Оригинальная статья / Original article

УДК 004.912 + 004.89

https://doi.org/10.21869/2223-1560-2024-28-4-154-176

Поиск и взвешивание треппин-сетов в квазициклических кодах методом поднятия и проекции мультиграфа

В. С. Усатюк [1], Ю. О. Кузнецов [1], С. И. Егоров [2] ✉

[1] ООО «Т8»
ул. Краснобогатырская, д. 44, стр. 1, г. Москва 107076, Российская Федерация

[2] Юго-Западный государственный университет
ул. 50 лет Октября, д. 94, г. Курск 305040, Российская Федерация

✉ e-mail: sie58@mail.ru

Резюме

Целью исследования *является разработка нового быстродействующего метода поиска треппин-сетов, и нового метода оценки вероятности ошибок, вызванных этими треппин-сетами, для квазициклических кодов с размером циркулянта, не являющимся простым числом.*

Методы. *Предложенный метод поиска треппин-сетов использует алгебраические свойства квазициклических кодов на графах. Применение операций подъема и проекции графа переводит задачу поиска треппин-сетов в пространство большей размерности, где треппин-сеты более различимы. Предложенный метод оценки вероятности ошибок, основанный на выборке по значимости, в сравнении с предложенным ранее методом Коула, позволяет осуществить распараллеливание вычислений без необходимости дублирования таблиц. Такой подход кратно уменьшает объем требуемой памяти и позволяет осуществлять вычисления по разделенным индексам.*

Результаты. *Предложенный метод поиска треппин-сетов удобен для аппаратной реализации, в частности, на платах-ускорителях, использующих ПЛИС. Для его реализации достаточно менее половины чиплета SLR (super logic regions) ускорителя BittWare XUP-P3R (в конфигурации с 128 Гб DDR4 ОЗУ) или ускорителя AMD Alveo U200/VCU1525 (64 Гб DDR4 ОЗУ). Это в сочетание с уменьшенными требованиями к объему ОЗУ позволяет расположить на кристалле ПЛИС AMD Virtex UltraScale+ XCVU9P [51] 5 исполнительных блоков вместо 2x, необходимых для модифицированного метода Коула. При этом ускорение поиска для матрицы с размером циркулянта 128 составит 2.5 раза. Применение предложенного метода для оценки вероятности ошибок, вызванных треппин-сетами, обеспечивает ускорение в 5.3 раза в сравнении с методом Коула для квазициклического кода с размером циркулянта 2048. Предложенный метод позволяет оценивать помехоустойчивость кода во всем диапазоне отношения сигнал/шум.*

Заключение. *Предложенный метод поиска треппин-сетов обладает высоким быстродействием и обеспечивает полноту поиска. Предложенный метод оценки вероятности ошибок, вызванных этими треппин-сетами, также обладает высоким быстродействием.*

Ключевые слова: *квазициклические LDPC-коды; треппин-сеты; методы выборки по значимости; подъем мультиграфа.*

Конфликт интересов: *Авторы декларируют отсутствие явных и потенциальных конфликтов интересов, связанных с публикацией настоящей статьи.*

Для цитирования: Усатюк В.С., Кузнецов Ю.О., Егоров С.И. Поиск и взвешивание треппин-сетов в квазициклических кодах методом поднятия и проекции мультиграфа // Известия Юго-Западного государственного университета. 2024; 28(4): 154-176. https://doi.org/10.21869/2223-1560-2024-28-4-154-176.

Поступила в редакцию 02.10.2024 *Подписана в печать 12.11.2024* *Опубликована 10.12.2024*

Search and weighting for trapping sets in quasi-cyclic codes by multigraph lift and projection method

Vasily S. Usatjuk [1], Yuri O. Kuznetsov [1], Sergey I. Egorov [2] ✉

[1] LLC "T8"
44, p. 1, Krasnobogatyrskaya str., Moscow 107076, Russian Federation

[2] Southwest State University
50 Let Oktyabrya str. 94, Kursk 305040, Russian Federation

✉ e-mail: sie58@mail.ru

Abstract

Purpose of research *is to develop a new high-speed method for searching for trapping sets, and a new method for estimating the probability of errors caused by these trapping sets for quasi-cyclic codes with a circulant size that is not a prime number.*

Methods. *The proposed method for searching for trapping sets uses the algebraic properties of quasi-cyclic codes on graphs. Using the graph lifting and projection operations, the problem of searching for trapping sets is transferred to a higher-dimensional space, where trapping sets are more distinguishable. The proposed method for estimating the probability of errors based on selection by importance, in comparison with the previously proposed Cole method, allows parallelization of calculations without the need to duplicate tables. This approach reduces the amount of required memory many times and allows calculations to be performed using separated indices.*

Results. *The proposed method of searching for trapping sets is convenient for hardware implementation, in particular, on accelerator boards using FPGAs. For its implementation, less than half of the SLR (super logic regions) chiplet of the BittWare XUP-P3R accelerator (in a configuration with 128 GB of DDR4 RAM) or the AMD Alveo U200/VCU1525 accelerator (64 GB of DDR4 RAM) is sufficient. This, combined with reduced requirements for RAM volume, allows placing 5 execution units on the AMD Virtex UltraScale+ XCVU9P FPGA [51] crystal instead of 2x, required for the modified Cole method. At the same time, the search acceleration for a matrix with a circulant size of 128 will be 2.5 times. The application of the proposed method for estimating the probability of errors caused by trapping sets provides a 5.3-fold acceleration compared to the Cole method for a quasi-cyclic code with a circulant size of 2048. The proposed method allows one to estimate the noise immunity of the code over the entire range of the signal-to-noise ratio.*

Conclusion. *The proposed method of searching for trapping sets has high performance and ensures completeness of the search. The proposed method of estimating the probability of errors caused by these trapping sets also has high performance.*

Keywords*: quasi-cyclic LDPC codes; trapping sets; importance sampling methods; multigraph lifting.*

Conflict of interest. *The authors declare the absence of obvious and potential conflicts of interest related to the publication of this article.*

For citation: Usatjuk V. S., Kuznetsov Y. O., Egorov S. I. Search and weighting for trapping sets in quasi-cyclic codes by multigraph lift and projection method // *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta = Proceedings of the Southwest State University*. 2024; 28(4): 154-176 (In Russ.). https://doi.org/10.21869/2223-1560-2024-28-4-154-176.

Received 02.10,2024 *Accepted 12.11.2024* *Published 10.12.2024*

***

Введение

Применение методов алгебраической теории графов привело к значительным успехам в помехоустойчивом кодировании, кодировании источников, машинном обучении, квантовых вычислениях и пост-квантовой криптографии [1-5]. С использованем теории графов в работе [6] установлена связь между лучшими архитектурами трансформеров глубоких нейронных сетей, такими как Mega и ChordMixer в области обработки естественного языка и обработки изображений. Исследование квази-Ньютоновских ландшафтов, ландшафтов локальной функции потерь (Бете-Гессиана) в глубоких нейронных сетях привело к изучению статистических свойств физических моделей, заданных структурными (блочными) древоподобными матрицами [7-14]. Число эквивалентных минимумов свободной энергии такой системы соответствует размеру циркулянта квазициклического кода [6]. Для усовершенствования глубоких нейронных сетей, а также нелинейных кодов (кодо-модуляционных конструкций) требуются эффективные методы поиска треппин-сетов в квазициклических кодах со сравнительно большим размером циркулянта от 256 и более.

Статья посвящена разработке методов поиска треппин-сетов и оценке вероятности ошибки, вызванной ими в квазициклических кодах с размером циркулянта, не являющимся простым числом, обладающих значительно меньшей вычислительной сложностью, чем известные.

Материалы и методы

Базовые определения

В данной работе расматриваются коды, заданные на простом поле $GF(2)\equiv F_2$. Формулы и выражения используют нумерацию с нуля. Мощность конечного множества M обозначается как $|M|$. В выражениях, где производится сложение в поле F_2, используется символ $\oplus$. Множество целых чисел обозначается как Z. Основные обозначения соответствуют обозначениям, приведенным в [15].

Определение 1. Матрица циклических перестановок (Циркулянтная матрица, Circulant Permutation Matrix, CPM) Q, $(\{Q_{ij}\}_{i,j=0}^{z-1}, z>0)$ определяется следующим образом:

$$Q_{ij}=\begin{cases}1, & \text{если } i+1\equiv j(\text{mod}z),\\ 0, & \text{иначе,}\end{cases}$$

для $i,j=0,\ldots,z-1$.

