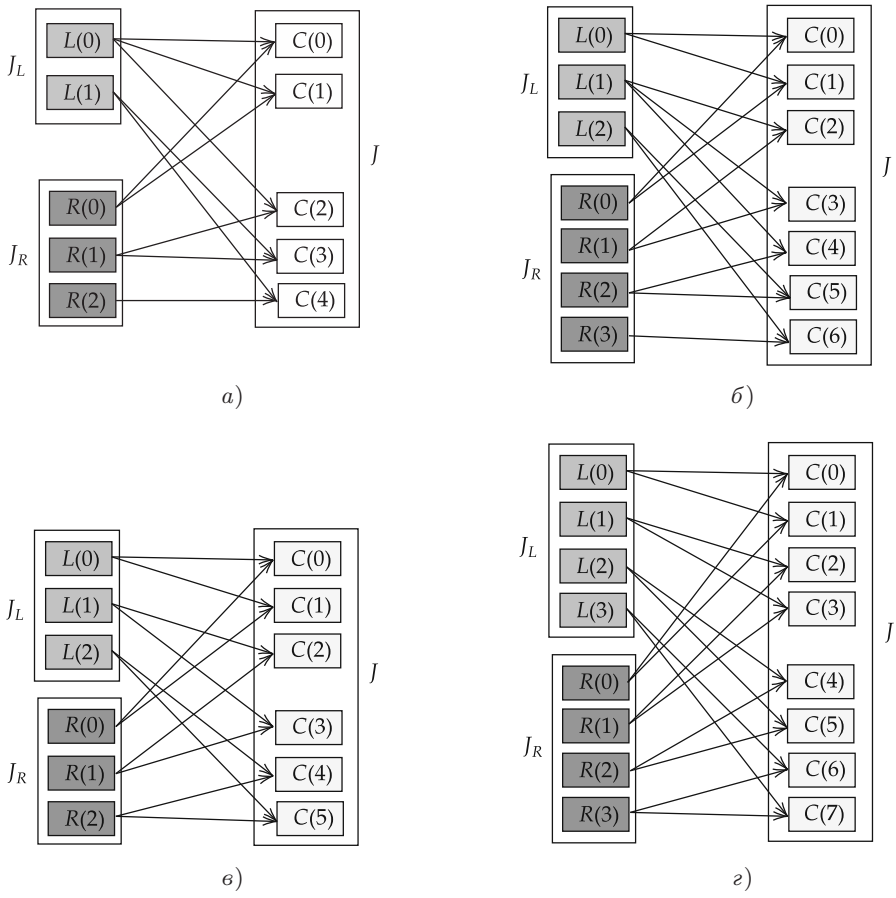


Рис. 1. Примеры вычислительных графов Γ алгоритма *FHT2DS* для различных значений ширины входного изображения w: *а*) $w = 5$, *б*) $w = 6$, *в*) $w = 7$, *г*) $w = 8$

и граф

$$\Gamma/K = (V_{\Gamma/K}, E_{\Gamma/K}),$$

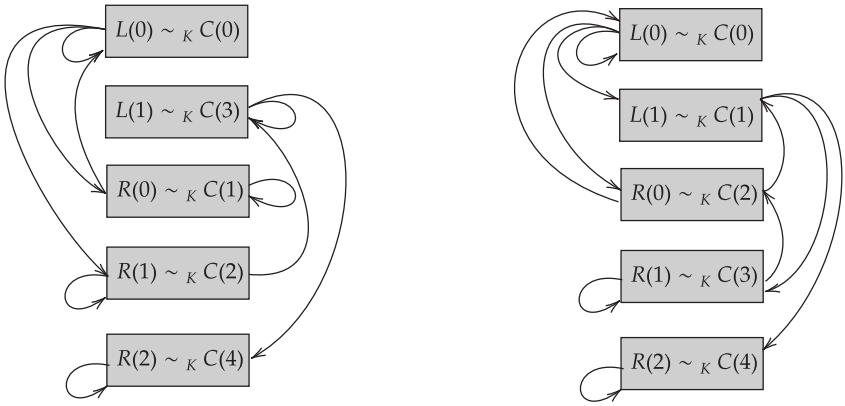
факторизованный следующим отношением эквивалентности вершин графа Γ

$$\forall v_1, v_2 \in V_\Gamma : \; v_1 \sim_K v_2 \iff v_1 = K(v_2) \vee v_2 = K(v_1).$$

Эквивалентно, мы получаем вершины $V_{\Gamma/K}$ после склеивания вершин V_Γ согласно отображению K, а ребра $E_{\Gamma/K}$ наследуются от ребер E_Γ:

$$V_{\Gamma/K} = \{v \in V_\Gamma \mid \forall v_1, v_2 \in V : \; v_1 \sim_K v_2 \Rightarrow v_1 = v_2\},$$
$$E_{\Gamma/K} = \{e \in E_\Gamma \mid \forall v_1', v_2', v_1'', v_2'' \in V_\Gamma : \; v_1' \sim_K v_1'', \; v_2' \sim_K v_2'' \; \Rightarrow \; (v_1', v_2') = (v_1'', v_2'')\}.$$

Пусть $\pi_\alpha \colon \mathbb{Z}_w \to 2^{\mathbb{Z}_w}$ – разбиение (ранга α) множества $\{0, 1, \ldots, w-1\}$ на подмножества, каждое из которых содержит не более $\alpha \in \mathbb{Z}_w$, $\alpha > 0$, элементов: для любого $i \in \mathbb{Z}_w$ выполняется $0 \leqslant |\pi_\alpha(i)| \leqslant \alpha$, $\bigcup \operatorname{Im} \pi_\alpha = \{0, 1, \ldots, w-1\}$ и $\pi_\alpha(i) \cap \pi_\alpha(j) = \varnothing$ для любых $i \neq j$.



а) *б*)

Рис. 2. Примеры вычислительных *FHT2DS* фактор-графов Γ/K для различных биекций $K \in \{K_{(a)}, K_{(b)}\}$ и соответствующих эквивалентностей $\sim_K$ на множестве вершин V_Γ, $w = 5$: *а*) Граф Γ с биекцией $K_{(a)}$, обладающий in-place свойством ранга 2: соответствующая функция разбиения задана как $\pi_2(0) = \{4\}$, $\pi_2(1) = \{3\}$, $\pi_2(2) = \{2\}$, $\pi_2(3) = \{0, 3\}$, $\pi_2(4) = \varnothing$. Пара $(K_{(a)}, \pi_2)$ обеспечивает in-place свойство ранга 2 для графа Γ; *б*) Пример графа Γ, который не обладает in-place свойством ранга 2 с биективной функцией $K_{(b)}$, т.е. не существует разбиения π_2 ранга 2, такого что пара $(K_{(b)}, \pi_2)$ обеспечивает in-place свойство ранга 2 для графа Γ. Заметим, что пара $(K_{(b)}, \pi_4)$ с $\pi_4(0) = \{4\}$, $\pi_4(1) = \{0, 1, 2, 3\}$, $\pi_4(2) = \pi_4(3) = \pi_4(4) = \varnothing$ гарантирует in-place свойство ранга 4 для графа Γ

Мы ищем пару (K, π_α), обеспечивающую следующее *in-place свойство* (*ранга* α) для графа Γ: для графа Γ/K и для любого $m \in \mathbb{Z}_w$, после удаления вершин

$$C(i) \in V_{\Gamma/K}, \quad i \in \bigcup \{\pi_\alpha(j) \mid 0 \leqslant j \leqslant m\},$$

и ребер из $E_{\Gamma/K}$, заканчивающихся в этих вершинах, результирующее множество оставшихся вершин Γ/K остается подграфом исходного графа Γ/K. В этом случае граф Γ, мы будем говорить, *обладает in-place свойством* (*ранга* α), *обеспеченным* (*гарантированным*) *парой* (K, π_α) (см. примеры на рис. 2). Когда индекс α не указан, его значение подразумевается равным 2.

Практически in-place свойство ранга α ($\alpha > 1$) для графа Γ означает, что алгоритм с вычислительным графом Γ может быть выполнен в in-place режиме, требующем выделения массива размера не более $(\alpha - 1)h$ для проведения промежуточных вычислений:

1. Последовательно для $0 \leqslant i \leqslant w - 1$ рассматриваются вершины $K(\mathrm{prev}(C(j))) \subset \subset V_\Gamma$, где $j \in \pi_\alpha(i)$, которые составляют образ множества $\mathrm{prev}(C(j))$ при отображении K;
2. Если $\pi_\alpha(i) \neq \varnothing$, выполняется суммирование векторов (merge-процедура), хранящихся в вершинах $K(\mathrm{prev}(C(j)))$ (при этом $|\mathrm{prev}(C(j))| = 2$ согласно уравнению (1));
3. После суммирования (merge-процедуры) векторов производится сохранение результата в вершинах $K(C(j)) \in V_\Gamma$, $j \in \pi_\alpha(i)$; поскольку $|\mathrm{prev}(C(j))| \leqslant \alpha$, эти вычисления могут быть выполнены не более чем с $\alpha - 1$ дополнительными временными массивами длины h.

Здесь, а также далее для произвольного ориентированного графа $\gamma = (V_\gamma, E_\gamma)$, если $C \in V_\gamma$, то

$$\mathrm{prev}(C) = \{v \in V_\gamma \mid (v, C) \in E_\gamma\}.$$

Также

$$\mathrm{next}(C) = \{v \in V_\gamma \mid (C, v) \in E_\gamma\}.$$

Таким образом, для построения in-place модификации алгоритма $FHT2DS$ необходимо найти соответствующую биективную функцию K и соответствующую функцию разбиения π_α. Описание таких функций требует более детального рассмотрения структуры вычислительного графа $FHT2DS$, что и является основным вопросом следующего параграфа.

§ 5. Анализ структуры вычислительного графа алгоритма $FHT2DS$

Здесь мы доказываем несколько замечательных свойств вычислительного графа алгоритма $FHT2DS$ (см. § 3). Эти свойства служат основой для обоснования корректности далее предложенного in-place алгоритма $FHT2IDS$.

Теорема 1. *Для любого* $t_L \in \mathbb{Z}_{w_L}$

$$\mathrm{next}(L(t_L)) = \left\{ C(t) \;\middle|\; \left\lfloor \frac{t_L - 1/2}{k_L} + 1 \right\rfloor \leqslant t \leqslant \left\lfloor \frac{t_L + 1/2}{k_L} \right\rfloor \wedge t \in \mathbb{Z}_w \right\}, \tag{5}$$

где $k_L = (w_L - 1)/(w - 1)$, $w_L = \lfloor w/2 \rfloor$, $w > 3$.

Также для любого $t_R \in \mathbb{Z}_{w_R}$

$$\mathrm{next}(R(t_R)) = \left\{ C(t) \;\middle|\; \left\lfloor \frac{t_R - 1/2}{k_R} + 1 \right\rfloor \leqslant t \leqslant \left\lfloor \frac{t_R + 1/2}{k_R} \right\rfloor \wedge t \in \mathbb{Z}_w \right\}, \tag{6}$$

где $k_R = (w_R - 1)/(w - 1)$, $w_R = w - \lfloor w/2 \rfloor$, $w > 2$.

Доказательство. Если $C(t) \in \mathrm{next}(L(t_L))$ для некоторого $t \in \mathbb{Z}_w$, $t_L \in \mathbb{Z}_{w_L}$, можем записать

$$t_L = [k_L t]$$

согласно устройству алгоритма $FHT2DS$ (см. § 2). Отсюда следует, что

$$t_L - 1/2 < k_L t \leqslant t_L + 1/2 \iff \frac{t_L - 1/2}{k_L} < t \leqslant \frac{t_L + 1/2}{k_L},$$

$k_L > 0$ при $w > 3$. Так как $t \in \mathbb{Z}_w$ – целое число, мы заключаем, что

$$\left\lfloor \frac{t_L - 1/2}{k_L} + 1 \right\rfloor \leqslant t \leqslant \left\lfloor \frac{t_L + 1/2}{k_L} \right\rfloor,$$

что и доказывает равенство (5).

В силу соотношения $t_R = [k_R t]$ равенство (6) доказывается аналогично. ▲

Далее мы будем обозначать

$$\deg^+(v) = |\{e \in E_\Gamma \mid e = (v, x) \wedge x \in V_\Gamma\}|$$

– *выходная степень* (*или степень исхода*) вершины $v \in V_\Gamma$ в ориентированном графе Γ,

$$\deg^-(v) = |\{e \in E_\Gamma \mid e = (x, v) \wedge x \in V_\Gamma\}|$$

будет обозначать *входную степень* (*или степень захода*) вершины $v \in V_\Gamma$ в ориентированном графе Γ, а

$$\deg(v) = \deg^+(v) + \deg^-(v)$$

– (*полная*) *степень* вершины $v \in V_\Gamma$.

Теорема 2. *Для всякого $t_L \in \mathbb{Z}_{w_L}$ справедливо неравенство*

$$2 \leqslant \deg^+(L(t_L)) \leqslant 3. \tag{7}$$

Более того, для всякого $t_R \in \mathbb{Z}_{w_R}$ имеет место неравенство

$$1 \leqslant \deg^+(R(t_R)) \leqslant 2. \tag{8}$$

Отметим, что для всякого $t \in \mathbb{Z}_w$

$$\deg^-(C(t)) = 2. \tag{9}$$

Доказательство. Поскольку вычислительный граф алгоритма *FHT2DS* не содержит кратных ребер, для доказательства неравенства (7) достаточно показать, что для всех $t_L \in \mathbb{Z}_{w_L}$ выполняется следующее неравенство:

$$2 \leqslant |\mathrm{next}(L(t_L))| \leqslant 3.$$

Для неравенства (8) необходимо показать, что, другими словами,

$$1 \leqslant |\mathrm{next}(R(t_R))| \leqslant 2.$$

Рассмотрим отношение $t_L = [k_L t]$, которое означает $C(t) \in \mathrm{next}(L(t_L))$. Заметим, что

$$k_L = \frac{\lfloor w/2 \rfloor - 1}{w - 1} < 1/2$$

при $w > 3$. Это означает, что любое множество вида $(k-1/2, k+1/2]$, $k \in \mathbb{Z}$, содержит хотя бы два различных элемента из последовательности $\{k_L n\}_{n \in \mathbb{Z}_w}$. Следовательно, $|\mathrm{next}(L(t_L))| \geqslant 2$ для $w > 3$. Построив графы Γ для случаев $w \in \{1, 2, 3\}$, можно убедиться, что последнее неравенство справедливо для $w \geqslant 1$.

Аналогично, рассматривая соотношение $t_R = [k_R t]$, оценка

$$k_R = \frac{w - \lfloor w/2 \rfloor - 1}{w - 1} \leqslant \frac{1}{2}$$

при $w > 3$ обосновывает тот факт, что любое множество вида $(k - 1/2, k + 1/2]$, $k \in \mathbb{Z}$, содержит хотя бы один элемент из последовательности $\{k_R n\}_{n \in \mathbb{Z}_w}$. Следовательно, $|\mathrm{next}(R(t_R))| \geqslant 1$ для $w > 3$. Построив графы Γ для случаев $w \in \{1, 2, 3\}$, можно убедиться, что неравенство $|\mathrm{next}(R(t_R))| \geqslant 1$ остается верным также для $w \in \{1, 2, 3\}$.