Q^{-1} обозначает нулевую матрицу **0** в $\mathbb{F}_2^{z\times z}$.

Определение 2. Проверочная матрица квазициклического кода H размером $mz\times nz$ с $m,n>0$, состоящая из цир-

кулянтных матриц размера z, выражается следующим образом:

$$H=\begin{bmatrix} Q^{a_{00}} & Q^{a_{01}} & \dots & Q^{a_{0,n-1}} \\ Q^{a_{10}} & Q^{a_{11}} & \dots & Q^{a_{1,n-1}} \\ \vdots & \vdots & \vdots & \vdots \\ Q^{a_{m-1,0}} & Q^{a_{m-1,1}} & \dots & Q^{a_{m-1,n-1}} \end{bmatrix}.$$

Определение 3. Экспоненциальная матрица (матрица сдвигов циркулянтной матрицы $E(H)$), принадлежащая $\mathbb{Z}^{m\times n}$, определяется для H следующим образом:

$$E(H)=\begin{bmatrix} a_{00} & a_{01} & \dots & a_{0,n-1} \\ a_{10} & a_{11} & \dots & a_{1,n-1} \\ \vdots & \vdots & \vdots & \vdots \\ a_{m-1,0} & a_{m-1,1} & \dots & a_{m-1,n-1} \end{bmatrix}.$$

Определение 4. Матрица-протограф или базовая матрица $M(H)$ получается из $E(H)$ заменой ее элементов, равных -1, на 0, а всех остальных - на 1.

Определение 5. Линейный код, определённый квазициклической проверочной матрицей H, обозначаемый как $C=C(H)$, является подмножеством $\mathbb{F}_2^{nz}$, определяемым следующим образом:

$$C(H)=\{x\in\mathbb{F}_2^{nz}: \quad Hx^T=0\}.$$

Определение 6. Код, заданный множеством низкоплотностных проверок на четность (LDPC-код, Low-Density Parity-Check) представляет собой линейный блочный код длиной N и размерностью K, определенный разряженной матрицей проверок на четность $H^{(N-K)\times N}$. Матрица H задает уравнения проверки четности и может быть визуализирована как двудольный граф Таннера. Например, граф Таннера для матрицы проверок на четность:

$$H=\begin{bmatrix} 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 \end{bmatrix},$$

приведен на рис. 1, слева.

Определение 7. Квазициклические низкоплотностные коды (QC-LDPC) – низкоплотностные коды, заданные квазициклической проверочной матрицей проверки на четность H. Граф Таннера QC-LDPC кода описывается матрицей проверок четности $H^{mz\times nz}$, состоящей из квадратных блоков размера $z \times z$, которые представляют собой либо нулевые матрицы, либо матрицы циклических перестановок.

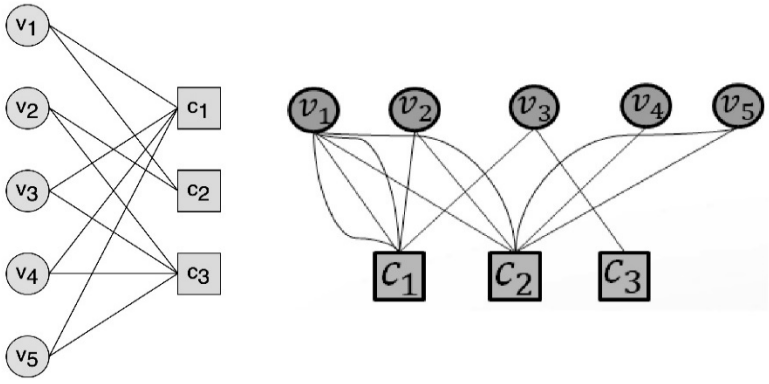

Рис. 1. Двудольный граф Таннера, заданный проверочной матрицей H (слева), протограф мультиграфа M (H_2) (справа)

Fig. 1. Bipartite Tanner graph defined by parity-check matrix H (left), protograph of multigraph $M(H_2)$ (right)

QC-LDPC- коды можно представить в виде мультиграфа, как показано на рис. 1 (справа), используя матрицу проверок на четность H_2, в которой структурные подматрицы представляют собой сумму матриц циклических перестановок:

$$H_2=\begin{pmatrix} Q_1+Q_2+Q_7 & Q_9 & Q_{23} & 0 & 0 \\ Q_{12}+Q_{37} & Q_{19} & 0 & Q_{32} & Q_{11}+Q_{12} \\ 0 & 0 & Q_{33} & 0 & 0 \end{pmatrix},$$

где сумма Q обозначает СРМ с весом > 1. Такие коды называются квазицикли-ческими кодами QC-LDPC ME-типа (Multi-Edge Type).

Определение 8. Треппин-сеты – это подграфы ($TS(a, b)$-подграфы), состо-ящие из a символьных (переменных) узлов графа Таннера, b из которых ин-цидентны проверочным узлам с нечет-ными степенями.

Определение 9. [16, определение 27] Псевдокодовым словом, образованным (a,b)-TS подграфом, является вектор $x \in \mathbb{F}_2^{nz}$, удовлетворяющий условию, что веса Хэмминга векторов x и Hx^{T} равны a и b соответственно, где a>0 и b>0.

Определение 10. Кодовым словом, образованным (a,b=0)-TS подграфом, является вектор $x \in \mathbb{F}_2^{nz}$, удовлетворяющий условию, что веса Хэмминга векторов x и Hx^{T} *равны* a и 0 соответственно, где $a>0$ и $b = 0$.

Поднятие графа квазициклических кодов

Метод поиска треппин-сетов предлагается для квазициклических матриц, у которых размер циркулянта z является произведением двух целых чисел l и z_*, причём оба числа l и z_* больше 1. Этим свойством обладает большое число применяемых на практике квазициклических матриц, в том числе матриц, адаптированных по длине квазициклических кодов, таких как коды QC-LDPC для усовершенствованной мобильной широкополосной связи (eMBB) 5G, коды спутниковой связи DVB-S2/X, тензоры и их матричные сечения в глубоких нейронных сетях и другие приложения.

Определение 11. ([15], 3.1), [17, 18]. Отображение $\mathbb{P}_{z\to z_*}$ из множества квазициклических матриц $\mathbb{F}_2^{mz\times nz}$ с размером циркулянта z в множество квазициклических матриц $\mathbb{F}_2^{mz_*\times nz_*}$ с размером циркулянта z_* определяется отображением их экспоненциальных матриц, обозначенных как $E_z \to E_{z_*}$. Здесь E_z и E_{z_*} принадлежат $\mathbb{Z}^{m\times n}$ и следуют модулярной схеме подъёма:

$$\{E_{z_*}\}_{ij}=\begin{cases} \{E_z\}_{ij} \bmod z_*, & \text{если } \{E_z\}_{ij}\geq 0, \\ -1, & \text{если } \{E_z\}_{ij}=-1. \end{cases}$$

Следует отметить, что отображение $\mathbb{P}_{z\to z_*}$ сохраняет структуру матрицы-протографа. В частности, граф Таннера квазициклической матрицы $\widetilde{H}$ с экспоненциальной матрицей $E_z=E(\widetilde{H})$, вместе с естественной проекцией, составляет l-граф-накрытие (накрыва-ющий граф, Covering graph) прове-рочной матрицы H с $E_{z_*}=E(H)$, [19].

Определение 12. Линейное отображение $P_{z\to z*}$ множества векторов F_2^{nz} в множество векторов $\mathbb{F}_2^{nz*}$ определяется следующим образом. Для произвольного $x\in\mathbb{F}_2^{nz}$ пусть $y=P_{z\to z*}x\in\mathbb{F}_2^{nz*}$. Компонента y_j, где $0\leqslant j<nz_*-1$ и $j=qz_*+r$, $0\leqslant q<n$, $0\leqslant r<z_*$, вычисляется следующим образом:

$$y_j=x_{qz+r}\oplus x_{qz+r+z*}\oplus\dots \oplus x_{qz+r+(l-1)z*}. \quad (1)$$

Если рассматривать x как кодовое слово квазициклической проверочной матрицы $\widetilde{H}$, то вектор $P_{z\to z*}x$ становится образом псевдокодового слова (как определено в [19]), соответствующего x под гомоморфизмом $\mathbb{Z}\to\mathbb{Z}_2$. Это l-накрытие определяется отображением $P_{z\to z*}$. Графовые накрытия, сопровождающиеся соответствующими операциями подъёма графа (graph lift as covering graph of a connected graph), можно рассматривать как результат применения сверточной операции, заданной полиномом задержки некоторого сверточного кода. С ростом памяти сверточного кода с "хвостозакусыванием" (Tail-Biting Convolutional Coding), растет размер циркулянта, как было показано в работах [20-22]. Причем веса кодовых слов TS(a, 0) [23-26] и псевдокодовых слов TS(a, b) [27-29] с ростом размера циркулянта приближаются к своим соответствующим верхним границам.

Определение 13. Оператор индекса проекции: функция $\pi_{z\to z*}$ определяется следующим образом:

$$\pi_{z\to z*}:\{0,1,2,\dots\}\to\{0,1,2,\dots\},$$

и действует следующим образом: для $j\geq 0$, $j=qz+r$, $0\leqslant r<z$,

$\pi_{z\to z*}(j)=qz_*+(r \bmod z_*)$.

Используя определение $\pi_{z\to z*}$, $P_{z\to z*}$ можно записать как:

$$\forall x\in\mathbb{F}_2^{nz}\quad \{P_{z\to z*}x\}_j=$$

$$=\sum_{k\in\pi^{-1}(j)} x_k,\ j=0,1,\dots,nz_*-1,$$

где суммирование выполняется в F_2.

Для произвольной квазициклической проверочной матрицы $H\in F_2^{mz\times nz}$ с размером циркулянта z, где $z=lz_*$, $l,z_*\in Z$, и $l,z_*>1$, справедлива следующая теорема.

Теорема 1. Для любого псевдокодового слова (a,b)-треппин-сета $x\in F_2^{nz}$ относительно проверочной матрицы H, вектор $x_*=P_{z\to z*}x\in\mathbb{F}_2^{nz*}$ будет (a',b')-*TS* псевдокодовым словом относительно матрицы проверки на четность $H_*=\mathbb{P}_{z\to z*}H$, при этом $a'\leqslant a$, $b'\leqslant b$, и выполняется:

(i) $b-b'=2r'$,

(ii) $a-a'=2r''$,

где $r',r''\in\{0,1,2,\dots\}$.

Следовательно, линейное преобразование

$$P_{z\to z*}:\mathbb{F}_2^{nz}\to\mathbb{F}_2^{nz*}$$

отображает линейный код $C(H)$ в некоторое подмножество кода $C(H_*)$.

Замечание 1. Случай $a'=0$ возможен, если a – четное число.

Замечание 2. Следствие, указанное в конце теоремы, напрямую вытекает из теоремы 4.4 из работы [19].

Доказательство. Для упрощения изложения доказательства мы будем использовать символы $\mathbb{P}$, P, π без индекса $z \to z_*$.

Первая часть теоремы утверждает, что количество ненулевых компонент вектора $x_*=Px$ будет таким же, как у вектора x или меньшим на четное число. Это следует из определения отображения P (1).

Докажем вторую часть теоремы.

Из определения веса Хэмминга следует, что достаточно доказать утверждение теоремы для m=1, то есть для проверочной матрицы $H \in F_2^{z \times nz}$ с одним блочным (циклическим) столбцом.

Для произвольного вектора $y \in \mathbb{F}_2^q$ через supp y мы обозначим множество индексов $j \in \{0,1,\ldots,q\text{-}1\}$, для которых y_j=1.

Пусть индекс i определяет некоторое скалярное линейное уравнение. Обозначим $\rho(i)$=1, если это уравнение не выполняется для вектора x (очевидно из контекста), и $\rho(i) = 0$ в противном случае.

Обозначим через V^0 и V_*^0 подмножества индексов вершин $\{i\}_{\mathrm{i}=0}^{nz\text{-}1}$ и $\{i\}_{i=0}^{nz_*\text{-}1}$ соответственно, соответствующие ненулевым циркулянтам матриц $\mathbb{F}_2^{z \times nz}$ и $\mathbb{F}_2^{z_* \times nz_*}$.

Для уравнения Hu=0 (H_*v=0), содержащего ненулевой компонент вектора x (x_*), зависящего только от supp $x \cap V^0$ (supp $x_* \cap V_*^0$), и, кроме того, для $\tilde{x}$, где supp $\tilde{x}$=supp $x \cap V^0$, выполняется равенство

$$\mathrm{supp}\ (\mathrm{P}\tilde{\mathrm{x}}) = \mathrm{supp}\ (Px) \cap V_*^0.$$

можно считать, что supp $x \subset V^0$, supp $x_* \subset V_*^0$. Тогда выполняются соотношения

$$\pi(V^0)=V_*^0, \quad \mathrm{supp}\ x_* \subset \pi(\mathrm{supp}\ x).$$

Структуры квазициклических матриц H и H_* однозначно определяют отображения

$$\mathrm{c}(\cdot): \quad V^0 \to \{0,1,\ldots,z\text{-}1\}$$
$$\mathrm{c}_*(\cdot): \quad V_*^0 \to \{0,1,\ldots,z_*\text{-}1\},$$

соответствующие индексу вершины j строки индекса i, так что элемент соответствующей проверочной матрицы в позиции (i,j) равен 1.

Докажем равенство

$$c_*(\pi(j))=\pi(c(j)), \quad j \in V^0. \qquad (2)$$

Фиксируем произвольный индекс $j \in V^0$, j=qz+r, 0<q<n, 0<r<z. Тогда $c(j)$= =(r-a_q) mod z, где a_q – экспонента q-го циркулянта матрицы H, и операция $a \bmod z$ определена так, что результат лежит в интервале [0,z-1] для любого целого a. Справедлива цепочка равенств:

$$\pi(c(j))=((r\text{-}\mathrm{a_q}) \bmod z) \bmod z_* =$$
$$=(r\text{-}a_q) \bmod z_* = (r \bmod z_* - a_q \bmod z_*) \bmod z_* =$$
$$=c_*(qz_*+r \bmod z_*)=c_*(\pi(\mathrm{j})).$$

Отображение

$$\pi: \{0,1,\ldots,nz\text{-}1\} \to \{0,1,\ldots,nz_*\text{-}1\}$$

определяет разбиение множества V^0 на непересекающиеся подмножества

$$W_j := \pi^{-1}(\mathrm{j}), j \in V_*^0.$$

По определению отображения P множество индексов supp x_* однозначно соответствует таким W_j, для которых $W_j \cap$ supp x состоит из нечетного числа элементов. Обозначим этот факт как $W_j \sim$ supp x_*.

Пусть $\sigma:\mathbb{Z}\to\mathbb{Z}_2$ – гомоморфизм аддитивных групп. Тогда можно сформулировать следующую лемму.

Лемма 1. Пусть W — произвольное семейство множеств W_j, $j\in V_*^0$.

$\widehat{W}:=\{W\in W:\ W\sim\operatorname{supp} x_*\}$.

Тогда следующие проверки совпадают:

$$\sigma(|\widehat{W}|)=\sigma(|\cup_{W\in W}\{W\cap\operatorname{supp} x\}|). \quad (3)$$

Доказательство. Предположим сначала, что $\widehat{W}=W$. Учитывая, что множества в объединении в правой части (3) не пересекаются, получим:

$$\left|\bigcup_{W\in W} W\cap\operatorname{supp} x\right| =$$
$$=\sum_{W\in W}|\{W\cap\operatorname{supp} x\}|=$$
$$=\sum_{W\in W}(1+2s_W)=|W|+2s,$$

где s_W, s – некоторые целые числа. Это доказывает лемму в специальном случае. Если $\widehat{W}\neq W$, то количество индексов supp x в каждом множестве $W\setminus\widehat{W}$ будет либо нулевым, либо положительным четным числом.

Обозначим через C^a, C_*^a множества активных проверок четности (варьируемых, при фиксированной остальной части) для H, H_* относительно векторов x, x_*:

$$C^a=c(\operatorname{supp} x),\quad C_*^a=c_*(\operatorname{supp} x_*),$$

тогда справедлива следующая лемма.