Далее зафиксируем $t_L \in \mathbb{Z}_{w_L}$, и пусть $t = t' \in \mathbb{Z}_{w-3}$ удовлетворяет соотношению $t_L = [k_L t]$. Мы докажем, что $t = t' + 3$ не может быть решением уравнения $t_L = [k_L t]$. Это привело бы к неравенству $|\mathrm{next}(L(t_L))| \leqslant 3$ (здесь используется монотонность функции $t_L(t) = [k_L t]$, т.е. $t'' > t' + 3$ не может быть решением уравнения $t_L = [k_L t]$, если $t' + 3$ не является его решением). Действительно, из $t_L = [k_L t']$ получаем

$$k_L t' > t_L - 1/2.$$

Далее, для $t = t' + 3$ имеем

$$k_L(t' + 3) > t_L + 3k_L - 1/2. \tag{10}$$

Заметим, что

$$t_L + 3k_L - 1/2 \geqslant t_L + 1/2 \iff k_L = \frac{\lfloor w/2 \rfloor - 1}{w-1} \geqslant 1/3 \tag{11}$$

при $w \geqslant 10$. Из уравнений (10) и (11) получаем

$$k_L(t'+3) > t_L + 1/2,$$

что означает

$$[k_L(t'+3)] \geqslant t_L + 1,$$

т.е. $t = t' + 3$ не является решением уравнения $t_L = [k_L t]$, и $|\mathrm{next}(L(t_L))| \leqslant 3$ при $w \geqslant 10$. Это неравенство остается верным и для $w < 10$, что подтверждается непосредственным построением графов Γ для $w < 10$.

Далее мы докажем, что $|\mathrm{next}(R(t_R))| \leqslant 2$ при $t_R \in \mathbb{Z}_{w_R}$.

Отметим, что выполняются следующие неравенства:

$$\frac{t}{2} - \frac{1}{2} < k_R t \leqslant \frac{t}{2}, \quad t < w-1, \tag{12}$$

и

$$\frac{t}{2} - \frac{1}{2} \leqslant k_R t \leqslant \frac{t}{2}, \quad t = w-1. \tag{13}$$

Действительно, оценки сверху в неравенствах (12) и (13) верны, коль скоро $k_R \leqslant \frac{1}{2}$.

Теперь поясним оценки снизу в неравенствах (12) и (13). Когда $w = 1 \bmod 2$, имеем $k_R = \frac{1}{2}$, и следовательно,

$$k_R t = \frac{t}{2} > \frac{t}{2} - \frac{1}{2}.$$

Когда $w = 0 \bmod 2$, $k_R < \frac{1}{2}$, имеем следующую цепочку неравенств для $t < w-1$:

$$\frac{t}{2} - k_R t < \frac{w-1}{2} - k_R(w-1) = \frac{w-1}{2} - (w_R - 1) = \frac{w-1}{2} - w_R + 1 =$$
$$= \frac{w}{2} - w_R + \frac{1}{2} = \frac{1}{2},$$

так как функция $f(t) = \frac{t}{2} - k_R t$ является возрастающей, а $w_R = \frac{w}{2}$.

Когда $w = 0 \bmod 2$, для $t = w-1$ имеем

$$\frac{w-1}{2} - \frac{1}{2} = \frac{w}{2} - 1 = w_R - 1 = k_R(w-1).$$

Таким образом, неравенства (12) и (13) доказаны. Из них следует, что каждый полуинтервал вида $(k/2 - 1/2, k/2]$, $k \in \mathbb{Z}_{w-1}$, содержит ровно одно число вида $k_R t$, а полуинтервал $((w-1)/2 - 1/2, (w-1)/2]$ содержит либо два числа вида $k_R t$ (когда $k_R < 1/2$ и $w = 0 \bmod 2$), либо одно число вида $k_R t$ (когда $k_R = 1/2$ и $w = 1 \bmod 2$). Поэтому в каждом множестве вида $(t_R - 1/2, t_R + 1/2]$, $t_R \in \mathbb{Z}_{w_R}$, содержится не более двух различных чисел вида $k_R t$, и следовательно, уравнение $t_R = [k_R t]$, $t_R \in \mathbb{Z}_{w_R}$, не имеет более двух различных решений, а значит, $|\mathrm{next}(R(t_R))| \leqslant 2$.

Таким образом, неравенства (7) и (8) доказаны.

Наконец, уравнение (1) объясняет соотношение (9). Теорема 2 полностью доказана. ▲

Следующие следствие и теорема характеризуют количество левых вершин $L(t_L)$ с выходной степенью 3 и правых вершин $R(t_R)$ с выходной степенью 1.

Следствие 1. *Число левых вершин с выходной степенью* 3 *равно числу правых вершин с выходной степенью* 1:

$$|\{L(t_L) \mid \deg^+(L(t_L)) = 3 \wedge t_L \in \mathbb{Z}_{w_L}\}| = \\ = |\{R(t_R) \mid \deg^+(R(t_R)) = 1 \wedge t_R \in \mathbb{Z}_{w_R}\}|. \tag{14}$$

Доказательство. Для графа Γ выполняется следующее стандартное тождество:

$$\sum_{v \in V_\Gamma} \deg(v) = 2|E_\Gamma|, \tag{15}$$

т.е. сумма степеней всех вершин графа равна удвоенному числу ребер.

Уравнение (15) можно переписать в следующем виде:

$$\sum_{t_L \in \mathbb{Z}_{w_L}} \deg^+(L(t_L)) + \sum_{t_R \in \mathbb{Z}_{w_R}} \deg^+(R(t_R)) + \sum_{t \in \mathbb{Z}_w} \deg^-(C(t)) = 2|E_\Gamma| \iff \tag{16}$$

$$\iff \sum_{t_L \in \mathbb{Z}_{w_L}} \deg^+(L(t_L)) + \sum_{t_R \in \mathbb{Z}_{w_R}} \deg^+(R(t_R)) + 2w = 4w \iff \tag{17}$$

$$\iff \sum_{t_L \in \mathbb{Z}_{w_L}} \deg^+(L(t_L)) + \sum_{t_R \in \mathbb{Z}_{w_R}} \deg^+(R(t_R)) = 2w. \tag{18}$$

Здесь мы воспользовались соотношением (9), а также тем фактом, что согласно архитектуре алгоритма *FHT2DS* справедливы следующие равенства для входных и выходных степеней вершин:

$$\deg^-(L(t_L)) = \deg^-(R(t_R)) = \deg^+(C(t)) = 0.$$

Кроме того,

$$|E_\Gamma| = 2|\{C(t) \mid t \in \mathbb{Z}_w\}| = 2w.$$

В силу теоремы 2 и уравнения (18), если $\deg^+(L(t_L)) = 3$ для некоторого $t_L \in \mathbb{Z}_{w_L}$, то по принципу Дирихле существует $t_R \in \mathbb{Z}_{w_R}$, для которого $\deg^+(R(t_R)) = 1$. И обратное утверждение также верно: как только $\deg^+(R(t_R)) = 1$ для некоторого $t_R \in \mathbb{Z}_{w_R}$, то по принципу Дирихле существует $t_L \in \mathbb{Z}_{w_L}$, для которого $\deg^+(L(t_L)) = 3$. Таким образом, тождество (14) доказано. ▲

Теорема 3. *Справедливы следующие утверждения*:

1. *Пусть* $w = 0 \bmod 2$. *Тогда все левые и правые вершины имеют выходную степень, равную* 2:

$$\deg^+(L(t_L)) = \deg^+(R(t_R)) = 2, \quad t_L \in \mathbb{Z}_{w_L}, \quad t_R \in \mathbb{Z}_{w_R}, \quad w = 0 \bmod 2. \tag{19}$$

2. *Пусть* $w = 1 \bmod 2$. *Тогда существует единственная левая вершина* $L([w/4]-1)$ *с выходной степенью* 3 *и единственная правая вершина* $R(w_R - 1)$ *с выходной степенью* 1. *Все остальные левые и правые вершины имеют выходную степень, равную* 2:

$$\deg^+(L([w/4] - 1)) = 3, \quad \deg^+(L(t_L)) = 2, \quad t_L \in \mathbb{Z}_{w_L} \setminus \{[w/4] - 1\}, \tag{20}$$

$$\deg^+(R(w_R - 1)) = 1, \quad \deg^+(R(t_R)) = 2, \quad t_R \in \mathbb{Z}_{w_R} \setminus \{w_R - 1\}, \tag{21}$$

$$w_R = w - \lfloor w/2 \rfloor, \quad w = 1 \bmod 2. \tag{22}$$

Доказательство. Во-первых, пусть $w = 0 \bmod 2$. Из неравенств (12) и (13) следует, что каждый полуинтервал $(k/2 - 1/2, k/2]$, $k \in \mathbb{Z}_{w-1}$, содержит одно число вида $k_R t$, в то время как полуинтервал $((w-1)/2 - 1/2, (w-1)/2]$ содержит ровно два числа вида $k_R t$. Следовательно, каждое множество вида $(t_R - 1/2, t_R + 1/2]$, $t_R \in \mathbb{Z}_{w_R}$, содержит ровно два числа из последовательности $\{k_R t\}_{t \in \mathbb{Z}_w}$, т.е. для любого $t_R \in \mathbb{Z}_{w_R}$ уравнение $t_R = [k_R t]$ имеет ровно два решения. Это эквивалентно тому, что $\deg^+(R(t_R)) = 2$ для всех $t_R \in \mathbb{Z}_{w_R}$, что, согласно следствию 1, также влечет $\deg^+(L(t_L)) = 2$ для всех $t_L \in \mathbb{Z}_{w_L}$. Первая часть текущей теоремы доказана.

Теперь положим $w = 1 \bmod 2$. В этом случае $k_R = 1/2$, и из неравенств (12) и (13) следует, что каждый полуинтервал $(k/2 - 1/2, k/2]$, $k \in \mathbb{Z}_w$, содержит ровно одно число вида $k_R t$, $t \in \mathbb{Z}_w$. Следовательно, полуинтервалы $(t_R - 1/2, t_R + 1/2]$, $t_R \in \mathbb{Z}_{w_R - 1}$, содержат два числа вида $k_R t$, а последний полуинтервал $(t_R - 1/2, t_R + 1/2]$, $t_R = w_R - 1$, содержит только одно такое число $w_R - 1$; $t \in \mathbb{Z}_w$. Другими словами, уравнение $t_R = [k_R t]$ имеет два решения, когда $t_R \neq w_R - 1$. Это доказывает, что существует только одна правая вершина $R(w_R - 1)$ графа Γ с выходной степенью 1. Согласно следствию 1 существует также единственная левая вершина с выходной степенью 3. Необходимо доказать, что ее порядковый номер равен $[w/4] - 1$.

Эквивалентно, проверим, что уравнение $[w/4] - 1 = [k_L t]$ имеет три различных решения для $t \in \mathbb{Z}_w$. Действительно, эти решения выражаются как

$$t \in \{2([w/4] - 1) + n \mid n = 0, 1, 2\}.$$

Если $w = 4k+1$, $k \in \mathbb{Z}$, $k > 0$, имеем $[w/4]-1 = k-1$, $w_L = \lfloor w/2 \rfloor = 2k$, $k_L = \dfrac{2k-1}{4k}$ и

$$k - \frac{3}{2} < \frac{(2k-1)(2k-2)}{4k} \leqslant k_L t = \frac{2k-1}{4k}(2k + n - 2) \leqslant k - \frac{1}{2} \Longrightarrow$$
$$\Longrightarrow [k_L t] = k - 1 = [w/4] - 1.$$

Если же $w = 4k - 1, k \in \mathbb{Z}$, $k > 0$, тогда $[w/4] - 1 = k - 1$, $w_L = \lfloor w/2 \rfloor = 2k - 1$, $k_L = \dfrac{k-1}{2k-1}$ и

$$k - \frac{3}{2} < \frac{(k-1)(2k-2)}{2k-1} \leqslant k_L t = \frac{k-1}{2k-1}(2k + n - 2) \leqslant \frac{2k(k-1)}{2k-1} \leqslant k - \frac{1}{2} \Longrightarrow$$
$$\Longrightarrow [k_L t] = k - 1 = [w/4] - 1.$$

Таким образом, $t \in \{2([w/4]-1)+n \mid n = 0, 1, 2\}$ действительно являются решениями уравнения $t_L = [k_L t]$, и можем заключить, что $\deg^+(L([w/4] - 1)) = 3$. Вторая часть теоремы 3 полностью доказана. ▲

Утверждение теоремы согласуется с рис. 1, на котором изображены вычислительные графы Γ алгоритма $FHT2DS$ для $w \in \{5, 6, 7, 8\}$.

В следующем параграфе, основываясь на доказанных структурных свойствах вычислительного графа алгоритма $FHT2DS$, мы представим его in-place модификацию – алгоритм $FHT2IDS$. Обоснование нового алгоритма также приведено в следующем параграфе.

§ 6. Описание in-place алгоритма *FHT2IDS*

Мы приводим описание нашего алгоритма $FHT2IDS$ – in-place модификации алгоритма $FHT2DS$, который был предложен ранее [26–28] и предназначен для вычисления БПХ для изображений произвольной ширины.

В алгоритме $FHT2IDS$ вершины $C(t)$ вычисляются в ходе выполнения последовательных merge-процедур, т.е. суммирования векторов, хранимых в левых $L(t_L)$ и правых $R(t_R)$ вершинах (см. § 3). Однако в отличие от алгоритма $FHT2DS$ алгоритм $FHT2IDS$ сохраняет результат либо в левой, либо в правой вершине, т.е. в той области памяти, где хранилось входное изображение I. При этом вектор, который ранее был сохранен в этой вершине, стирается и больше не может быть использован для дальнейших вычислений. Поэтому в алгоритме $FHT2IDS$ каждый результат вычислений должен быть сохранен в вершине, которая впоследствии не используется для вычислений других $C(t)$. Это требование накладывает существенные ограничения на порядок вычисления вершин $C(t)$. Важно отметить, что при таком обходе вершин столбцы результата преобразования Хафа могут вычисляться не по порядку (заданном st-параметризацией), в отличие от алгоритма $FHT2DS$. Однако даже при таком порядке обхода можно переставить столбцы выходного изображения так, чтобы оно совпало с результатом работы алгоритма $FHT2DS$.

Мы опишем правильный порядок обхода вычисляемых вершин $C(t)$, а также порядок сохранения результатов слияний (в ходе merge-процедур) векторов (т.е. столбцов Хаф-образов) из левых и правых вершин.