Лемма 2. Для любого $s\in C_*^a$ существует непустое множество индексов $\{i_1^s,\ldots,i_{k(s)}^s\}$, $i_1^s\in C^a$, $l=1,\ldots,k(s)$, такое что

$$\rho(s)=\rho(i_1^s)\oplus\ldots\oplus\rho(i_{k(s)}^s),$$

и для различных s',s'' множества индексов не пересекаются.

Доказательство. Зафиксируем $s\in C_*^a$ и рассмотрим уравнение

$$s=c_*(\pi(j)),\quad j\in\operatorname{supp} x. \quad (4)$$

Множество решений уравнения (4) обозначим через R=R(s). Из равенства следует, что следующие множества либо полностью содержатся в R(s), либо не пересекаются с ним:

$$\operatorname{supp} x\cap W_j, j\in V_*^0, \quad (5)$$

$$\operatorname{supp} x\cap c^{-1}(i), i=0,1,\ldots,z-1. \quad (6)$$

Непустые пересечения (5), (6) образуют два разбиения этого множества. Определим

$$W(s):=\{W_j:\ \operatorname{supp} x\cap W_j\subset R(s), j\in V_*^0\}.$$

В множестве $\{i_l^s\}_{l=1}^{k(s)}$ возьмем различные индексы i, $0\leqslant i<z-1$, для которых множества (6) непусты и содержатся в $R(s)$. Количество решений уравнения $c_*(j_*)=s$, $j_*\in\operatorname{supp} x_*$ равно числу таких $a\in W(s)$, что $a\sim\operatorname{supp} x_*$. Следовательно, из леммы 1 получаем равенство:

$$\rho(s)=\sigma W\in W(s):W$$
$$\sim\operatorname{supp}x_*=\sigma(|R(s)|)=$$
$$=\sigma\left(\left|\bigcup_{l=1}^{k(s)}\{\operatorname{supp} x\cap c^{-1}(i_1^s)\}\right|\right)=$$
$$=\sum_{l=1}^{k(s)}\sigma(|\{\operatorname{supp} x\cap c^{-1}(i_1^s)\}|)=\sum_{l=1}^{k(s)}\rho(i_1^s),$$

где суммирование производится в $\mathbb{F}_2$.

Если $s'\in C_*^a$, $s'\neq s$, то множество индексов $\{i_l^{s'}\}_{l=1}^{k(s')}$ не может пересекаться с множеством $\{i_l^s\}_{l=1}^{k(s)}$ из-за равенства (2).

Замечание 3. В определенных сценариях возможно, что нет решений j для

уравнения y_j=1, где j проецируется в supp x_* согласно определению, приведенному в уравнении (4). Повторение рассуждений, представленных в доказательстве леммы, делает очевидным факт, что в таких случаях количество активных невыполнимых проверок, обозначаемых как $\{i_l^s\}_{l=1}^{k(s)}$, будет четным.

Из Леммы 2 следует, что для любой невыполнимой проверки $s\in C_*^a$, где $\rho(s)=1$, количество невыполнимых проверок среди $\{i_l^s\}_{l=1}^{k(s)}\subset C^a$ будет либо равно 1, либо превышать 1 на четное число.

В соответствии с замечанием 3 проверки из C^a, не связанные с C_*^a, будут давать четное число невыполненных проверок.

Теорема 1 доказана.

Числовой пример. Рассмотрим квазициклическую проверочную матрицу H с размером циркулянта z=128 и экспоненциальной матрицей

$$E(H)=[E_1|E_2], \text{ где} \quad (7)$$

$$E_1=\begin{bmatrix} 21 & 65 & 126 & 39 & 84 & 89 & 16 & 89 & 94 & 28 \\ 12 & 29 & 27 & 105 & 0 & 0 & 0 & 2 & 108 & 78 \\ 47 & 26 & 97 & 40 & 66 & 0 & 81 & 1 & 33 & 97 \\ 113 & 21 & 64 & 26 & 53 & 103 & 85 & 81 & 102 & 48 \end{bmatrix}, \quad (8)$$

$$E_2=\begin{bmatrix} 31 & 12 & 101 & 9 & 122 & 99 & 90 & 65 & 41 & -1 \\ 126 & 98 & 27 & 1 & 41 & 118 & 49 & 17 & 12 & 11 \\ 89 & 37 & 52 & 42 & 2 & 120 & 38 & -1 & 86 & 7 \\ 113 & 98 & 71 & 70 & 121 & 66 & 95 & 40 & -1 & 23 \end{bmatrix}. \quad (9)$$

Для матрицы H выполнена операция подъема графа (graph lifting), обратная операции проекции. Для этого использовалась бинарная матрица $B=\{b_{ij}\}$, i=0,…,3, j=0,…,19, элементы которой выбраны из псевдослучайного бинарного распределения. Полученная матрица $\widetilde{H}\in\mathbb{F}_2^{1024\times5120}$, $\tilde{z}=2z=256$, определена своей экспоненциальной матрицей $\widetilde{\mathrm{E}}=\mathrm{E}(\widetilde{\mathrm{H}})$,

$$\{\widetilde{E}\}_{ij}=\begin{cases}\{E\}_{ij}+b_{ij}\mathrm{z}, & \{E\}_{ij}\geq 0,\\ -1, & \{E\}_{ij}=-1,\end{cases}$$

где $\mathrm{E}=E(H)$.

Очевидно, что $H=\mathbb{P}_{256\to128}(\widetilde{H})$.

Приближенным методом было определено, что в пространстве $\mathbb{F}_2^{5120}$ существует 874903 (a,b)-TS слов при условиях $a\leqslant30$ и $b\leqslant a$. Для каждого из этих слов выполнялась проекция с использованием операции $P_{256\to128}$ (см. определение 12). Полученные слова были разбиты по классам (a',b'). Распределение разностей (a-a',b-b') представлено в табл. 1. Обратите внимание, что два класса $(a,b)\to(a,b)$, $(a,b)\to\to(a,b\text{-}2)$составляют 99.26% всех случаев для данной квазициклической проверочной матрицы и соответствующего ей подъема графа.

Поиск треппин-сетов в квазициклических кодах

Предположим, что у нас есть квазициклическая проверочная матрица $H\in\mathbb{F}_2^{mz\times nz}$, $m,n,z>0$. Группа автоморфизмов ([32][Гл. 8, § 5]) линейного кода $C(H)$ содержит подгруппу $\mathcal{G}_{\mathrm{n,z}}$ квазициклических сдвигов. $\mathcal{G}_{\mathrm{n,z}}$ – циклическая группа порядка z с порождающей

функцией π_0, которая действует на индексы j, $0 \leqslant j < nz-1$, $j=qz+r$, $0 \leqslant r < z$ следующим образом:

$$\pi_0(j)=qz+(r+1) \bmod z.$$

Таблица 1. Распределение разностей $(a-a', b-b')$

Table 1. Distribution of differences $(a-a', b-b')$

$a-a'$	$b-b'$	Частота, %
0	0	93.071689
0	2	6.191543
0	4	0.191678
0	6	0.002743
2	0	0.360154
2	2	0.152474
2	4	0.016458
2	6	0.000571
4	0	0.006400
4	2	0.003543
4	4	0.001257
6	0	0.000685
6	2	0.000457
6	4	0.000228
8	4	0.000114

Запишем в виде нелинейного оператора $D_k: \mathbb{R}^{nz} \to \mathbb{R}^{nz}$ $k>0$ шагов абстрактного декодера.

Если $x \in \mathbb{F}_2^{nz}$ – некоторое (a,b)-TS псевдокодовое слово, $a>0$, и $\pi_0 \in \mathcal{G}_{n,z}$ коммутирует с D_k, $k=1,2,\ldots$, то в пределе вероятность ошибки декодирования [33] [Шаг 2] будет одинакова для всей орбиты $\{\pi x: \pi \in G_{n,z}\}$. Кроме того, автоморфизмы $\mathcal{G}_{n,z}$ коммутируют с проекторами P (определение 12) псевдокодовых слов треппин-сетов. Отсюда вытекает следующее утверждение.

Лемма 3. Для любого делителя z_* и любого $s=0,1,\ldots,z-1$, справедлива следующая диаграмма:

$$\begin{array}{ccc} \mathbb{F}_2^{nz} & \xrightarrow{\pi_0^s} & \mathbb{F}_2^{nz} \\ \downarrow{\scriptstyle P_{z\to z_*}} & & \downarrow{\scriptstyle P_{z\to z_*}} \\ \mathbb{F}_2^{nz_*} & \xrightarrow{\pi_0^{s \bmod z_*}} & \mathbb{F}_2^{nz_*} \end{array}$$

Доказательство. Следует из определений $P_{z\to z_*}$ и $\pi_0 \in \mathcal{G}_{n,z}$.

Из предыдущих обсуждений и леммы 3 ясно, что в процессе поиска псевдокодовых слов достаточно оставлять слова попарно не эквивалентные друг другу.

Выберем некоторый приближенный метод (обозначим его solve) поиска множества X псевдокодовых слов для квазициклической проверочной матрицы $H \in \mathbb{F}_2^{mz \times nz}$, которые не эквивалентны попарно с точки зрения группы $\mathcal{G}_{n,z}$. Существует множество критериев выбора таких треппин-сетов. Приведем две ключевые стратегии применения предложенного метода:

• Стратегия 1. Получить максимальное количество $x \in \mathbb{F}_2^{nz}$ с минимальным расстоянием d_ε^2 до требуемой вероятности ошибки при фиксированной модели канала с неравновероятным распределением ошибок по отношению к треппин-сетам.

• Стратегия 2. Получить максимальное количество $x \in \mathbb{F}_2^{nz}$ таких, что $x \sim (a,b)$ при условиях $a \leqslant w_{\max}$, $v \leqslant v(a)$, где в простейшем случае $v(a)=w$. Подразумевается равновероятное распределение

ошибок в вершинах b подграфа TS(a,b) треппин-сета.

Приведем **Алгоритм 1** для поиска треппин-сетов в квазициклических кодах

Вход: $E(H)$, $N>0$ – количество поднимаемых матриц E_j.

Выход: массив узлов треппин-сетов, образующих псевдокодовые слова.

Строим поднятые (lifting) матрицы E_j экспоненциальной матрицы $E(H)$ с использованием B_j (см. числовой пример 1) и соответствующих проверочных матриц H_j, j=0,…,N-1.

Вычисляем $X_{2z,j}$:=solve(H_j) согласно одной из стратегий.

Вычисляем $X_{z,j}$:=$P_{2z\to z}X_{2z,j}$.

Находим максимальное подмножество множества $\cup_{j=0}^{N-1}X_{z,j}$, состоящее из попарно неэквивалентных псевдокодовых слов.

Алгоритм 1 имеет вычислительную сложность, определяемую преимущественно шагом 4.

Предложенный метод поиска треппин-сетов использует алгебраические свойства квазициклических кодов на графах. С использованием операций подъема и проекции графа задача поиска треппин-сетов переводится в пространство большей размерности, где треппин-сеты более различимы.

Поиск потенциальных треппин-сетов ограничивается орбитами, созданными квазициклическими сдвигами. Поскольку автоморфизмы коммутируют с этими квазициклическими сдвигами, треппин-сеты демонстрируют согласованные шаблоны во всех кодовых словах в пределах заданной орбиты. Предварительное определение репрезентативных элементов каждой орбиты значительно сокращает пространство поиска, при этом охват потенциальных треппин-сетов сохраняется.

Пример работы метода поиска треппин-сетов

Рассмотрим квазициклическую проверочную матрицу с размером циркулянта 128, заданную формулами (7) - (9), и используем 32 поднятые матрицы (N=32), (см. Алгоритм 1). В результате 20 часов вычислений с использованием четырех параллельных потоков на процессоре Intel Xeon v4 E5-2696 было найдено общее количество 28,623,960 различных (a,b)-TS, (использовалась стратегия 1). Распределение этих треппин-сетов по классам (a, b) представлено в табл. 2-4.

При поиске треппин-сетов использовался объем ОЗУ в 7.4 Гб, что в 2.16 раза меньше по сравнению с предложенным в работе [30] методом при приблизительно равной полноте поиска (вероятности пропуска треппин-сетов).

Предложенный метод поиска треппин-сетов удобен для аппаратной реализации, в частности, на платах-ускорителях, использующих ПЛИС. Для его реализации достаточно менее половины чиплета SLR (super logic regions) ускорителя BittWare XUP-P3R (в конфигурации с 128 Гб DDR4 ОЗУ) или ускорителя AMD Alveo U200/ VCU1525 (64 Гб DDR4 ОЗУ).

Таблица 2. Распределение треппин-сетов (*a*, *b*)

Table 2. Distribution of the trapping-sets (*a*, *b*)

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
1	0									
2	0	0								
3	0	0	0							
4	0	0	0	1						
5	0	0	0	2	33					
6	0	0	1	4	38	803				
7	0	0	0	1	82	1079	11109			
8	0	0	0	10	125	1886	13722	82004		
9	0	0	1	11	184	2212	23952	74786	257967	
10	0	0	0	20	210	3122	20028	150662	221759	537954
11	0	0	0	14	249	2548	23038	89105	523548	434372
12	0	0	1	26	249	2288	14748	87080	253567	1141220
13	0	0	0	10	168	1638	10379	49714	199393	485167
14	0	0	0	14	153	1217	7100	32254	113767	314533
15	0	0	0	11	105	781	4771	20972	70765	183229
16	0	0	2	9	97	578	3331	14007	46158	116059
17	0	0	0	5	51	414	2116	9305	30284	75122
18	0	0	0	3	54	258	1451	6401	20480	49946
19	0	0	0	0	28	168	1013	4129	13555	33001
20	0	0	1	1	21	116	658	2759	9017	22248
21	0	0	0	1	6	73	394	1808	6010	14602
22	0	0	0	1	8	50	247	1184	3860	9805
23	0	0	0	0	3	39	176	746	2511	6446
24	0	0	0	0	0	24	91	499	1665	4356
25	0	0	0	0	1	12	81	294	1023	2759
26	0	0	0	0	0	7	51	196	672	1893
27	0	0	0	0	1	6	26	144	416	1226
28	0	0	0	0	0	4	15	73	269	798
29	0	0	0	0	0	2	10	55	179	488
30	0	0	0	0	0	0	5	36	113	327

Это в сочетании с уменьшенными требованиями к объему ОЗУ позволяет расположить на кристалле ПЛИС AMD Virtex UltraScale+ XCVU9P [31] 5 исполнительных блоков вместо 2-х, необходимых для метода [30].

Таким образом, ускорение поиска для матрицы с размером циркулянта 128 составит 2.5 раза. Сравнительно малый объем требуемого методом ОЗУ позволяет применять ускорители с высокоскоростной HBM памятью (HBM, High Bandwidth Memory), что обеспечит дальнейшее кратное ускорение поиска треппин-сетов.

Кроме того, данный подход позволяет применять более дешевые ускорители с меньшим объемом доступных аппаратных ресурсов.

Таблица 3. Распределение треппин-сетов (*a*, *b*)

Table 3. Distribution of the trapping-sets (*a*, *b*)

	11	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**
11	679776								
12	525023	461020							
13	1554985	389114	258537						
14	657114	1513059	256509	148743					
15	388532	708806	1321139	151631	57480				
16	229091	418952	672534	1008359	72773	31479			
17	147227	250979	403669	568414	710717	38932	10705		
18	96483	163431	244586	355260	445490	486945	16008	5903	
19	64579	108837	161037	219709	296733	331568	310638	8197	2002
20	43578	72524	108737	146508	189905	231101	234722	201297	3294
21	29199	49859	73927	99409	127728	153153	176151	162620	123395
22	19921	33911	50844	68599	88005	104483	121419	127482	108790
23	13300	23202	35095	48044	60763	73618	84312	92364	91845
24	9024	15971	24386	33526	42439	51402	59559	65539	69455
25	5995	10959	17133	23651	30270	36385	42387	47275	51012
26	4229	7595	11775	16658	21354	25734	30061	34101	36835
27	2736	5223	8275	11831	15478	18698	21929	24431	27141
28	1832	3441	5762	8268	11245	13586	15902	18157	19773
29	1222	2310	3990	5878	7974	9774	11568	13128	14457
30	813	1537	2753	4178	5723	7115	8728	9757	10807

Таблица 4. Распределение треппин-сетов (*a*, *b*)

Table 4. Distribution of the trapping-sets (*a*, *b*)

	20	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**
20	1199										
21	1696	516									
22	77348	748	311								
23	71948	47293	429	194							
24	63799	46955	28875	252	112						
25	49967	44223	30251	17646	168	70					
26	38506	36280	29742	19657	10920	106	52				
27	28602	28679	25804	19810	12566	6552	72	35			
28	21540	21716	21008	17841	13092	7953	4066	41	32		
29	15185	16371	16191	14971	12051	8647	4949	2507	0	0	
30	11685	12146	12662	12214	10673	8224	5699	3093	1485	0	0

Метод ускоренной оценки вероятностей ошибок, вызванных треппин-сетами, для квазициклических кодов

В данном разделе описывается метод ускоренной оценки вероятности ошибок, вызванных треппин-сетами, полученный на основе модификации метода Коула, [33, шаг 3]. Предложенный метод в отличие от метода Коула использует свойство квазицикличности

кодов. Метод основан на применении предварительного табличного расчета, в соответствии с [34, § 4.1].