Рассмотрим случай четной ширины $w = 0 \bmod 2$. Согласно ключевой теореме 3 из предыдущего § 5 каждая левая вершина $L(t_L)$ и правая вершина $R(t_R)$ вычислительного графа имеют выходную степень, равную 2, в то время как входная степень каждой вершины $C(t)$ также равна 2. Кроме того, ребра, инцидентные левой/правой вершине с большим индексом t_L или t_R, входят в вершину $C(t)$ с большим индексом t. Таким образом, вычислительный граф алгоритма $FHT2DS$ в случае четной ширины w распадается на $w/2$ так называемых бабочек (изображенных на рис. 3). Более точно, бабочка здесь определяется как подграф $\mathcal{B}(k)$, $k \in \mathbb{Z}_{w/2}$, графа Γ и имеет следующий вид:

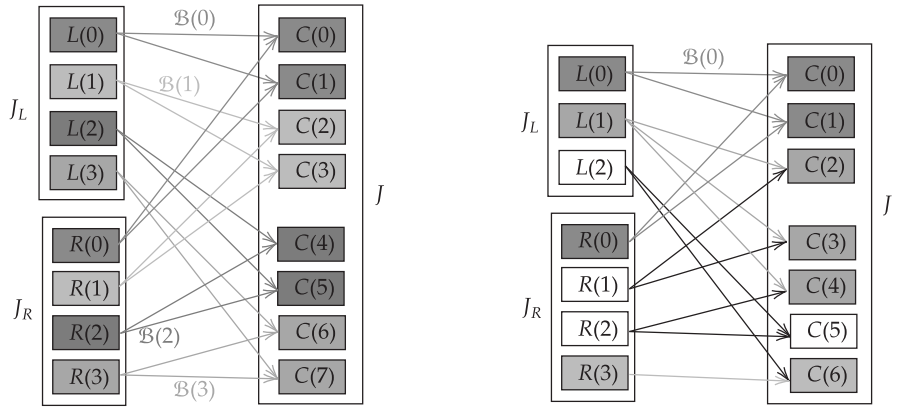
$$\mathcal{B}(k) = (V_{\mathcal{B}(k)}, E_{\mathcal{B}(k)}) \subset \Gamma,$$
$$V_{\mathcal{B}(k)} = \{L(k), R(k), C(2k), C(2k+1)\},$$
$$E_{\mathcal{B}(k)} = \{(L(k), C(i)), (R(k), C(i)) \mid i = 2k, 2k+1\}.$$

Для случая четной ширины w мы предлагаем обрабатывать все бабочки вычислительного графа Γ последовательно. При обработке отдельной бабочки $\mathcal{B}(k)$, $k \in \mathbb{Z}_{w/2}$, используется дополнительный временный массив размера $1 \times h$ (обозначаемый через tmp):

1. Сначала вычисляется вершина $C(k)$ (соответствующая некоторому значению параметра $t = 2k$ в алгоритме 13) по вершинам $L(k)$ и $R(k)$ (которые хранят вектора $J_L(k_L, :)$ и $J_R(k_R, :)$ в алгоритме 13 соответственно), и результат сохраняется во временном массиве tmp (см. алгоритм 13 для обработки структуры бабочки, строка 9);
2. Затем вычисляется вершина $C(2k+1)$ (соответствующая значению $t = 2k+1$) по вершинам $L(k)$ и $R(k)$ (которые содержат вектора $J_L(k_L, :)$ и $J_R(k_R, :)$ в алгоритме 13 соответственно), и результат сохраняется в $R(k)$ (эта вершина соответствует $J_R(k_R, :)$, см. алгоритм 13, строка 10);
3. Наконец, содержимое tmp записывается обратно в $L(k)$ (соответствующий вектор есть $J_L(k_L, :)$, алгоритм 13, строка 11).

Алгоритм, вычисляющий вершины в структуре отдельной бабочки, приведен в виде псевдокода 13.

Переформулировав алгоритм 13 в терминах, введенных в § 4 с графовой интерпретацией in-place свойства, можно заключить, что функция K, которая определяет левые и правые вершины для последовательной записи вычисляемых вершин $C(t)$, $t \in \mathbb{Z}_w$, отображает вершины $C(2k)$ в вершины $L(k)$, а вершины $C(2k+1)$ – в вер-



а) *б*)

Рис. 3. Пример структуры бабочек в вычислительном графе Γ алгоритма *FHT*2*DS*: *а*) $w = 8$; *б*) $w = 7$. Кроме бабочки $\mathcal{B}(0)$, также выделены вершины со степенями 3 и 1

Алгоритм 13 Алгоритм *ProcessButterfly*

Input: изображения J_L и J_R, их ширины w_L и w_R, они имеют равную высоту h, массивы индексов K_L и K_R, по которым восстанавливается правильный порядок столбцов изображений J_L и J_R (индуцированный *st*-параметризацией), индекс n бабочки $\mathcal{B}(n)$, которую необходимо вычислить, массив K, по которому восстанавливается правильный порядок столбцов изображения J (индуцированный *st*-параметризацией)
$t \leftarrow 2k$
$t_L \leftarrow k$
$t_R \leftarrow k$
$s_L \leftarrow (t - t_R) \bmod h$
$s_R \leftarrow (t + 1 - t_R) \bmod h$
$k_L \leftarrow K_L(t_L)$
$k_R \leftarrow K_R(t_R)$
$tmp \leftarrow J_L(k_L, :) + Concat(J_R(k_R, s_L : h), J_R(k_R, 0 : s_L))$ ▷ выделение массива размера h
$J_R(k_R, :) \leftarrow J_L(k_L, :) + Concat(J_R(k_R, s_R : h), J_R(k_R, 0 : s_R))$
$J_L(k_L, :) \leftarrow tmp$
$K(t) \leftarrow k_L$
$K(t + 1) \leftarrow w_R + k_R$

шины $R(k)$:

$$K(C(2k)) = L(k), \quad K(C(2k + 1)) = R(k), \quad k \in \mathbb{Z}_{w/2}. \tag{23}$$

Кроме того, разбиение π_2 ранга 2 множества $\mathbb{Z}_w$, которое определяет порядок перезаписи левых и правых вершин, для нашего алгоритма выражается следующей формулой:

$$\pi_2(k) = \{2k, 2k + 1\}, \quad k \in \mathbb{Z}_{w/2},$$
$$\pi_2(k) = \varnothing, \quad k \in \mathbb{Z}_w \setminus \mathbb{Z}_{w/2}.$$

Для наглядности порядок вычислений (merge-операций) в алгоритме *FHT*2*IDS* и порядок вершин, используемых для хранения результатов вычислений, определенный парой (K, π_2), приведены на рис. 4.

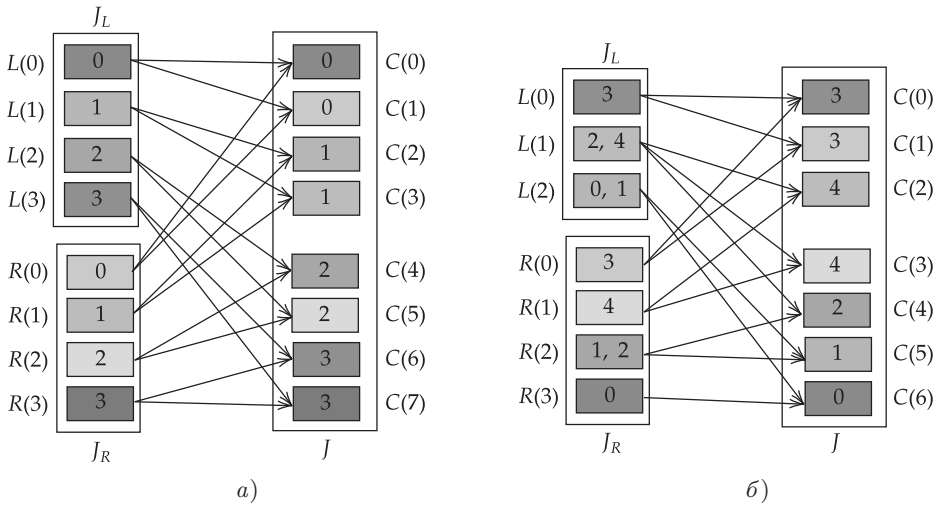


Рис. 4. Порядок merge-операций и сохранений в алгоритме $FHT2IDS$. Номера вершин указывают на этапы алгоритма, на которых они используются: *а*) $w = 8$. Биекция K задается следующим образом: $K(C(0)) = L(0)$, $K(C(1)) = R(0)$, $K(C(2)) = L(1)$, $K(C(3)) = R(1)$, $K(C(4)) = L(2)$, $K(C(5)) = R(2)$, $K(C(6)) = L(3)$, $K(C(7)) = R(3)$; *б*) $w = 7$. Биекция K задается следующим образом: $K(C(0)) = L(0)$, $K(C(1)) = R(0)$, $K(C(2)) = L(1)$, $K(C(3)) = R(1)$, $K(C(4)) = R(2)$, $K(C(5)) = L(2)$, $K(C(6)) = R(3)$

Теоремы из предыдущего § 5 доказывают, что такая пара (K, π_2) обеспечивает in-place свойство (ранга 2) для графа Γ: результаты вычислений алгоритмов $FHT2IDS$ и $FHT2DS$ для идентичных входных изображений, с учетом перестановки K их столбцов, совпадают. Каждый рекурсивный вызов (без обработки дочерних вызовов) выделяет массив размера не более чем $w' + h$: массив размера $w' \leqslant w$ для хранения массива для K и отдельный массив размера h для обработки всех бабочек.

Следует отметить, что тот факт, что граф Γ в случае четной ширины изображения является дизъюнктным объединением бабочек, делает наш алгоритм $FHT2IDS$ схожим с алгоритмом Кули – Тьюки для быстрого вычисления одномерного преобразования Фурье. Вычислительный граф алгоритма Кули – Тьюки для случая БПФ с декомпозицией по основанию 2 и с прореживанием по времени также представляется как объединение бабочек [37]. Таким образом, для случая четной ширины w вычислительный граф алгоритма $FHT2DS$ изоморфен вычислительному графу алгоритма Кули – Тьюки для одномерного БПФ с основанием 2 (в частности, степени всех вершин в соответствующих вычислительных графах алгоритмов совпадают).

Теперь опишем алгоритм $FHT2IDS$ для изображений с нечетной шириной $w = 1 \bmod 2$. Согласно ключевой теореме 3 из предыдущего параграфа, граф Γ имеет одну левую вершину $L(\lfloor w/4 \rfloor - 1)$ с выходной степенью 3 и одну правую вершину $R(w_R - 1)$ с выходной степенью 1, в то время как все остальные левые и правые вершины имеют выходную степень 2. Поскольку вершина $R(w_R - 1)$ участвует только в одном вычислении вершины $C(w - 1)$, мы предлагаем рассматривать ее первой и записывать результат merge-операции вершин $L(w_L - 1)$ и $R(w_R - 1)$ в нее. После этого мы мысленно удаляем вершины $R(w_R - 1)$ и $C(w - 1)$, а также ребра $(L(w_L - 1), C(w - 1))$ и $(R(w_R - 1), C(w - 1))$ из графа Γ (см. рис. 5, шаг 1).

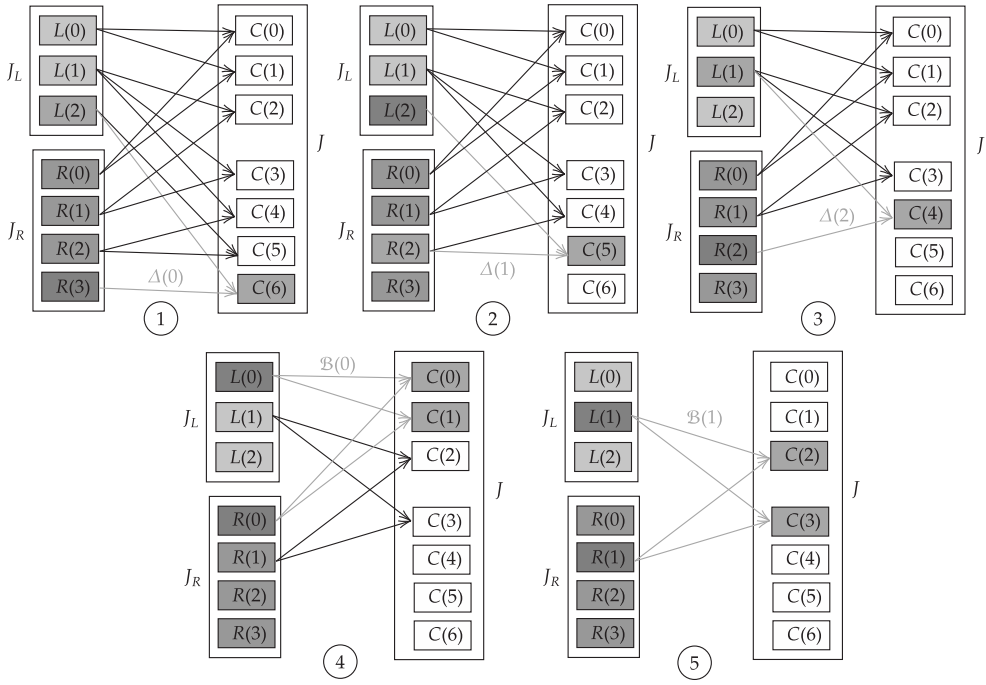


Рис. 5. Последовательные преобразования графа Γ в алгоритме *FHT2IDS* для случая $w = 7$. На каждом шаге рассматриваются вершины треугольника или бабочки. Результаты вычислений записываются в вершины, составляющие основание треугольника или бабочки

Процедура, выполняемая при обработке следующего подграфа типа треугольника

$$\Delta(0) = (V_{\Delta(0)}, E_{\Delta(0)}) \subset \Gamma,$$
$$V_{\Delta(0)} = \{L(w_L - 1), R(w_R - 1), C(w - 1)\},$$
$$E_{\Delta(0)} = \{(L(w_L - 1), C(w - 1)), (R(w_R - 1), C(w - 1))\},$$

будет называться *ProcessTriangle* (см. алгоритм 14). На этом конкретном шаге алгоритма, в ходе исполнения *ProcessTriangle* применительно к $\Delta(0)$, результат вычислений сохраняется в правой вершине $R(w_R - 1)$.

После этого, мысленно удалив ребра, входящие в вершину $C(w - 1)$, выходная степень вершины $L(w_L - 1)$ станет равной либо 1, либо 2. Если она станет равной 1, выполняется процедура *ProcessTriangle* для вершин треугольника

$$\Delta(1) = (V_{\Delta(1)}, E_{\Delta(1)}) \subset \Gamma,$$
$$V_{\Delta(1)} = \{L(w_L - 1), R(w_R - 2), C(w - 2)\},$$
$$E_{\Delta(1)} = \{(L(w_L - 1), C(w - 2)), (R(w_R - 2), C(w - 2))\},$$

и результат вычисления $C(w-2)$ записывается в левую вершину $L(w_L-1)$ (см. шаг 2 на рис. 5). После этого действия степень исхода вершины $R(w_R-2)$ станет равной 1, и мы можем повторить шаги, описанные выше, для правой вершины $R(w_R - 2)$ (см. шаг 3 на рис. 5).