Как и в предыдущих разделах, в описании метода используются следующие обозначения: $H\in\mathbb{F}_2^{mz\times nz}$, $m,n,z>0$ – проверочная матрица, определяющая линейный код C(H), $D_k:\mathbb{R}^{nz}\to\mathbb{R}^{nz}$ – декодер с $k>0$ итерациями, $\mathcal{G}_{n,z}$ – циклическая группа автоморфизмов кода C(H), действующая слева на множествах $\mathbb{F}_2^{nz}$, $\mathbb{R}^{nz}$ с генератором π_0, $s:=nz$.

Вероятность ошибки при декодировании кодового слова длины s с нормально распределенным шумом $\xi\sim\mathcal{N}(\theta,\Sigma)$, $\Sigma=\mathrm{diag}(\sigma^2,\ldots,\sigma^2)$ равна [34, формула 5]:

$$P_f=\int_{R^N} I_e(y)w(y)f^*(y)\,\mathrm{d}y, \qquad (10)$$

где $\mathrm{I_e}(\cdot)$ – характеристическая функция некодовых слов из $\mathbb{R}^s$, $f^*(\cdot)$ – смещенное плотное распределение,

$$f^*(y)=1/|V|\sum_{x\in V} f(y,x),\quad V\subset R^s,\ y\in R^s$$

$$f(y,x)=(2\pi\sigma^2)^{-s/2}\exp(-1/2\sigma^2\,\|y-x\|^2_{\mathbb{R}^s}),$$

$$x,y\in\mathbb{R}^s$$

$$w(y)=f(y,c)/f^*(y),\quad y\in\mathbb{R}^s$$

$f(\cdot,\mathrm{x})$ – плотность распределения s-мерной нормальной случайной величины, $f\sim\mathcal{N}(x,\Sigma)$. $c=(1,\ldots,1)\in\mathbb{R}^s$ – нулевое кодовое слово после двоичной фазовой манипуляции. Здесь $\|\cdot\|$ обо-значает евклидову норму.

Как отмечено в предыдущем разделе, $d_\varepsilon^2(\mathrm{y})=d_\varepsilon^2(\pi y)$, $\pi\in\mathcal{G}_{n,z}$, $y\in\mathbb{F}_2^N$ при условии $D_k\pi=\pi\mathrm{D_k}$, $k=1,2,\ldots$. Следовательно, разумно выбрать множество V в виде набора непересекающихся полных $\mathcal{G}_{n,z}$-орбит элементов. Будем обозначать эти элементы $V_0=\{y_0^{\mathrm{b}},\ldots,y_{\mathrm{p-1}}^{\mathrm{b}}\}$, $y_i^b\nsim y_j^b$, $i\neq j$, $y_j^b\in\mathbb{R}^N$. Затем

$$V=V_0\sqcup\pi_0V_0\sqcup\ldots\sqcup\pi_0^{\mathrm{z-1}}V_0\ \ |V|=pz,$$

$$f^*(\cdot)=1/z\sum_{\mathrm{j}=0}^{\mathrm{z-1}} f_j^*(\cdot),$$

где $f_j^*(\cdot)=1/p\sum_{\mathrm{x}\in\pi_0^{\mathrm{j}}\mathrm{V}_0} f(\cdot,\mathrm{x})$, $j=0,\ldots,\mathrm{z-1}$.

Элементы отображения (кодирования) $0,1\in\mathbb{F}_2$ представлены числами $1,-1\in\mathbb{R}$ соответственно. Следовательно, для TS $x\in\mathbb{F}_2^s$ слово можно выразить как $c\text{-}\mu x\in\mathbb{R}^s$, где $\mu>0$ – параметр метода. Для упрощения вычислений мы используем естественное вложение $\mathbb{F}_2^s\subset\mathbb{R}^s$, переводя элементы $\{0,1\}\in\mathbb{F}_2$ в числа $\{0,1\}\in\mathbb{R}$. "Базисные" вектора из V_0 можно представить в виде $y_j^b=c\text{-}\mu x_j^b$, где $x_j^b\in\mathbb{R}^s$, $j=0,\ldots,p\text{-}1$, $\{x_j^b\}_{\mathrm{i}}\in\{0,1\}$, и $i=0,\ldots,s\text{-}1$.

Следующее утверждение упрощает выражение P_f для такого выбора множества V. Предложим, что $\pi D_{\mathrm{k}}=D_k\pi$, для всех $\pi\in\mathcal{G}_{n,z}$, $k=1,2,\ldots$. Тогда

$$P_f=\int_{R^s} I_e(y)\mathrm{w}(y)f_0^*(y)\,\mathrm{d}y. \qquad (11)$$

Доказательство.

$$P_f=1/z\sum_{\mathrm{j}=0}^{z-1}\int_{\mathrm{R^N}} I_e(y)w(y)f_j^*(y)\,\mathrm{dy}=:1/z\sum_{j=0}^{z-1}A_j.$$

Коммутативность π и D_k дает равенство $I_e(\pi y)=I_e(y)$, $y\in\mathbb{R}^s$. Также легко проверить, что $\mathrm{f}(\pi\mathrm{y},\mathrm{x})=f(y,\pi^{-1}\mathrm{x})$, $f^*(\pi y)=f^*(y)$, $\forall x,y\in R^N$, $\pi\in G_{n,z}$.

Произведем замену переменной в выражении для A_{j}:

$$A_j=\int\limits_{R^s} I_e(\pi_0^{\mathrm{j}}u)\frac{\mathrm{f}(\pi_0^{\mathrm{j}}u,c)}{\mathrm{f}^*(\pi_0^{\mathrm{j}}u)}\mathrm{f}_{\mathrm{j}}^*(\pi_0^{\mathrm{j}}u)\,\mathrm{du}=$$

$$\int\limits_{\mathrm{R^s}} I_e(u)\frac{\mathrm{f}(u,\pi_0^{-\mathrm{j}}c)}{\mathrm{f}^*(u)}\mathrm{f}_0^*(u)\,\mathrm{du}=A_0,$$

где $\pi c=c$.

Приближение (11) можно получить с использованием метода Монте-Карло:

$$P_f=1/L\ \sum_{\mathrm{l}=0}^{L-1} I_e(y_1)w(y_l), \qquad (12)$$

где y_l выбирается из плотности распределения $f_0^*(\cdot)$. Вычисление значения (12) может быть дополнительно упрощено.

$y_l=y^b_{l\bmod p}+\xi_{\mathrm{l}}=c-\mu x^b_{l\bmod p}+\xi_{\mathrm{l}}$, $l=0,1,\ldots,$

$\xi_{\mathrm{l}}\sim N(\theta,\Sigma)$.

Преобразуем выражение для $a(y_l)$:

$$\mathrm{w}(y_1)=\mathrm{f}(y_1,c)/\mathrm{f}^*(y_1)=\mathrm{zp}\frac{\mathrm{R}(\mathrm{y}_1)}{\mathrm{S}(y_l)},$$

$\mathrm{l}=1,2,\ldots.$

$$\mathrm{R}(y_1)=\exp\left(-1/2\sigma^2\ \|y_l-c\|^2_{R^s}+\delta\right)=$$

$$=\exp\left(-1/2\sigma^2\ \|\mu x^b_{l\bmod p}-\xi_{\mathrm{l}}\|^2_{\mathrm{R^s}}+\delta\right) \qquad (13)$$

$$\mathrm{S}(y_1)=\sum_{\mathrm{j}=0}^{\mathrm{z-1}}\sum_{x\in\pi_0^{\mathrm{j}}\mathrm{V}_0}\exp\left(-1/2\sigma^2\ \|y_l-x\|^2_{\mathrm{R^s}}+\delta\right)=$$

$$=\sum_{\mathrm{j}=0}^{\mathrm{z-1}}\sum_{\mathrm{k}=0}^{\mathrm{p-1}}\exp\Big(-1/2\sigma^2\|\mu(\pi_0^{\mathrm{j}}\mathrm{x}^{\mathrm{b}}_{\mathrm{k}}-\mathrm{x}^{\mathrm{b}}_{\mathrm{l}\bmod \mathrm{p}})+\xi_{\mathrm{l}}\|^2_{\mathrm{R^s}}+\delta\Big), \qquad (14)$$

где δ – вещественное число, используемое в вычислениях для нормализации аргумента экспоненты.