Алгоритм 14 Алгоритм *ProcessTriangle*

Input: Изображения J_L и J_R, их ширины w_L и w_R, их одинаковая высота h, массивы индексов K_L и K_R, которые позволяют восстановить правильный порядок столбцов изображений J_L и J_R, индекс n треугольника $\Delta(n)$, подлежащего обработке, и массив K, который позволяет восстановить правильный порядок столбцов изображения J

$t \leftarrow w - 1 - n$

$t_L \leftarrow w_L - 1 - \lfloor n/2 \rfloor$

$t_R \leftarrow w_R - 1 - \lfloor (n+1)/2 \rfloor$

$s \leftarrow (t - t_R) \bmod h$

$k_L \leftarrow K_L(t_L)$

$k_R \leftarrow K_R(t_R)$

if $n = 0 \bmod 2$ **then** ▷ результат сохраняется в правую вершину

    $J_R(k_R, :) \leftarrow J_L(k_L, :) + Concat(J_R(k_R, s:h), J_R(k_R, 0:s))$

    $K(t) \leftarrow w_R + k_R$

else ▷ результат сохраняется в левую вершину

    $J_L(k_L, :) \leftarrow J_L(k_L, :) + Concat(J_R(k_R, s:h), J_R(k_R, 0:s))$

    $K(t) \leftarrow k_L$

Таким образом, снизу вверх по вычислительному графу обрабатывая последовательные треугольники, при этом чередуя процесс записи merge-результатов в левые и правые вершины (шаги 1–3), мы в конечном итоге дойдем до вершины $L([w/4]-1)$, которая будет иметь степень исхода, равную 2 (хотя изначально ее степень исхода была равна 3). На этом этапе степень исхода всех оставшихся (не удаленных из рассмотрения) левых и правых вершин будет равна 2, и следовательно, оставшийся граф раскладывается на бабочки (шаг 4 на рис. 5, также см. рис. 3,*б*)). В дальнейшем остается последовательно обработать полученные бабочки с помощью повторных применений процедуры *ProcessButterfly*, как это описано выше для изображений с четной шириной (см. алгоритм 13). Схематично алгоритм *FHT2IDS* проиллюстрирован на рис. 5.

Следуя алгоритму *FHT2IDS*, можно уточнить, что для нечетной ширины w граф Γ состоит из $[w/4]$ последовательных бабочек $\{\mathcal{B}(k)\}_{k\in\mathbb{Z}_{[w/4]}}$ и $w - 2[w/4]$ треугольников $\{\Delta(k)\}_{k\in\mathbb{Z}_{w-2[w/4]}}$. Треугольник $\Delta(k)$ с номером k задается следующим образом:

$$\Delta(k) = (V_{\Delta(k)}, E_{\Delta(k)}) \subset \Gamma, \quad k \in \mathbb{Z}_{w-2[w/4]},$$
$$V_{\Delta(k)} = \{L(w_L - 1 - \lfloor k/2 \rfloor), R(w_R - 1 - \lfloor (k+1)/2 \rfloor), C(w-1-k)\},$$
$$E_{\Delta(k)} = \{(v, C(w-1-k)) \mid v = L(w_L - 1 - \lfloor k/2 \rfloor) \vee$$
$$\vee\, v = R(w_R - 1 - \lfloor (k+1)/2 \rfloor)\}.$$

Важно отметить, что стадия обработки любого треугольника не требует выделения дополнительной памяти и выполняется исключительно в пределах существующей памяти. В то же время, для бабочек, как было отмечено при рассмотрении случая с четной шириной w, необходимо выделить только один массив размера h. Это приводит к тому, что $M_{IDS}(w,h) = h$.

Используя графовую интерпретацию из § 4, в терминах пары отображений (K, π_α) при $\alpha = 2$ мы предлагаем следующие отображения для случая нечетной ширины w:

$$K(C(w-1-k)) = R(w_R - 1 - \lfloor (k+1)/2 \rfloor), \quad k = 0 \bmod 2, \quad k \in \mathbb{Z}_{w-2[w/4]}, \quad (24)$$
$$K(C(w-1-k)) = L(w_L - 1 - \lfloor k/2 \rfloor), \quad k = 1 \bmod 2, \quad k \in \mathbb{Z}_{w-2[w/4]}, \quad (25)$$
$$K(C(2k)) = L(k), K(C(2k+1)) = R(k), \quad k \in \mathbb{Z}_{[w/4]}, \quad (26)$$

и

$$\pi_2(k) = \{w-1-k\}, \quad k \in \mathbb{Z}_{w-2[w/4]},$$
$$\pi_2(k+w-2[w/4]) = \{2k, 2k+1\}, \quad k \in \mathbb{Z}_{[w/4]},$$
$$\pi_2(k) = \varnothing, \quad k < w \wedge k \geqslant w - [w/4].$$

Для нечетного w структура отображения K наглядно проиллюстрирована на рис. 4,*б*). Образ целого числа $n \in \mathbb{Z}_w$ при отображении π_2 есть множество индексов t вершин $C(t)$, помеченных на рис. 4 числом n.

Согласно представленным рассуждениям, основанным на теоремах из § 5, пара (K, π_2) обеспечивает in-place свойство (ранга 2) для вычислительного графа Γ и в конечном итоге на теоретическом уровне доказывает, что на каждом шаге рекурсии алгоритм *FHT2IDS* требует аллоцирования дополнительного массива памяти размера не более $w + h$. Действительно, каждый рекурсивный вызов (без обработки дочерних вызовов) требует аллоцирования массива размера не более $w' + h$, точнее, массива размера $w' \leqslant w$ для хранения массива K (строка 10 алгоритма 15) и массива размера h для обработки всех бабочек (строка 9 алгоритма 13). Кроме того, доказано, что результаты работы алгоритмов *FHT2IDS* и *FHT2DS* совпадают, с точностью до известной перестановки K столбцов Хаф-образа, и вычислительная сложность (т.е. количество выполненных суммирований) обоих алгоритмов одинакова.

Обобщая вышеизложенное, можно сказать, что предложенный нами алгоритм *FHT2IDS* состоит из двух основных этапов: редукции вычислительного графа до вида объединения бабочек путем удаления треугольников (этот этап отсутствует для случая четного w), а затем последовательной обработки бабочек. Общий псевдокод алгоритма, который использует процедуры *ProcessButterfly* и *ProcessTriangle*, представлен в виде алгоритма 15.

Алгоритм 15 Алгоритм *FHT2IDS*

1: **Input:** изображение $I\colon \mathbb{Z}_w \times \mathbb{Z}_h \to \mathbb{A}$, его ширина w и высота h
2: **Output:** изображение $I\colon \mathbb{Z}_w \times \mathbb{Z}_h \to \mathbb{A}$, массив $K\colon \mathbb{Z}_w \to \mathbb{Z}_w$ индексов столбцов изображения J для их естественного переупорядочения (согласованного с результатом выполнения алгоритма *FHT2DS*)
3: **if** $w > 1$ **then**
4: $w_L \leftarrow \lfloor w/2 \rfloor$
5: $w_R \leftarrow w - w_R$
6: $I_L \leftarrow I(0 : w_L, :)$ ▷ выделение памяти не производится
7: $I_R \leftarrow I(w_L : w, :)$ ▷ выделение памяти не производится
8: $\langle I_L, K_L \rangle \leftarrow FHT2IDS(I_L, w_L, h)$
9: $\langle I_R, K_R \rangle \leftarrow FHT2IDS(I_R, w_R, h)$
10: $K \leftarrow CreateZeroedArray(w)$ ▷ выделяется память для массива размера w
11: **if** $w = 0 \bmod 2$ **then**
12: **for** $n \leftarrow 0$ **to** $w/2 - 1$ **do** ▷ последовательная обработка бабочек
13: $ProcessButterfly(I_L, I_R, w_L, w_R, h, K_L, K_R, n, K)$
14: **else** ▷ последовательная обработка треугольников и затем – бабочек
15: **for** $n \leftarrow 0$ **to** $w - 2[w/4] - 1$ **do**
16: $ProcessTriangle(I_L, I_R, w_L, w_R, h, K_L, K_R, n, K)$
17: **for** $n \leftarrow 0$ **to** $[w/4] - 1$ **do**
18: $ProcessButterfly(I_L, I_R, w_L, w_R, h, K_L, K_R, n, K)$
19: **else**
20: $K(0) \leftarrow 0$
return I, K

§ 7. Экспериментальная оценка предложенного алгоритма *FHT2IDS*

Чтобы сравнить время работы алгоритмов *FHT2DS* и *FHT2IDS*, мы реализовали их на языке C++ и протестировали на квадратных изображениях с линейным размером от 1 до 4100, заполненных случайными числами от 0 до 255. Тип входного изображения был 32-битным, с плавающей запятой или целым беззнаковым. Эксперименты проводились на процессоре Intel Core i9-9900KF с архитектурой x86_64. В реализации алгоритма *FHT2IDS* задействовалась операция векторного суммирования. Мы использовали интринсики семейства SSE. Операции выполнялись на 128-битных регистрах, которые могут хранить либо 4 числа с плавающей запятой одинарной точности, либо 4 беззнаковых целых.

Результаты представлены на рис. 6 и 7. Видно, что in-place версия алгоритма *FHT2DS* оказалась быстрее остальных. В среднем, во всем рассматриваемом диапазоне ширин, выигрыш в скорости от использования in-place версии составил около 26%. При этом чем больше линейный размер изображения, тем существеннее эффект от использования алгоритма *FHT2IDS*. В табл. 1 приведено среднее время работы для изображения размера 2048. Видно, что для этого размера ускорение составило около 28% для каждого типа данных входного изображения. Использование целочисленного типа незначительно повлияло на время работы: наиболее частая арифметическая операция в этих алгоритмах, сложение, выполняется практически одинаково быстро как на целых числах, так и на числах с плавающей запятой.

§ 8. Обсуждение

Предложенный алгоритм *FHT2IDS*, хотя и является in-place алгоритмом, не является in-order алгоритмом, что означает, что Хаф-образ, возвращаемый этим алгоритмом, в общем случае отличается от Хаф-образа, возвращаемого алгоритмом

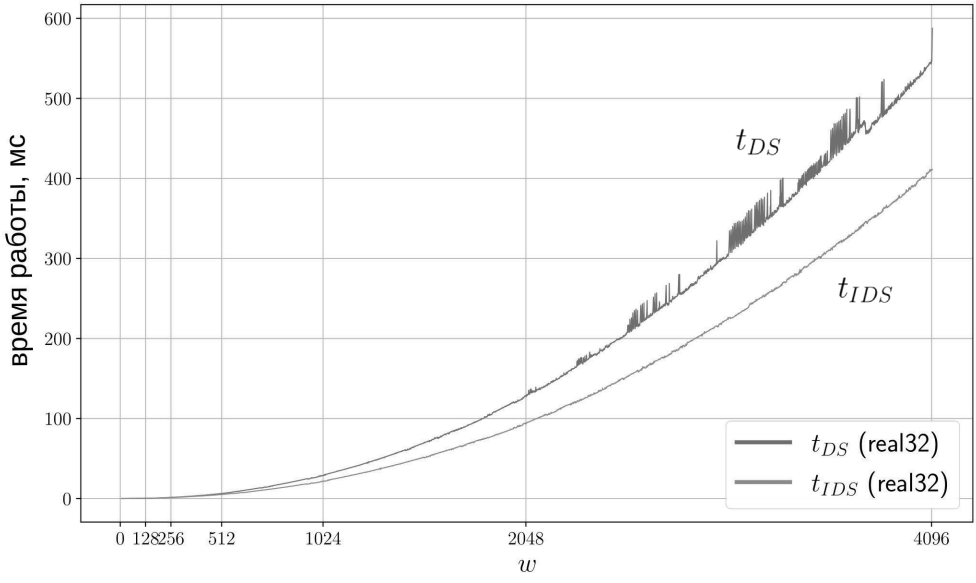


Рис. 6. График времени работы алгоритмов *FHT2DS* и *FHT2IDS* на изображениях 32-битного типа данных с плавающей запятой с линейным размером от 1 до 4100

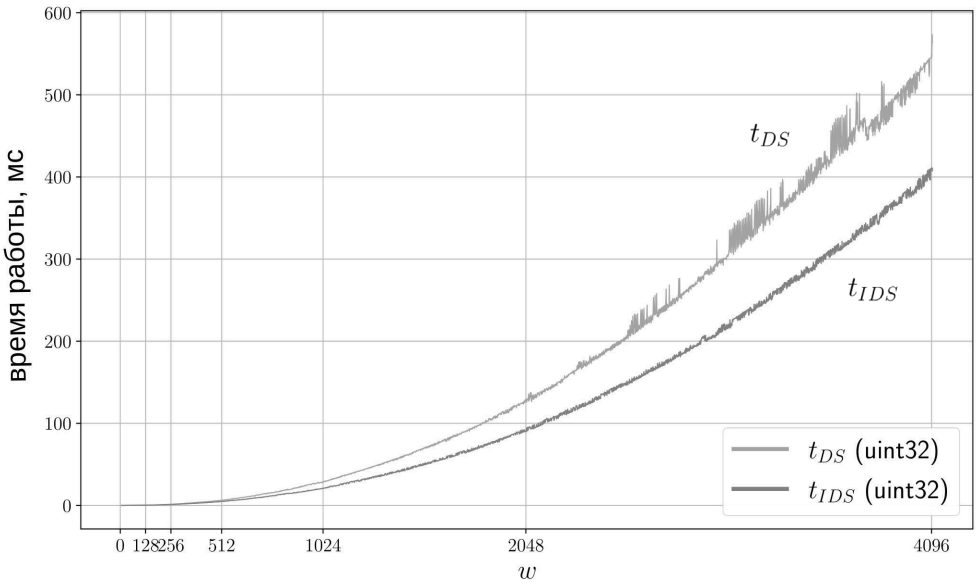


Рис. 7. График времени работы алгоритмов $FHT2DS$ и $FHT2IDS$ на изображениях 32-битного беззнакового целочисленного типа данных с линейным размером от 1 до 4100

Таблица 1

Среднее время работы алгоритмов $FHT2DS$ и $FHT2IDS$ на изображении размера 2048

	$\bar{t}_{DS}$	$\bar{t}_{IDS}$
real32	129 мс	93 мс
uint32	129 мс	93 мс

$FHT2DS$, перестановкой столбцов согласно отображению K. Для in-place алгоритма Кули – Тьюки для вычисления одномерного БПФ с основанием 2 и прореживанием по времени, как показано в работе [37], результирующий массив частот записывается в порядке битовой обратной записи, когда размер входного массива является степенью двойки. Поскольку, как было показано, вычислительные графы алгоритма $FHT2DS$ и алгоритма Кули – Тьюки для БПФ с основанием 2 и прореживанием по времени изоморфны для изображений четной ширины, можно заключить, что для ширин, являющихся степенями двойки, функция K также определяет порядок битовой обратной записи индексов столбцов Хаф-образа. Для произвольных ширин w порядок, определяемый функцией K, вычисляется в соответствии с уравнениями (23) и (24). Переупорядочение столбцов Хаф-образа в алгоритме $FHT2IDS$ требует дополнительных вычислительных затрат, и с целью оптимизации ПХ актуальна разработка не только in-place, но и in-order алгоритмов для вычисления БПХ [40].

Другим важным направлением для улучшения алгоритмов вычисления БПХ является разработка in-place модификаций более точных алгоритмов. В частности, алгоритм $FHT2DT$, предложенный в работах [27, 28], несколько медленнее алгоритма $FHT2DS$, но представляет более точный метод вычисления преобразования Хафа. Вычислительный граф алгоритма $FHT2DT$ имеет более сложную структу-

ру, и его эффективный анализ требует расширения результатов, приведенных в § 5. В этом направлении представляется возможным разработать более быстрые алгоритмы ПХ, сохраняя при этом высокую точность. Однако это, вероятно, потребует разработки более тонкого теоретического фундамента.

§ 9. Заключение

В настоящей статье предложен in-place алгоритм *FHT2IDS* для вычисления БПХ для изображений произвольной ширины. Предложенный алгоритм предстает in-place модификацией ранее предложенного в литературе алгоритма *FHT2DS* [26]. Нами приведено теоретическое обоснование корректности алгоритма *FHT2IDS*, показано, что его вывод совпадает с результатом работы алгоритма *FHT2DS* для изображений произвольной ширины. Это обоснование основано на анализе структуры вычислительного графа исходного алгоритма. Используемые методы доказательства могут быть полезны читателю при выводе in-place модификаций других алгоритмов, аналогичных тем, что применяются для вычисления БПХ или БПФ.

Продемонстрировано, что на каждом шаге рекурсии, при условии, что входное изображение имеет размерность $w \times h$, предложенный алгоритм *FHT2IDS* требует выделения массива размера не более $w + h$, что значительно меньше массива размера wh, который выделяется на каждом шаге рекурсии в рамках алгоритма *FHT2DS*. Снижение сложности алгоритма по объему вспомогательной памяти, в связи с оптимизацией управления кэшированием, а также уменьшением накладных расходов на копирование данных, приводят к значительному увеличению скорости при применении алгоритма *FHT2IDS* к большим изображениям по сравнению с алгоритмом *FHT2DS*. Экспериментальные результаты показывают, что алгоритм *FHT2IDS*, реализованный на языке C/C++, превосходит его out-of-place аналог *FHT2DS* на 26% по скорости. Алгоритм *FHT2IDS* реализован на языке Python и в настоящее время является частью общедоступной библиотеки *adrt* [41], которую читатель может найти полезной для собственных исследований. Мы рекомендуем читателю использовать разработанный алгоритм *FHT2IDS*, так как он существенно быстрее и значительно более экономен в смысле оперативной памяти по сравнению с алгоритмом *FHT2DS*.

СПИСОК ЛИТЕРАТУРЫ

1. *Hough P.V.C.* Machine Analysis of Bubble Chamber Pictures // Proc. 2nd Int. Conf. on High-Energy Accelerators and Instrumentation (HEACC 1959). CERN, Geneva, Switzerland. Sept. 14–19, 1959. P. 554–558.
2. *Rahmdel P.S., Comley R., Shi D., McElduff S.* A Review of Hough Transform and Line Segment Detection Approaches // Proc. 10th Int. Conf. on Computer Vision Theory and Applications (VISAPP 2015). Berlin, Germany. Mar. 11–14, 2015. V. 2. P. 411–418. https://doi.org/10.5220/0005268904110418
3. *Mukhopadhyay P., Chaudhuri B.B.* A Survey of Hough Transform // Pattern Recognit. 2015. V. 48. № 3. P. 993–1010. https://doi.org/10.1016/j.patcog.2014.08.027
4. *Illingworth J., Kittler J.* A Survey of the Hough Transform // Comput. Vision Graph. Image Process. 1988. V. 44. № 1. P. 87–116. https://doi.org/10.1016/S0734-189X(88)80033-1
5. *Xu Z., Shin B., Klette R.* Accurate and Robust Line Segment Extraction Using Minimum Entropy with Hough Transform // IEEE Trans. Image Process. 2014. V. 24. № 3. P. 813–822. https://doi.org/10.1109/TIP.2014.2387020
6. *Алиев М.А., Николаев Д.П., Сараев А.А.* Построение быстрых вычислительных схем настройки алгоритма бинаризации Ниблэка // Тр. ИСА РАН. 2014. Т. 64. № 3. С. 25–34.
7. *Nikolaev D.P., Nikolayev P.P.* Linear Color Segmentation and Its Implementation // Comput. Vis. Image Und. 2004. V. 94. № 1–3. P. 115–139. https://doi.org/10.1016/j.cviu.2003.10.012

8. *Saha S., Basu S., Nasipuri M., Basu D.* A Hough Transform Based Technique for Text Segmentation // J. Comput. 2010. V. 2. № 2. P. 134–141.
9. *Кунина И.А., Гладилин С.А., Николаев Д.П.* Слепая компенсация радиальной дисторсии на одиночном изображении с использованием быстрого преобразования Хафа // Компьютерная оптика. 2016. Т. 40. № 3. С. 395–403. https://doi.org/10.18287/2412-6179-2016-40-3-395-403
10. *Асватов Е.Н., Ершов Е.И., Николаев Д.П.* Робастная ортогональная линейная регрессия для маломерных гистограмм // Сенсорные системы. 2017. Т. 31. № 4. С. 331–342.
11. *Brady M.L., Yong W.* Fast Parallel Discrete Approximation Algorithms for the Radon Transform // Proc. 4th Ann. ACM Symp. on Parallel Algorithms and Architectures (SPAA'92). San Diego, California, USA. June 29 – July 1, 1992. P. 91–99. https://doi.org/10.1145/140901.140911
12. *Карпенко С.М., Ершов Е.И.* Исследование свойств диадического паттерна быстрого преобразования Хафа // Пробл. передачи информ. 2021. Т. 57. № 3. С. 102–111. https://doi.org/10.31857/S0555292321030074
13. *Jahan R, Suman P., Singh D.K.* Lane Detection Using Canny Edge Detection and Hough Transform on Raspberry Pi // Int. J. Adv. Res. Comput. Sci. 2018. V. 9. № 2. P. 85–89.
14. *Thongpan N., Rattanasiriwongwut M., Ketcham M.* Lane Detection Using Embedded System // Int. J. Comput. Internet Manag. 2020. V. 28. № 2. P. 46–51.
15. *Panfilova E., Shipitko O.S., Kunina I.* Fast Hough Transform-Based Road Markings Detection For Autonomous Vehicle // 13th Int. Conf. on Machine Vision (ICMV 2020). Rome, Italy. Nov. 2–6, 2020. Proc. SPIE. V. 11605. P. 671–680. https://doi.org/10.1117/12.2587615
16. *Котов А.А., Коноваленко И.А., Николаев Д.П.* Прослеживание объектов в видеопотоке, оптимизированное с помощью быстрого преобразования Хафа // ИтиВС. 2015. № 1. С. 56–68.
17. *Tropin D.V., Ilyuhin S.A., Nikolaev D.P., Arlazarov V.V.* Approach for Document Detection by Contours and Contrasts // Proc. 25th Int. Conf. on Pattern Recognition (ICPR 2020). Milan, Italy. Jan. 10–15, 2021. P. 9689–9695. https://doi.org/10.1109/ICPR48806.2021.9413271
18. *Bezmaternykh P.V., Nikolaev D.P.* A Document Skew Detection Method Using Fast Hough Transform // 12th Int. Conf. on Machine Vision (ICMV 2019). Amsterdam, Netherlands. Nov. 16–18, 2020. Proc. SPIE. V. 11433. P. 132–137. https://doi.org/10.1117/12.2559069
19. *Min-Allah N., Qureshi M.B., Alrashed S., Rana O.F.* Cost Efficient Resource Allocation for Real-Time Tasks in Embedded Systems // Sustainable Cities And Society. 2019 V. 48. Article No. 101523. https://doi.org/10.1016/j.scs.2019.101523
20. *Gupta R.K., De Micheli G.* Specification and Analysis of Timing Constraints for Embedded Systems // IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 1997. V. 16. № 3. P. 240–256. https://doi.org/10.1109/43.594830
21. *Surianarayanan C., Lawrence, J.J., Chelliah P.R., Prakash E., Hewage C.* A Survey on Optimization Techniques for Edge Artificial Intelligence (AI) // Sensors. 2023. V. 23. № 3. Papaer No. 1279 (33 pp.). https://doi.org/10.3390/s23031279
22. *Ranjith M.S., Parameshwara S., Pavan Yadav A., Hegde S.* Optimizing Neural Network for Computer Vision Task in Edge Device, https://arxiv.org/abs/2110.00791 [cs.CV], 2021.
23. *Sharmila B.S., Santhosh H.S., Parameshwara S., Swamy M.S., Baig W.H., Nanditha S.V.* Optimizing Deep Learning Networks for Edge Devices with an Instance of Skin Cancer and Corn Leaf Disease Dataset // SN Comput. Sci. 2023. V. 4. Article No. 793. https://doi.org/10.1007/s42979-023-02239-5
24. *Comeagă A.-M., Marin I.* Memory Management Strategies for an Internet of Things System, https://arxiv.org/abs/2311.10458 [cs.SE], 2023.
25. *Almutairi R., Bergami G., Morgan G.* Advancements and Challenges in IoT Simulators: A Comprehensive Review // Sensors. 2024. V. 24. № 5. Paper No. 1511 (35 pp.), https://doi.org/10.3390/s24051511

26. *Anikeev F.A., Raiko G.O., Limonova E.E., Aliev M.A., Nikolaev D.P.* Efficient Implementation of Fast Hough Transform Using CPCA Coprocessor // Program. Comput. Soft. 2021. V. 47. № 5. P. 335–343. https://doi.org/10.1134/S0361768821050029
27. *Kazimirov D., Nikolaev D., Rybakova E., Terekhin A.* Generalization of Brady–Yong Algorithm for Fast Hough Transform to Arbitrary Image Size, https://arxiv.org/abs/2411.07351 [cs.CV], 2024.
28. *Kazimirov D., Nikolaev D., Rybakova E., Terekhin A.* Generalization of Brady–Yong Algorithm for Fast Hough Transform to Arbitrary Image Size // Proc. 5th Symp. on Pattern Recognition and Applications (SPRA 2024). Istanbul, Turkey. Nov. 11–13, 2024 (to appear).
29. *Gava M.A., Rocha H.R.O., Faber M.J., Segatto M.E.V., Wörtche H., Silva J.A.L.* Optimizing Resources and Increasing the Coverage of Internet-of-Things (IoT) Networks: An Approach Based on LoRaWAN // Sensors. 2023. V. 23. № 3. Paper No. 1239 (17 pp.). https://doi.org/10.3390/s23031239
30. *Almurshed O., Meshoul S., Muftah A., Kaushal A.K., Almoghamis O., Petri I., Auluck N., Rana O.* A Framework for Performance Optimization of Internet of Things Applications // Euro-Par 2023: Parallel Processing Workshops (Euro-Par 2023 Int. Workshops. Limassol, Cyprus. Aug. 28 – Sept. 1, 2023. Revised Selected Papers, Part I). Lect. Notes Comput. Sci. V. 14351. Cham: Springer, 2024. P. 165–176. https://doi.org/10.1007/978-3-031-50684-0_13
31. *Chakraborty S., Mukherjee A., Raman V., Satti S.R.* A Framework for In-place Graph Algorithms // 26th Annu. Europ. Symp. on Algorithms (ESA 2018). Helsinki, Finland. Aug. 20–22, 2018. Leibniz Int. Proc. Inform. (LIPIcs). V. 112. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Germany: Dagstuhl Publ., 2018. P. 13:1–13:16. https://doi.org/10.4230/LIPIcs.ESA.2018.13
32. *Axtmann M., Witt S., Ferizovic D., Sanders P.* Engineering In-place (Shared-Memory) Sorting Algorithms // ACM Trans. Parallel Comput. 2022. V. 9. № 1. P. 1–62. https://doi.org/10.1145/3505286
33. *Gu Y., Obeya O., Shun J.* Parallel In-place Algorithms: Theory and Practice // 2nd Symp. on Algorithmic Principles of Computer Systems (APOCS 2020). Virtual Conf. Jan. 13, 2021. P. 114–128. https://doi.org/10.1137/1.9781611976489.9
34. *Brönnimann H., Chan T.M., Chen E.Y.* Towards In-place Geometric Algorithms and Data Structures // Proc. 12th Annu. Symp. on Computational Geometry (SCG'04). Brooklyn, New York, USA. June 8–11, 2004. P. 239–246. https://doi.org/10.1145/997817.997854
35. *Kuszmaul W., Westover A.* Cache-Efficient Parallel-Partition Algorithms Using Exclusive-Read-and-Write Memory // Proc. 32nd ACM Symp. on Parallelism in Algorithms and Architectures (SPAA'20). Virtual Event, USA. July 15–17, 2020. P. 551–553. https://doi.org/10.1145/3350755.3400234
36. *Bramas B., Bramas Q.* On the Improvement of the In-place Merge Algorithm Parallelization, https://arxiv.org/abs/2005.12648 [cs.DC], 2020.
37. *Cooley J.W., Tukey J.W.* An Algorithm for the Machine Calculation of Complex Fourier Series // Math. Comp. 1965. V. 19. № 90. P. 297–301. https://doi.org/10.2307/2003354
38. *Brady M.L., Yong W.* Fast Parallel Discrete Approximation Algorithms for the Radon Transform // Proc. 4th Ann. ACM Symp. on Parallel Algorithms and Architectures (SPAA'92). San Diego, California, USA. June 29 – July 1, 1992. P. 91–99. https://doi.org/10.1145/140901.140911
39. *Khanipov T.* Computational Complexity Lower Bounds of Certain Discrete Radon Transform Approximations, https://arxiv.org/abs/1801.01054 [cs.CC], 2018.
40. *Johnson H., Burrus C.* An In-order, In-place Radix-2 FFT // Proc. ICASSP'84: IEEE Int. Conf. on Acoustics, Speech, and Signal Processing. San Diego, CA, USA. Mar. 19–21, 1984. P. 473–476. https://doi.org/10.1109/ICASSP.1984.1172660
41. IITP Vision Lab. adrt: Approximate Discrete Radon Transform. GitHub repository, accessed 21.10.2024. https://github.com/iitpvisionlab/adrt

Казимиров Данил Дмитриевич
Институт проблем передачи информации
им. А.А. Харкевича РАН, Москва
Московский государственный университет
им. М.В. Ломоносова, Москва
ООО “Смарт Энджинс Сервис”, Москва
d.kazimirov@smartengines.com
Николаев Дмитрий Петрович
ООО “Смарт Энджинс Сервис”, Москва
Федеральный исследовательский центр
“Информатика и управление” РАН, Москва
d.p.nikolaev@smartengines.com
Рыбакова Екатерина Олеговна
Московский государственный университет
им. М.В. Ломоносова, Москва
ООО “Смарт Энджинс Сервис”, Москва
e.rybakova@smartengines.com
Терехин Арсений Павлович
Институт проблем передачи информации
им. А.А. Харкевича РАН, Москва
ars@iitp.ru

Поступила в редакцию
07.12.2024
После доработки
18.12.2024
Принята к публикации
25.12.2024

ПРОБЛЕМЫ ПЕРЕДАЧИ ИНФОРМАЦИИ

Том 60 **2024** **Вып. 4**

УДК 519.21 : 519.72 : 512.75 : 519.1

© 2024 г. **М.Л. Бланк**

О ЗАСЕДАНИЯХ ДОБРУШИНСКОГО СЕМИНАРА В 2024 Г. (ЧАСТЬ 2)

Добрушинский семинар посвящен основным направлениям фундаментальной математики, которые развиваются в Добрушинской математической лаборатории: стохастической и детерминированной динамике больших систем, теории информации и теории кодирования, алгебраической геометрии и теории чисел, комбинаторным и вероятностным аспектам теории представлений. Представлена общая информация о семинаре, а также подробная информация о заседаниях семинара, прошедших с сентября 2024 г.

Ключевые слова: Добрушинский семинар, математика, динамика больших систем, эргодическая теория, теория информации, теория кодирования, алгебраическая геометрия, теория чисел.

DOI: 10.31857/S0555292324040077, **EDN:** CHEEXS

Общие сведения о семинаре

Семинар ранее проходил в ИППИ РАН, а начиная с сентября проходит в рамках Высшей школы современной математики (ВШМ) МФТИ, по вторникам с 16:15 до 18:00, МФТИ, радиотехнический корпус, РТ113.