В последнем выражении (14) поменяем порядок суммирования, и сумму показателей степеней по j от 0 до $z-1$ обозначим S_{lk}. Тогда $\mathrm{S}(y_1)=\sum_{\mathrm{k}=0}^{\mathrm{p-1}}\mathrm{S_{lk}}$. Для любой пары l,k, где $\mathrm{k}=0,\ldots,\mathrm{p}-1$, $l\geq 0$, выберем два подмно-жества индексов j:

$$J^l_{lk}=\{j:\ 0\leqslant \mathrm{j}<z-1,\ \pi_0^{\mathrm{j}}(\mathrm{supp\ x^b_k})\cap$$

$$\cap\mathrm{supp}\ x^b_{l\bmod p}\neq\emptyset\}\ ; \qquad (15)$$

$$J^2_{\mathrm{lk}}=\{\mathrm{j}:\ 0\leqslant j<z-1,\ \pi_0^{\mathrm{j}}(\mathrm{supp\ x^b_k})\cap$$

$$\cap\mathrm{supp\ x^b_{l\bmod p}}=\emptyset\}. \qquad (16)$$

Разобъем величину S_{lk} на две части:

$$S_{lk}=\sum_{j\in J^l_{lk}}\exp(\ldots)+\sum_{j\in J^2_{lk}}\exp(\ldots)=S^1_{lk}+S^2_{lk}.$$

Вычисление величин S^1_{lk} и S^2_{lk} может быть упрощено.

$$S^l_{lk}=\exp\left(-1/2\sigma^2\ \|\xi_{\mathrm{l}}\|^2_{R^s}+\delta\right)\times$$

$$\sum_{\mathrm{j}\in J^l_{lk}}\exp[-\mu^2/2\sigma^2\ |(\pi_0^{\mathrm{j}}(\mathrm{supp}\ x^b_k))\Delta$$

$$\Delta(\mathrm{supp}\ x^b_{l\bmod p})|$$

$$+\mu/\sigma^2(\sum\nolimits_{\mathrm{i}\in\mathrm{supp}\ x^b_{l\bmod p}\setminus\pi_0^{\mathrm{j}}(\mathrm{supp}\ x^b_k)}\{\xi_{\mathrm{l}}\}_{\mathrm{i}} \qquad (17)$$

$$-\sum\nolimits_{\mathrm{i}\in\pi_0^{\mathrm{j}}(\mathrm{supp}\ x^b_k)\setminus\mathrm{supp}\ x^b_{l\bmod p}}\{\xi_{\mathrm{l}}\}_{\mathrm{i}})],$$

где Δ – симметрическая разность множеств, и

$$S^2_{lk}=\exp(-1/2\sigma^2\ \|\xi_{\mathrm{l}}\|_{\mathrm{R^s}}-\mu^2/2\sigma^2\ (|\mathrm{supp}\ x^b_k|$$

$$+|\mathrm{supp}\ x^b_{l\bmod p}|)+\mu/\sigma^2\times\sum_{\mathrm{i}\in\mathrm{supp\ x^b_{l\bmod p}}}\{\xi_{\mathrm{l}}\}_{\mathrm{i}}+\delta)$$

$$\times\sum\nolimits_{j\in\mathrm{J^2_{lk}}}\exp\left(-\mu/\sigma^2\sum\nolimits_{i\in\pi_0^{\mathrm{j}}(\mathrm{supp}\ x^b_k)}\{\xi_{\mathrm{l}}\}_{\mathrm{i}}\right). \qquad (18)$$

Свойством равенств (17), (18) является то, что вычисления разделены на не зависящие от индекса j части и части, зависящие от него. Для упрощения прямых вычислений достаточно сохранить две таблицы множеств:

$$T^1_{klj}=\pi_0^{\mathrm{j}}(\mathrm{supp\ x^b_k})\setminus\mathrm{supp}\ x^b_l \qquad (19)$$

$$T^2_{klj}=\mathrm{supp}\ x^b_{\mathrm{l}}\setminus\pi_0^{\mathrm{j}}\left(\mathrm{supp}\ x^b_k\right), \qquad (20)$$

где $k,l=0,\ldots,p-1$, $j=0,\ldots,z-1$. Семейства множеств (19), (20) связаны простыми

отношениями: $\pi_0^{z-j} T^1_{k,l,j}=T^2_{l,k,z-j}$, k,l=0,…,p-1, j=0,…,z-1. На практике большинство множеств (19), (20) не пересекаются.

Результаты и их обсуждение

При реализации метода ускоренной оценки вероятностей ошибок совместно с предложенным методом поиска TS (Алгоритм 1, число поднятых матриц N=32,) были использованы формулы (19) и (20). Набор входных данных для кода (7)-(9) состоял из 13,312,678 псевдокодовых слов TS, из которых 2392 были выбраны с минимальным d_ε^2 с использованием Стратегии 2.

Вычисления вероятности ошибок для значений отношения сигнал-шум (ОСШ) Eb/N0 в диапазоне от 2.0 до 7.0 с шагом 0.2 были выполнены с использованием предложенного метода выборки по значимости, включающего табличные модификации. Полученные кривые, иллюстрирующие вероятность битовой ошибки (BER) и вероятности блочной ошибки (FER), представлены на рис. 2. На этом же рисунке представлены кривые BER и FER, полученные по методу Монте-Карло.

Представленные кривые получены при использовании декодера Quantize Normalize Min-Sum с 25 итерациями [35, 36], 4 бита применялись для представления мягкой метрики сообщения от узла проверки к вершине, байт - для мягкой метрики от вершины к узлам проверок.

Предложенный метод оценки в сравнении с предложенным ранее методом Коула [33], позволяет осуществить распараллеливание вычислений без необходимости дублирования таблиц. Такой подход кратно уменьшает объем требуемой памяти и позволяет осуществлять вычисления по разделенным индексам. Ускорение зависит от структуры проверочной матрицы и растет почти линейно с ростом размера циркулянта, в частности для предложенного кода достигает 5.3 раза при использование циркулянта размером 2048. Кроме этого разделение переменных и размер таблиц позволяет применять кэширование для дальнейшего ускорения вычислений.

Из рисунка видно, что предложенный метод обеспечивает высокую точность вычисления BER и FER (кривые оценок, полученные с использованием предложенного метода, практически совпадают с кривыми, полученными по методу Монте-Карло), с быстродействием значительно большим по сравнению с методом, предложенным в [33].

Предложенный метод позволяет оценивать помехоустойчивость кода на всем диапазоне ОСШ для размера циркулянта, не являющегося простым числом. Мультипликативное изменение циркулянта, рост длины квазициклического кода с циркулянтом z=z*2,3,4, …, слабо влияет на рост сложности поиска треппин-сетов и вычисления вероятности ошибки (взвешивание треппин-сетов).