Руководитель семинара и заведующий Добрушинской лабораторией профессор Михаил Львович Бланк.

Семинар посвящен основным направлениям, которые развиваются в Добрушинской математической лаборатории:

- стохастическая и детерминированная динамика больших систем;
- теория информации и кодирования;
- алгебраическая геометрия и теория чисел;
- комбинаторные и вероятностные аспекты теории представлений.

На семинаре с докладами выступают как сотрудники лаборатории и ВШМ, так и приглашенные докладчики. Приглашаются все желающие принять участие в обсуждении.

Семинар открыт для достаточно широкого круга математических вопросов в соответствии с научными интересами участников семинара.

Желающие выступить на семинаре, пожалуйста, обращайтесь к М.Л. Бланку (mlblank@gmail.com).

Сайты семинара:

https://www.mathnet.ru/conf167

https://sites.google.com/view/dobr-seminar

Заседание 17 сентября 2024 г.

Тема семинара: Наследование свойства отслеживания в динамических полугруппах.

Докладчик: М.Л. Бланк, Московский физико-технический институт (государственный университет), г. Долгопрудный, Московская обл.; Высшая школа экономики, г. Москва.

Аннотация: Проблема отслеживания псевдо-траекторий (траекторий системы под действием слабых возмущений) связана с классическими результатами Д.В. Аносова о робастности равномерно гиперболических систем. Основная трудность здесь состоит в анализе бесконечно многих (по времени) возмущений системы. Я расскажу о новом подходе к концепции отслеживания, позволяющем преодолеть эту трудность и драматически расширить применимость теории отслеживания, в частности, к динамическим полугруппам и более общим типам возмущений (например, гауссовым). Все необходимые математические сведения будут обсуждены по ходу дела, при этом мы постараемся пройти весь путь от элементарных постановок до еще не решенных задач.

Заседание 24 сентября 2024 г.

Тема семинара: Какая часть корней системы случайных полиномов вещественна?

Докладчик: Б.Я. Казарновский, Московский физико-технический институт (государственный университет), г. Долгопрудный, Московская обл.

Аннотация: Какова вероятность вещественности корня многочлена степени n с вещественными коэффициентами? Ответ М. Каца (1942): она асимптотически равна $2/\pi \log(n)/n$. Многочлен Лорана, вещественный на единичной окружности, назовем вещественным, как и его корни на этой окружности. Оказывается, что вероятность вещественности корня многочлена Лорана растущей степени стремится не к нулю, а к $1/\sqrt{(}3)$. Т.е. предел $> 1/2$! Верно, что феномен асимптотической конечности доли вещественных корней сохраняется для систем многочленов Лорана многих переменных, а также для многочленов на произвольной компактной группе Ли. В частности, корни многочленов на группе матриц сваливаются на унитарную подгруппу. В случае многочленов Лорана от n переменных, соответствующая компактная группа есть n-мерный тор. Асимптотика доли вещественных корней вычисляется через смешанные объемы некоторых выпуклых компактных множеств, определяющих рост системы полиномов. Формулировки теорем элементарны и будут приведены полностью. Будет также приведено "объяснение" доказательств. Они основаны на применении двух результатов о числе корней систем уравнений. Это версии теоремы БКК (Бернштейна – Кушниренко – Хованского) о числе корней полиномиальной системы уравнений для соответственно гладких функций и для полиномов на комплексной линейной группе.

Заседание 1 октября 2024 г.

Тема семинара: Обобщенный спектральный радиус – откуда взялся и зачем нужен.

Докладчик: В.С. Козякин, Московский физико-технический институт (государственный университет), г. Долгопрудный, Московская обл.

Аннотация: Одним из наиболее распространенных способов решения векторного линейного уравнения $x = Ax + f$ является метод простых итераций, суть которого в нахождении решения путем последовательных приближений $x(n+1) = Ax(n) + f$. Распространен также другой способ решения исходного уравнения – так называемый метод Гаусса – Зейделя, который может быть представлен как поочередное вычисление итераций либо по формуле $x(n+1) = Bx(n) + f$, либо по формуле $x(n+1) = Cx(n) + f$, где B и C – специальным образом сконструированные по A более простые матрицы "построчного пересчета". Возникает вопрос, что случится, если в методе Гаусса – Зейделя матрицы B и C начнут применяться не поочередно,

а в произвольном порядке? Этот вопрос далеко не искусственный, и имеется множество примеров "реальных" задач, приводящих к нему, – распараллеливание вычислений, системы управления с асинхронным обменом данными, поведение систем коллективного поведения, проблема гладкости вейвлетов, теория арбитражных операций валютного рынка и т.д. Переход от итерационных процедур с одной матрицей к процедурам с несколькими матрицами, применяемыми в произвольном порядке, мгновенно делает практически бесполезной всю технику классической линейной алгебры и переводит задачу из алгоритмически и вычислительно простой в разряд "супертрудных", основные методы решения которой в настоящее время ассоциируются с так называемой теорией обобщенного или совместного спектрального радиуса наборов матриц. В докладе будут представлены начальные постановки и определения соответствующей теории, будет дано объяснение алгебраической неразрешимости и NP-сложности вычисления обобщенного спектрального радиуса, будет сформулирована "неконструктивная" теорема о так называемых "нормах Барабанова", до сих пор вопреки своей неконструктивности являющаяся одним из немногих работающих инструментов данной теории.

Заседание 8 октября 2024 г.

Тема семинара: Покрытие полосками и уклонение от множества нулей многочлена

Докладчик: Р.Н. Карасёв, Институт проблем передачи информации им. А.А. Харкевича Российской академии наук, г. Москва; Московский физико-технический институт (государственный университет), г. Долгопрудный, Московская обл.

Аннотация: Со времен Альфреда Тарского известна задача о том, какова должна быть минимальная суммарная ширина набора полосок, которые покрывают круг (или шар, произвольное выпуклое тело и пр.). Достаточно общий случай этой задачи решил Тогер Банг с помощью оптимизации некоторого квадратичного функционала на булевом кубе. Этот метод оказался достаточно плодотворным и был распространен, например, Китом Боллом на покрытие полосками единичного шара банахова пространства, а также Александром Полянским и Цзылинем Цзяном на задачу Ласло Фейеш Тота о покрытии сферы зонами. Но сравнительно недавно Оскар Ортега-Морено и Юйфей Чжао придумали в каком-то смысле более прямой метод решения задач про полоски через свойства максимума многочленов. По сути все сводится к доказательству того, что максимум модуля вещественного или комплексного многочлена на сфере находится достаточно далеко от множества нулей этого многочлена. Какие-то оценки в этом вопросе следуют из классического неравенства Бернштейна для нормы производной, но для точной оценки рассуждение надо модифицировать. Также надо немного поработать, чтобы применить этот метод не к сфере, а к шару. Развивая полиномиальный метод, удается доказать гипотезу Полянского – Цзяна, обобщающую их теорему, и несколько усилить теорему Кита Болла о покрытии шара комплексными полосками, которая до этого доказывалась довольно запутанным образом. При этом возникает и не решенная до конца задача о вещественном полиномиальном аналоге покрытия сферы (или чего-то еще) полосками разной ширины. Работа совместная с Алексеем Глазыриным (ун-т Техаса в долине Рио-Гранде) и Александром Полянским (ун-т Эмори).

Заседание 15 октября 2024 г.

Тема семинара: Кластерное преобразование Дональдсона – Томаса и q-характеры модулей Кириллова – Решетихина.

Докладчик: Г.А. Кошевой, Институт проблем передачи информации им. А.А. Харкевича Российской академии наук, г. Москва.

Аннотация: Кластерные алгебры, введенные Сергеем Фоминым и Андреем Зелевинским около 2000 года, являются коммутативными алгебрами, генераторы и соотношения которых строятся рекурсивным образом. Среди этих алгебр есть алгебры однородных координат на грассманианах, на флаговых многообразиях и на многих других многообразиях, которые играют важную роль в геометрии и теории представлений. Основной целью Фомина и Зелевинского было создание комбинаторной структуры для изучения так называемых канонических базисов, которыми обладают эти алгебры и которые тесно связаны с понятием полной положительности в ассоциированных многообразиях. Быстро выяснилось, что комбинаторика кластерных алгебр также появляется во многих других предметах, например, в пуассоновой геометрии; дискретных динамических системах; высших пространств Тейхмюллера; комбинаторике и, в частности, изучение многогранников, таких как ассоциэдры Сташефа; некоммутативной алгебраической геометрия и, в частности, изучения условий стабильности в смысле Бриджленда, алгебр Калаби – Яу, инвариантов Дональдсона – Томаса и в теории представлений колчанов и конечномерных алгебр. Колчан – это ориентированный граф. Мутация колчана – это элементарная операция над колчанами и базовый комбинаторный ингредиент определения кластерных алгебр, которые рекурсивно строятся путем многократной мутации исходного семени (Q, x), состоящего из колчана Q и набора переменных x, связанных с вершинами колчана Q. Важное свойство мутаций – лорановское изменение переменных. Граф обмена имеет вершины, являющиеся семена, полученные из начального (Q, x) путем итерационной мутации, а ребра соответствуют мутациям. Максимальные зеленые последовательности введены Келлером при решении гипотезы периодичности системы Замолодчикова, хотя неявно имеются уже в работе Гайотто – Мура – Ницке. Максимальная зеленая последовательность – это специальный (конечный) путь в ориентированном графе обмена. Не все колчаны имеют максимальные зеленые последовательности. Существование таких последовательностей важно для положительного подтверждения гипотезы Фока – Гончарова. Гончаров и Шен назвали кластерным преобразованием Дональдсона – Томаса преобразование начальных переменных, соответствующие зеленым последовательностям. Для изучения этого преобразования важно не только доказать существование, но и знать явный вид максимальных зеленых последовательностей. Для класса колчанов для координатных колец однородных пространств. Колчаны для квантовых аффинных алгебр можно включить в этот класс. В нашей совместной работе с Канакубо и Накашима показано, что q-характеры модулей Кириллова – Решетихина квантовых аффинных алгебр можно вычислить, используя кластерные преобразования, следуя специфической максимальной зеленой последовательности. Это позволяет получать явные формулы кластерных преобразований Дональдсона – Томаса для координатных колец больших клеток Брюа. А используя алгоритм Френкеля – Мухина или наш алгоритм с Канакубо и Накашима получить явные формулы гораздо быстрее кластерного вычисления преобразования Дональдсона – Томаса. Доклад должен быть понятен неспециалистам.

Заседание 22 октября 2024 г.

Тема семинара: Невихревые уравнения Максвелла в цилиндре, обратные волны, резонансное излучение и линейный операторный пучок Келдыша.

Докладчик: А.Л. Делицын, Московский физико-технический институт (государственный университет), г. Долгопрудный, Московская обл.

Аннотация: Если бросить камень в воду, то круги от него будут расходиться, а не сходиться. С математической точки зрения это утверждение формулируется как совпадение знаков фазовой и групповой скорости волны. В то же время, начиная с работ Лэмба, известно что в упругом цилиндре знаки фазовой и групповой

скоростей волны могут иметь противоположное направление. Подобные волны называются обратными. Аналогичное явление имеет место и в электродинамике. Возникает вопрос об исследовании соответствующих спектральных задач и правильной постановке условий излучения в цилиндре. Уравнения Максвелла содержат восемь уравнений для шести неизвестных функций – шесть вихревых и два для дивергенций полей. При постановке начально-краевой задачи для уравнений Максвелла традиционно в качестве уравнений выбирают вихревые уравнения. Оставшиеся два уравнения рассматривают как их следствия. Это приводит к неоправданно тяжелым, по сути не поддающимся исследованию, спектральным задачам при попытках рассмотрения задач в цилиндре. Задачи в цилиндре имеют широкий круг приложений, например, к ним относятся задачи волоконной оптики. Мы используем подход, основанный на ином выборе основных уравнений, который сразу сводит спектральную задачу к линейному пучку Келдыша и очень сильно упрощает ее исследование. Рассматриваются различные примеры и приложения – излучение обратных волн, картины дисперсионных кривых их особых точек, резонансное излучение с аномальной скоростью роста.

Заседание 29 октября 2024 г.

Тема семинара: Об уравнении Кортевега – де Фриза.

Докладчик: В.В. Соколов, Московский физико-технический институт (государственный университет), г. Долгопрудный, Московская обл.

Аннотация: С каждым интегрируемым уравнением связано много различных интересных математических объектов и структур. Самым знаменитым эволюционным интегрируемым уравнением является уравнение Кортевега – де Фриза. На его примере я постараюсь описать часть из них. А именно, речь пойдет о локальных законах сохранения, инфинитезимальных симметриях, рекурсивных операторах, локальных гамильтоновых структурах и представлении Лакса.

Заседание 12 ноября 2024 г.

Тема семинара: О распределении одной статистической суммы, связанной с двоичным симметричным каналом.

Докладчик: М.В. Бурнашев, Московский физико-технический институт (государственный университет), г. Долгопрудный, Московская обл.

Аннотация: Исследуется функция распределения суммы независимых, одинаково распределенных случайных величин специального вида. С помощью такой суммы описываются текущие апостериорные вероятности сообщений для случайно выбранного кода в двоичном симметричном канале. Получены близкие между собой неасимптотические оценки снизу и сверху для этой функции распределения.

Заседание 19 ноября 2024 г.

Тема семинара: Апериодические точки внешних бильярдов.

Докладчик: В.А. Тиморин, Национальный исследовательский университет "Высшая школа экономики", г. Москва.

Аннотация: Внешний бильярд вокруг выпуклой фигуры на плоскости – отображение, отправляющее каждую точку вне данной фигуры в другой конец отрезка, начинающегося в этой точке и касающегося данной фигуры посередине. Итерации внешнего бильярда были предложены Ю. Мозером в качестве грубой модели движения планет. Если фигура – многоугольник, то получаются нетривиальные примеры кусочно-евклидовых перекладываний многоугольных кусков, двумерные аналоги перекладываний отрезков. Мы рассмотрим внешние бильярды относительно

правильных N-угольников. Ранее известные строгие результаты в этом направлении опирались на динамическое самоподобие (такой подход был впервые применен С. Табачниковым), за исключением "тривиальных" (или "интегрируемых") случаев $N = 3, 4, 6$. Самоподобия обнаружены, на текущий момент, только в случаях $N = 5, 7, 8, 9, 10, 12$. В своем докладе на международном математическом конгрессе 2022 года Р. Шварц высказал гипотезу о том, что "внешний бильярд на правильном N-угольнике имеет апериодическую орбиту, если N не равно 3, 4, 6". Наша работа доказывает гипотезу Шварца методами, не имеющими отношения к самоподобию. Основные инструменты приходят из теории равносоставленности, в виде аддитивных инвариантов, обобщающих инвариант Са – Арну – Фати (инвариант перекладываний отрезков) на многомерный случай, с использованием инварианта трансляционной равносоставленности Хадвигера и Глура. Основано на совместных проектах с А. Белым, А. Канель-Беловым, Ф. Руховичем, В. Згурским.

Заседание 26 ноября 2024 г.

Тема семинара: Классификация интегрируемых эволюционных уравнений.

Докладчик: В. В. Соколов, Московский физико-технический институт (государственный университет), г. Долгопрудный, Московская обл.

Аннотация: Будет предъявлена бесконечная серия необходимых условий интегрируемости. В частности, на примере простейшей классификационной задачи, мы продемонстрируем, что нескольких первых условий достаточно, чтобы получить конечный список уравнений.

Заседание 3 декабря 2024 г.

Тема семинара: Эквивалентные формулировки гипотезы Римана и их анализ.

Докладчик: О.Р. Мусин, отделение математики, Техасский университет в Браунсвилле.

Аннотация: Гипотеза Римана (ГР) эквивалентна многим другим гипотезам о скорости роста некоторых арифметических функций. Типичным примером является теорема Робена о сумме делителей – функции $\sigma(n)$. В 1915 году Рамануджан доказал асимптотические неравенства для $\sigma(n)$, которые эквивалентны ГР. В этом докладе я расскажу об этой и других эквивалентных формулировках для колоссально избыточных (СА) чисел, которые впервые были изучены Рамануджаном, а позднее Эрдёшом с соавторами. Я также расскажу о работе нашей группы по численному анализу этих гипотез. В частности, подтверждается гипотеза, что константы Рамануджана точны. Мы также изучали экстремумы этих функций на множестве СА и обнаружили некоторые интересные закономерности. Часть из них можно доказать при условии, что ГР верна, а часть является открытыми проблемами.

Заседание 10 декабря 2024 г.

Тема семинара: Инвариантные меры непрерывной модели контактов и случайные блуждания.

Докладчик: Е.А. Жижина, Московский физико-технический институт (государственный университет), г. Долгопрудный, Московская обл.

Аннотация: В докладе я расскажу про стохастическую модель контактов в непрерывном пространстве. Будет рассмотрен так называемый критический режим, когда рождение и гибель находятся в равновесии. Обсудим, какие условия на интенсивности рождения и гибели гарантируют существование инвариантных мер. Оказывается, эти условия различны для малых ($d = 1, 2$) и больших размерностей про-

странства ($d > 2$). Все результаты, о которых пойдет речь в докладе, получены совместно с С. Пироговым, Ю. Кондратьевым и О. Кутовым.

Заседание 17 декабря 2024 г.

Тема семинара: Об h-принципе для отображений с заданными бордмановскими особенностями.

Докладчик: А.Д. Рябичев, Московский физико-технический институт (государственный университет), г. Долгопрудный, Московская обл.

Аннотация: Пусть даны гладкие многообразия M и N одинаковой размерности. Мы хотим классифицировать отображения $M \to N$ с заданными особенностями. А именно, пусть в каждой точке замкнутого подмножества $S \subset M$ задан росток бордмановской особенности отображения в $\mathbb{R}^n$, причем все такие ростки локально согласованы. Задача: существует ли отображение $M \to N$ с особенностями, локально L-эквивалентными заданным? Оказывается, проще не строить отображение, а искать его в заданом гомотопическом классе. Для этого используется так называемый h-принцип, известный по работам Смейла, Хирша, Громова, Элиашберга и др. По заданным в S росткам естественно строится векторное расслоение E над M, такое что отображение $F\colon M \to N$ гомотопно отображению с заданными особенностями, если и только если расслоения f^*TN и E изоморфны. В докладе я напомню бордмановскую классификацию особенностей, а также опишу построение расслоения E и расскажу идею доказательства основной теоремы.

Бланк Михаил Львович
Высшая школа современной математики МФТИ, Москва
Национальный исследовательский университет
“Высшая школа экономики”, Москва
mlblank@gmail.com

ПРОБЛЕМЫ ПЕРЕДАЧИ ИНФОРМАЦИИ

Том 60 **2024** **Вып. 4**

УДК 621.39

© 2024 г. **В.А. Логинов**

О ЗАСЕДАНИЯХ МОСКОВСКОГО ТЕЛЕКОММУНИКАЦИОННОГО СЕМИНАРА В 2024 Г. (ЧАСТЬ 2)

Московский телекоммуникационный семинар организован научными группами ИППИ РАН, МФТИ и НИУ ВШЭ. Он посвящен научным аспектам связи и дает исследователям возможность представить и обсудить новые идеи и инновационные подходы в области телекоммуникационных технологий и тесно связанных с ними областей. Представлена общая информация о семинаре, а также подробная информация о заседаниях семинара, прошедших в конце 2024 года. Информацию о ближайших семинарах и форму подачи заявки на выступление можно найти на сайте семинара `https://wnlab.ru/seminar/`.

Ключевые слова: Московский телекоммуникационный семинар, телекоммуникации.

DOI: `10.31857/S0555292324040089`, **EDN:** DFRKCW

Общие сведения о семинаре

Московский телекоммуникационный семинар организован научными группами Института проблем передачи информации им. А.А. Харкевича Российской академии наук, Московского физико-технического института и Высшей школы экономики. Он посвящен научным аспектам связи и дает исследователям возможность представить и обсудить новые идеи и инновационные подходы в области телекоммуникационных технологий и тесно связанных с ними областей.

На семинар приглашаются ведущие исследователи, желающие поделиться своими недавними результатами в области передовых технологий и систем связи, которые продолжают менять мир и предоставляют всем пользователям доступ к беспрецедентному спектру высокоскоростных, безотказных глобальных телекоммуникационных услуг. Подать заявку на выступление можно на сайте семинара `https://wnlab.ru/seminar/`. Также приглашаются студенты и молодые ученые, желающие расширить и углубить свои познания в области новейших технологий передачи информации и улучшить свои профессиональные связи.

Основным языком является английский. Избранные доклады, связанные с кандидатскими и докторскими диссертациями, обсуждаются на русском языке. По умолчанию выступления записываются и публикуются через три месяца после мероприятия.

Семинар проходит очно в ИППИ РАН. Принять участие можно также в режиме видеоконференции.

Председатель семинара: Евгений Михайлович Хоров, д.т.н.

Ученый секретарь семинара: Вячеслав Аркадьевич Логинов, к.т.н.

Заседание 18 октября 2024 года

Тема семинара: Методы машинного обучения и оптимизации для повышения эффективности многоантенных систем.

Докладчик: Евгений Бобров, Московский государственный университет им. М.В. Ломоносова, Российский научно-исследовательский институт Huawei.

Аннотация: В докладе будут представлены новые математические методы оптимизации качества приема сигналов в беспроводной системе связи поколения 5G, обеспечивающих ее оптимальную пропускную способность и максимальное качество сигнала. Будут рассмотрены теоремы и методы, упрощающую исходную постановку задачи многолучевого распространения сигнала, и позволяющие выписать вычислительно более простые и эффективные функции оптимизации. В докладе будут рассмотрены как аналитические методы построения матрицы формирования луча сигнала, так и итерационные квази-ньютоновские процедуры. Также будет рассмотрен новый метод адаптации канала связи на основе машинного обучения, который по оценке отношения сигнала к шуму способен предсказывать оптимальную схему модуляции и кодирования сигнала в условиях нестационарного окружения.

Информация о докладчике: Евгений Бобров – выпускник аспирантуры факультета ВМК Московского государственного университета. Область научных интересов: машинное обучение, оптимизация и современные проблемы радиосвязи.

Ключевые слова: 5G, машинное обучение, многоантенные системы, формирование луча.

Заседание 25 октября 2024 года

Тема семинара: Прототипирование активных и пассивных радиокомпонентов с использованием аддитивного производства.

Докладчик: Кирилл Глинский, ИППИ РАН.

Аннотация: Этот семинар исследует применение технологии 3D-печати в изготовлении как пассивных, так и активных компонентов для систем беспроводной связи, включая антенны, линзы и реконфигурируемые интеллектуальные поверхности. В нем будет рассказано о преимуществах аддитивного производства в этой области, особенно его способность быстро создавать прототипы сложных устройств. Особое внимание уделяется повышенной скорости и гибкости процесса прототипирования, позволяющего быстро итерировать и оптимизировать дизайн компонента. Кроме того, в докладе рассматривается, как 3D-печать позволяет создавать новые структуры и материалы, которые ранее было сложно или невозможно изготовить традиционными методами.

Информация о докладчике: Кирилл Глинский в настоящее время работает над кандидатской диссертацией под научным руководством Евгения Хорова. Он получил степень магистра с отличием по прикладной математике и физике в Московском физико-техническом институте в 2022 году. Он является научным сотрудником Лаборатории беспроводных сетей Института проблем передачи информации Российской академии наук. Его научные интересы включают применение методов машинного обучения в беспроводных сетях и разработку прототипов.

Ключевые слова: аддитивное производство, беспроводные сети нового поколения, антенные системы.

Заседание 1 ноября 2024 года

Тема семинара: Применение упрощенной теории возмущений к компенсации искажений сигналов в волоконно-оптических линиях связи.

Докладчик: Никита Светличный, МФТИ, ИППИ РАН.

Аннотация: Современные волоконно-оптические линии связи характеризуются ростом скорости, мощности и дальности передачи. Эти тенденции приводят к нарастанию нелинейных искажений, препятствующих восстановлению информации из принимаемых сигналов. Разработанные научным сообществом алгоритмы нелинейной эквализации эффективны, но обладают высокой временной сложностью. Поэтому задача построения методов компенсации, сочетающих в себе низкую временную сложность и приемлемую точность, становится все более актуальной. На семинаре будет проведен обзор существующих алгоритмов компенсации искажений и детально рассмотрено применение теории возмущений (англ.: PB-NLC) к решению поставленной задачи. Будет предложено несколько способов упрощения метода PB-NLC, основанных на примечательных аналитических свойствах тензорного оператора. Эффективность разработанных алгоритмов подтверждается численными экспериментами.

Информация о докладчике: В июне 2024 г. Никита Сергеевич Светличный защитил бакалаврскую работу "Быстрые алгоритмы компенсации нелинейных искажений в волоконно-оптической линии связи на основе малоранговых тензорных аппроксимаций". В настоящее время он обучается на первом курсе магистратуры МФТИ (Физтех-школа радиотехники и компьютерных технологий). Докладчик также работает стажером-исследователем в Лаборатории беспроводных сетей ИППИ РАН. С июля 2023 г. Никита Светличный занимается научными исследованиями под руководством доктора физ.-мат. наук Андрея Леонидовича Делицына.

Ключевые слова: ВОЛС, нелинейные искажения, компенсация искажений.

Заседание 15 ноября 2024 года

Тема семинара: Алгоритм агрегации пакетов, увеличивающий пропускную способность многоканальных устройств сетей Wi-Fi 7.

Докладчик: Владислав Парошин, МФТИ, ИППИ РАН.

Аннотация: Стандарт IEEE 802.11be, известный как Wi-Fi 7, вводит революционную технологию многоканальной передачи, позволяющую передавать данные одновременно по нескольким каналам. Для обеспечения гарантированной доставки пакетов многоканальные устройства используют единое и конечное скользящее окно BlockAck, ограничивающее количество доступных пакетов на передачу. Поэтому чем больше пакетов передается на одном канале, тем меньше пакетов остается для передачи на остальных. Это приводит к нехватке пакетов на передачу и уменьшению пропускной способности многоканальных устройств. На семинаре будет предложено решение установленной проблемы. Во-первых, будет разработана теория оптимального алгоритма агрегации. Во-вторых, основываясь на этой теории, будет представлен алгоритм агрегации, увеличивающий пропускную способность при помощи выбора количества пакетов на передачу. Наконец, будут представлены результаты имитационного моделирования в среде NS-3, подтверждающие увеличение пропускной способности до 50% по сравнению с базовыми алгоритмами агрегации.

Информация о докладчике: Владислав Парошин получил степень бакалавра прикладной математики и физики в Московском физико-техническом институте (МФТИ) в 2023 году, и в настоящее время он продолжает обучение в магистратуре. С 2021 года Владислав работает в Лаборатории беспроводных сетей ИППИ РАН. Его научные интересы включают разработку и исследование алгоритмов для устройств сетей Wi-Fi, в том числе с привлечением методов машинного обучения.

Ключевые слова: Wi-Fi 7, multi-link, алгоритм агрегации пакетов.

Заседание 6 декабря 2024 года

Тема семинара: Обеспечение сверхнадежной связи для индустриального интернета вещей при помощи Wi-Fi 6.

Докладчик: Антон Карамышев, ИППИ РАН, МФТИ.

Аннотация: Приложения индустриального интернета вещей требуют низких и детерминированных задержек доставки данных, обеспечение чего является сложной технологической задачей для современной беспроводной связи. В то время как грядущие Wi-Fi 7 и Wi-Fi 8, предназначенные для обеспечения сверхнадежной связи в реальном времени, находятся в стадии разработки, возможности Wi-Fi 6, уже сейчас широко доступного на рынке, могут быть использованы для поддержки промышленных приложений. В докладе будут обобщены основные выводы, полученные в ходе промышленного проекта по обеспечению связи при помощи Wi-Fi 6 в сценарии индустриальной автоматизации. На основе представленных выводов будет сформулировано решение, удовлетворяющее требованиям к качеству обслуживания трафика в промышленных сценариях. Численные результаты показывают, что данное решение более чем в два раза увеличивает эффективную емкость сети по сравнению со стандартной настройкой Wi-Fi 6. Доклад основан на недавней публикации: *Karamyshev A., Liubogoshchev M., Lyakhov A., Khorov E.* Enabling Industrial Internet of Things with Wi-Fi 6: An Automated Factory Case Study // IEEE Trans. Ind. Inform. 2024. V 20. № 11. P. 13090–13100. https://doi.org/10.1109/TII.2024.3431086 (Q1, IF 11.7).

Информация о докладчике: Антон Карамышев окончил бакалавриат и магистратуру МФТИ в 2021 и 2023 годах соответственно. В настоящее время он обучается в аспирантуре МФТИ и трудится над кандидатской диссертацией в области телекоммуникаций. Он также работает младшим научным сотрудником в Лаборатории беспроводных сетей ИППИ РАН и является действующим членом IEEE 802.11 с правом голоса. В сферу его научных интересов входят беспроводные сети будущих поколений, а именно 5G/6G и Wi-Fi применительно к URLLC и промышленным сценариям.

Ключевые слова: Wi-Fi 6, Индустриальный интернет вещей, промышленная автоматизация.

Логинов Вячеслав Аркадьевич
Институт проблем передачи информации
им. А.А. Харкевича Российской академии наук, Москва
loginov@wnlab.ru

ПРОБЛЕМЫ ПЕРЕДАЧИ ИНФОРМАЦИИ

Том 60 **2024** **Вып. 4**

УДК 004.93'1 : 004.032.26 : 612.843.3

© 2024 г. **И.П. Николаев**

О ЗАСЕДАНИЯХ СЕМИНАРА "ЗРИТЕЛЬНЫЕ СИСТЕМЫ" В 2024 Г.

Семинар "Зрительные системы" посвящен следующим направлениям: экспериментальному исследованию и моделированию работы зрительных механизмов человека, включая цветовое зрение, обработке изображений, современным нейросетевым методам и технологиям технического зрения, томографической реконструкции и применению нейросетевых моделей для рентгеновской диагностики, анализу мульти- и гиперспектральных изображений, в том числе возникающих при дистанционном зондировании Земли, и др. Представлена общая информация о семинаре, а также подробная информация о заседаниях семинара, прошедших с начала 2024 г.

Ключевые слова: зрительные системы, зрение человека, техническое зрение, обработка изображений, нейросетевые методы, томография, рентгеновская диагностика.

DOI: 10.31857/S0555292324040090, **EDN:** IUWNBK

Общие сведения о семинаре

Семинар проходит в аудитории 307 ИППИ РАН, по четвергам, с 15:00 до 17:00, по адресу: Москва, Большой Каретный пер., д. 19, стр. 1. Руководитель семинара: заведующий лабораторией № 11, к.ф.-м.н. Илья Петрович Николаев. Семинар посвящен следующим направлениям:

- экспериментальное исследование и моделирование работы зрительных механизмов человека, включая цветовое зрение;
- обработка изображений и современные нейросетевые методы и технологии технического зрения;
- томографическая реконструкция и применение нейросетевых моделей для рентгеновской диагностики;
- анализ мульти- и гиперспектральных изображений, в том числе возникающих при дистанционном зондировании Земли.

На семинаре с докладами выступают как сотрудники лаборатории № 11 ИППИ РАН, так и приглашенные докладчики. Семинар открыт для достаточно широкого круга вопросов человеческого и технического зрения, в соответствии с научными интересами участников семинара. Желающие выступить на семинаре, пожалуйста, обращайтесь к И.П. Николаеву (i.p.nikolaev@iitp.ru).

Заседание 8 февраля 2024 г.

Тема семинара: Методы оцифровки и визуализации многослойных физических изображений без механического воздействия: история возникновения задачи, примеры объектов, полуавтоматические и автоматические методы сегментации.

Докладчик: Кулагин Петр Андреевич, м.н.с. лаборатории № 11 ИППИ РАН.

Аннотация: Виртуальное разворачивание — неинвазивный процесс восстановления плоского изображения из изображения, нанесенного на свернутую/скрученную поверхность. Наибольшую значимость эта задача получила в контексте расшифровки текста древних документов, например, папирусов Геркуланума. В докладе будет рассмотрена постановка задачи, общий пайплайн для ее решения, включая использование компьютерной томографии, а также примеры объектов, для которых применялась данная технология. В рамках пайплайна будет сделан обзор методов сегментации, рассмотрен вопрос их автоматичности и оценки качества.

Заседание 15 февраля 2024 г.

Тема семинара: Методы решения больших систем линейных уравнений при ограниченности ресурсов для хранения результата.

Докладчик: Шер Артем Владимирович, Smart Engines, МФТИ.

Аннотация: Системы линейных уравнений с большим числом неизвестных ($\sim 10^{10}$) возникают в разных аспектах прикладных задач, в том числе в компьютерной томографии высокого разрешения и обучении нейронных сетей. При этом, в силу большого объема результата, ресурсов для его хранения в памяти вычислителя может не хватить. Возникает проблема декомпозиции исходной задачи на подзадачи меньшей размерности, с возможностью получения прежнего результата. На данном семинаре будут рассмотрены такие методы, представленные в литературе.

Заседание 14 марта 2024 г.

Тема семинара: О перспективном методе калибровки RGB-сенсоров.

Докладчик: Николаев Дмитрий Петрович, д.т.н., Smart Engines, МФТИ.

Аннотация: Большинство фото- и видео-камер, с которыми нам приходится иметь дело, предназначены (в том числе) для последующего воспроизведения полученных ими цифровых изображений. Подразумевается, что человек увидит на репродукции те же цвета, что он наблюдал бы в натуре. При этом изготовление светочувствительных матриц с функциями спектральной чувствительности, близкими к человеческим, сопряжено со значительными технологическими трудностями. Поэтому, как правило, используются матрицы с тремя функциями чувствительности, которые примерно соответствуют трем областям видимого диапазона (“красной”, “зеленой” и “синей”), а для обеспечения цветовоспроизведения трехкомпонентный вектор сигналов с сенсора пересчитывается в систему координат, в которой ошибка относительно реакций стандартного наблюдателя в некотором смысле мала. Будем называть процесс установления параметров такого преобразования калибровкой. На сегодняшний день стандартным методом калибровки является вычисление параметров модели Финлейсона (root-polynomial, линейной в пространстве средних геометрических) по данным фотографирования цветовых мишеней при стандартном освещении.

В докладе будет обосновываться, во-первых, разумность использования нелинейных моделей калибровки (к которым относится и финлейсоновская) и, во-вторых, использование набора конкретных красителей и источников света. Затем будет показана принципиальная ограниченность стандартного метода при воспроизведении насыщенных цветов и предложен новый метод, основанный на контроле формы спектрального локуса.

Заседание 11 апреля 2024 г.

Тема семинара: Практика использования российского микротомографа: от отладки до результатов в прикладных задачах.

Докладчик: Гильманов Марат Ирикович, к.ф.-м.н., н.с. лаб. № 11 ИППИ РАН.

Аннотация: На семинаре будут представлены результаты первого года работы на российском микротомографе, приобретенном ИППИ РАН у компании ЭлТех-мед. Доклад будет состоять из трех частей. В первой части будет рассказано, что такое томограф и зачем он нужен. Из второй части вы узнаете, с какими проблемами мы столкнулись и как в процессе их решения почти переписали программу для сбора данных и значительно улучшили программу для томографической реконструкции. В заключении будут представлены свежие результаты геммологического исследования ювелирных изделий.

Заседание 2 мая 2024 г.

Тема семинара: Нейросетевой метод малоракурсной томографической реконструкции, максимально согласующейся с измеренными проекциями.

Докладчик: Ямаев Андрей Викторович, Smart Engines, МГУ.

Аннотация: Диссертационная работа посвящена разработке нейросетевых методов компьютерной томографии на основе малого числа проекций. Использование малого числа проекций весьма актуально, так как позволяет существенно снижать дозовую нагрузку на исследуемые объекты. Однако задача томографической реконструкции на основе малого числа проекционных углов существенно недоопределена, так как число измеренных проекций меньше размерности реконструируемого изображения. Часть информации о внутренней структуре изучаемого объекта присутствует в томографических проекциях, а часть теряется. В диссертации для восстановления потерянной информации предлагается метод, дополняющий недостающие данные на основе методов машинного обучения и согласующий их с известными данными на основе теоремы о центральном сечении. Разработанный метод позволил при сохранении точности реконструкции ускорить время работы в 1,5 раза и согласованность реконструкции к исходным проекциям в 10 раз относительно широко цитируемого алгоритма LPDR.

Заседание 23 мая 2024 г.

Тема семинара: Структура офтальмопатологии у пациентов с оперированными опухолями головного мозга.

Докладчик: Рычкова Светлана Игоревна, д.м.н., в.н.с. ИППИ РАН.

Аннотация: Доклад посвящен одному из современных направлений в области нейроофтальмологии и физиологии зрения – исследованию офтальмологических нарушений у пациентов с опухолями головного мозга.

Современные методы диагностики с использованием нейровизуализации и развитие микрохирургической техники позволили расширить показания к хирургическому лечению объемных образований головного мозга с учетом нейроофтальмологической симптоматики.

Не менее важной задачей является исследование офтальмологической патологии и ее динамики после хирургического лечения в период ремиссии. Существуют наблюдения, свидетельствующие о том, что примерно у половины больных ранний послеоперационный период характеризуется нарастанием симптоматики за счет поражения центра горизонтального взора, медиального продольного пучка, корешка отводящего нерва и лицевого нерва. Затем, появившаяся после операции симптома-

тика на протяжении раннего и отдаленного послеоперационного периодов частично регрессирует, в том числе и по сравнению с дооперационным периодом.

В предыдущих исследованиях, проведенных в НИИ развития мозга и высших достижений РУДН было показано, что у пациентов, оперированных по поводу опухоли мозжечка, как правило, наблюдаются различные саккадические нарушения и плохая стабильность взгляда. С этими нарушениями связаны и серьезные проблемы со способностью к чтению у этих пациентов (более длительное время чтения, большее количество фиксаций и регрессивных саккад, более длительная продолжительность фиксации).

Целью проводимого в настоящее время исследования на базе НИИ развития мозга и высших достижений РУДН, совместно с ИППИ им. А.А. Харкевича РАН, кафедры глазных болезней ФМБА и объединения клиник "Ясный взор", является изучение структуры офтальмопатологии и состояния зрительных функций у пациентов с оперированными опухолями головного мозга на этапе реабилитации.

Заседание 6 июня 2024 г.

Тема семинара: 4,6-битное квантование как метод быстрого и точного вычисления нейронных сетей на ЦПУ.

Докладчик: Трусов Антон Всеволодович, МФТИ, ФИЦ ИУ РАН, Smart Engines.

Аннотация: В данном докладе рассматривается новая схема квантования нейронных сетей с 4,6-битной точностью. Она обеспечивает эффективное использование ресурсов ЦПУ. Эта схема обладает бóльшим количеством квантованных уровней по сравнению с 4-битным квантованием, обеспечивая более высокую точность и сохраняя высокую вычислительную эффективность. Эксперименты с различными сверточными нейронными сетями на наборах данных CIFAR-10 и ImageNet показывают, что 4,6-битные квантованные сети на процессоре ARMv8 работают значительно быстрее, чем 8-битные сети, при этом не вызывают столь значительного падения качества, как 4-битные. Таким образом, 4,6-битное квантование может служить промежуточным звеном между быстрыми и неточными малобитными квантованными нейронными сетями и точными, но относительно медленными 8-битными моделями. Доклад подготовлен на основе статьи "4.6-Bit Quantization for Fast and Accurate Neural Network Inference on CPUs", опубликованной в журнале MDPI Mathematics (Q1 WOS & Scopus).

Заседание 08 августа 2024 г.

Тема семинара: Сопоставление оптических и радиолокационных изображений с использованием алгоритмов на основе ключевых точек.

Докладчик: Волков Владислав Владимирович, ФИЦ ИУ РАН.

Аннотация: Диссертационная работа посвящена разработке алгоритмов сопоставления оптических и радиолокационных (SAR) спутниковых снимков. Необходимость подобного сопоставления возникает в задачах детектирования объектов, обнаружения изменений, навигации, комплексирования изображений. Одним из способов сопоставления изображений является поиск ключевых точек и их сопоставление по дескрипторам-описаниям, вычисляемым для каждой точки, с последующим нахождением преобразования при помощи геометрической модели. В работе показано, что зачастую исследования алгоритмов сопоставления производятся на небольших (десятки пар изображений) закрытых датасетах оптических и SAR изображений. В данной работе предлагается собственный датасет из 100 выравненных пар изображений. Разработан гибридный алгоритм с использованием нейросети на основе U-Net и алгоритма сопоставления на основе ключевых точек для искажений изображений типа сдвига, позволяющий добиться 1-пиксельной точности сопоставления

для 93% изображений. Также разработан нейросетевой алгоритм сопоставления оптических и SAR изображений на основе сиамской нейросети и алгоритма RANSAC с геометрической моделью подобия.

Заседание 12 сентября 2024 г.

Тема семинара: Перспективы технического зрения: куда идти, если глазу не за что зацепиться.

Докладчик: Николаев Петр Петрович, д.ф.-.м.н., г.н.с. лаборатории № 11 ИППИ РАН.

Аннотация: В системах технического зрения часто возникает задача распознавания образов, подвергнутых проективному преобразованию с неизвестными параметрами. Для решения этой задачи часто используются ключевые точки, однако не на каждом контуре их легко найти. Например, существует такой класс гладких замкнутых кривых как овалы, для которого никакие стандартные подходы не применимы. В докладе будет обобщенно рассказано о многолетних исследованиях автора в области проективно-инвариантного описания овалов в различных постановках с подробным описанием алгоритма в одном из потенциально важных частных случаев. В качестве заключения будет обсуждена связь разработанного автором научного направления с актуальными задачами современного технического зрения.

Заседание 26 сентября 2024 г.

Тема семинара: Ахроматическая дальтонизация изображений с сохранением натуральности и локальных контрастов.

Докладчик: Сидорчук Дмитрий Сергеевич, м.н.с. лаборатории № 11 ИППИ РАН.

Аннотация: Дефекты цветового зрения (ДЦЗ), заключающиеся в нарушениях в работе одного из типов рецепторов-колбочек, встречаются у 5% людей. Такие дефекты снижают способность человека различать определенные цвета, что приводит к неудобствам в повседневной жизни, в том числе при использовании цветных дисплеев. Для помощи людям с ДЦЗ разрабатываются методы дальтонизации изображений. Их цель – трансформация цветных изображений таким образом, чтобы элементы изображения, хорошо различимые людьми с нормальным цветовым зрением (НЦЗ) и плохо различимые людьми с ДЦЗ, становились для последних различимы лучше. В докладе будет рассмотрена задача дальтонизации с дополнительным требованием сохранения натуральности одновременно для наблюдателя с ДЦЗ и наблюдателя с НЦЗ. Будет предложен новый метод дальтонизации изображений. Сохранение натуральности в нем обеспечивается за счет того, что модификации подвергается только яркостная компонента изображения. Повышение различимости элементов достигается в части сохранения локальных контрастов. Будет проведено сравнение нового метода с современным методом дальтонизации, также сохраняющим локальный контраст, но не ограниченным яркостной компонентой. Будет показано, что предложенный метод дает близкий уровень повышения контраста при значительно более высоком уровне сохранения натуральности относительно исходных изображений.

Заседание 17 октября 2024 г.

Тема семинара: Использование модели двух плоскостей для геометрической ректификации документов с одним произвольным сгибом.

Докладчик: Ершов Александр Михайлович, аспирант ИППИ РАН.

Аннотация: Геометрическая ректификация изображений с документами (исправление физических искажений этих документов) является важной задачей в анализе документов. Большинство существующих передовых подходов в этой области используют нейросети. Однако, несмотря на удовлетворительное качество, долгое время работы таких методов делает их непригодными для работы на мобильных устройствах. В докладе будет рассмотрен конкретный (но распространенный) случай физических искажений документа, когда он содержит ровно один, но произвольный сгиб. Будет представлен разработанный метод геометрической ректификации изображений с такими искажениями. Будет показано, что предложенный метод превосходит передовые нейросетевые решения по ключевым метрикам ректификации и может быть использован на мобильном устройстве. Оригинальная статья опубликована (`https://ieeexplore.ieee.org/document/10705295`) в IEEE Access.

Заседание 14 ноября 2024 г.

Тема семинара: Цветокоррекция и наука о цвете в киноиндустрии.

Докладчик: Докучаев Федор, DI инженер, студия постпродакшена MANGA.

Аннотация: Цифровые методы обработки изображений произвели революцию в киноиндустрии. Цифровая цветокоррекция стала неотъемлимой частью постпродакшена. Но сейчас она снова на распутье. Заканчивается трудный переход от работы с Rec.709 к RAW и scene-linear. Возникла необходимость работать с HDR. Идут поиски нового эстетического языка, свободного от влияния пленки. Нужны новые цифровые инструменты для работы с изображениями. Новым вызовам и поискам в сфере цветокоррекции и будет посвящен доклад.

Николаев Илья Петрович
Институт проблем передачи информации
им. А.А. Харкевича Российской академии наук, Москва
`i.p.nikolaev@visillect.com`

АВТОРСКИЙ УКАЗАТЕЛЬ, т. 60, 2024 г.

Адрес редакции: 127051, Москва, Б. Каретный пер., 19, стр. 1, тел. (495) 650-47-39

Оригинал-макет подготовил *Д.Ю. Ногин*

**Москва
ФГБУ «Издательство «Наука»**