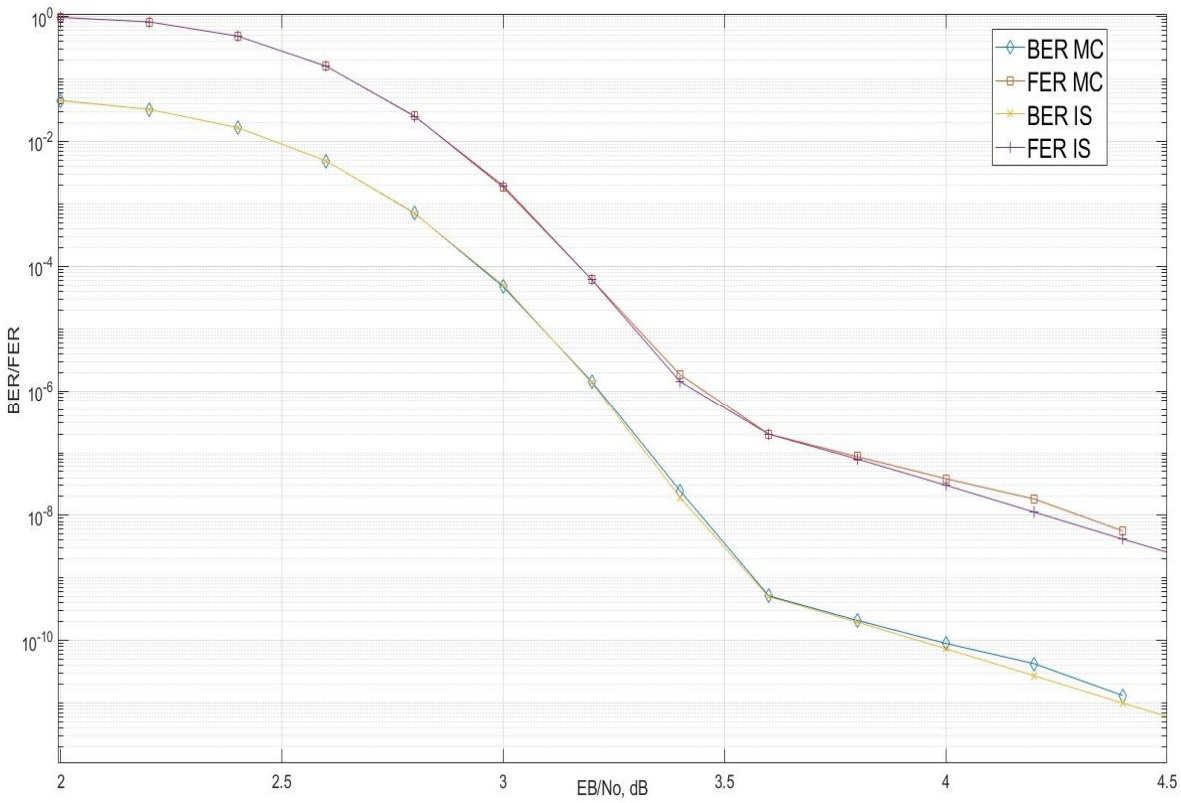

Рис. 2. Сравнение предложенного метода с методом Монте-Карло

Fig. 2. Comparison of the proposed method with the Monte Carlo method

Выводы

В данной статье представлены метод поиска треппин сетов и метод оценки вероятности ошибок, вызванных треппин-сетами, для квазициклических кодов с размером циркулянта, не являющимся простым числом.

Предложенный метод поиска треппин-сетов использует алгебраические свойства квазициклических кодов на графах. С использованием операций подъема и проекции графа задача поиска треппин-сетов переводится в пространство большей размерности, где треппин-сеты более различимы.

Метод обеспечивает поиск треппин-сетов в квазициклических кодах большой длины с высоким быстродействием. Так, время поиска треппин-сетов для кода с размером циркулянта 128 уменьшилось в 2.5 раза по сравнению с ранее предложенным методом [30]. С ростом размера циркулянта выигрыш по быстродействию увеличивается. Также предложенный метод удобен для аппаратной реализации на БИС и ПЛИС за счет уменьшения требований к объему ОЗУ.

Предложенный метод выборки по значимости в сравнении с ранее предложенным методом Коула, позволяет осуществить распараллеливание вычислений без необходимости дублирования таблиц. Такой подход кратно умень-

шает объем требуемой памяти и позволяет осуществлять вычисления по разделенным индексам.

Применение предложенного метода для оценки вероятности ошибок, вызванных треппин-сетами, обеспечивает ускорение в 5.3 раза в сравнении с методом Коула [33] для циркулянта размером 2048. Предложенный метод позволяет оценивать помехоустойчивость кода на всем диапазоне ОСШ.

Рост длины квазициклического кода слабо влияет на рост сложности поиска треппин-сетов и вычисления вероятности ошибок (взвешивания треппин-сетов), реализуемых предложенными методами.

Список литературы

1. Forney G. D. Codes on graphs: normal realizations // IEEE Transactions on Information Theory. 2001. Vol. 47, no. 2. P. 520-548.

2. Huang B., Jebara T. Approximating the permanent with belief propagation. URL: arxiv.org/abs/0908.1769. (accessed: 15.06.2024).

3. Vontobel P. O. A factor-graph approach to Lagrangian and Hamiltonian dynamics // IEEE International Symposium on Information Theory. 2011. P. 2183-2187.

4. Glasser I., Pancotti N., Cirac J. I. From Probabilistic Graphical Models to Generalized Tensor Networks for Supervised Learning // IEEE Access. 2020. Vol. 8. P. 68169-68182.

5. Ikeda S., Tanaka T., Amari S. Information geometry of turbo and low-density parity–check codes // IEEE Transactions on Information Theory. 2004. Vol. 50, no. 6. P. 1097-1114.

6. Usatyuk V., Sapozhnikov D., Egorov S. Topology-Aware Exploration of Energy-Based Models Equilibrium: Toric QC-LDPC Codes and Hyperbolic MET QC-LDPC Codes. URL: arxiv.org/abs/2401.14749. (accessed: 15.06.2024).

7. Welling M., Hinton G. E. A new learning algorithm for mean field boltzmann machines // Int. Conf. on Artificial Neural Networks, 2002. P. 351–357.

8. Fischer A., Igel Empirical analysis of the divergence of gibbs sampling based learning algorithms for restricted boltzmann machines // Int. Conf. on Artificial Neural Networks, 2010. P. 208–217.

9. Dinh L., Krueger D., Bengio Y. NICE: Non-linear independent components estimation. URL: arxiv.org/abs/1410.8516. (accessed: 15.06.2024).

10. Sohl-Dickstein J., et al. Deep unsupervised learning using nonequilibrium thermodynamics // Int. Conf. on Machine Learning, 2015. P. 2256–2265.

11. Gao R., et al. Learning generative convnets via multi-grid modeling and sampling // IEEE Conference on Computer Vision and Pattern Recognition, 2018. P. 9155–9164.

12. Bose A. J., Ling H., Cao Y. Adversarial contrastive estimation // The 56-th Annual Meeting of the Association for Comput. Linguistics, 2018. Vol. 1. P. 1021–1032.

13. Ceylan C., Gutmann M. U. Conditional noise-contrastive estimation of unnormalised models // International Conference on Machine Learning, 2018. P. 726–734.

14. Dai B., et al. Exponential family estimation via adversarial dynamics embedding // 33rd International Conference on Neural Information Processing Systems, 2019. P. 10979–10990.

15. Polyanskii N., Usatyuk V., Vorobyev I. Floor Scale Modulo Lifting for QC-LDPC codes. URL: arxiv.org/abs/1701.07521. (accessed: 15.06.2024).

16. Price A., Hall J. A Survey on Trapping Sets and Stopping Sets. URL: arxiv.org/abs/1705.05996. (accessed: 15.06.2024).

17. Myung S., Yang K. Extension of quasi-cyclic LDPC codes by lifting // IEEE International Symposium on Information Theory, 2005. P. 2305-2309.

18. Myung S., Yang K., Kim Y. Lifting methods for quasi-cyclic LDPC codes // IEEE Comm. Lett. 2006. Vol. 10, no. 6. P. 489-491.

19. Characterizations of pseudo-codewords of (low-density) parity-check codes / R. Koetter, W.-C. W. Li, P. O. Vontobel, J. L. Walker // Advances in Mathematics. 2007. Vol. 213, is. 1. P. 205-229.

20. On Deriving Good LDPC Convolutional Codes from QC LDPC Block Codes / A. E. Pusane, R. Smarandache, P. O. Vontobel, D. J. Costello // IEEE International Symposium on Information Theory, Nice, France, 2007. P. 1221-1225.

21. Deriving Good LDPC Convolutional Codes from LDPC Block Codes / A. E. Pusane, R. Smarandache, P. O. Vontobel, D. J. Costello // IEEE Transactions on Information Theory. 2011. Vol. 57, no. 2. P. 835-857.

22. LDPC block and convolutional codes based on circulant matrices / R. M. Tanner, D. Sridhara, A. Sridharan, T. E. Fuja, D. J. Costello // IEEE Transactions on Information Theory. 2004. Vol. 50, no. 12. P. 2966-2984.

23. MacKay D. J., Davey M. C. Evaluation of Gallager codes for short block length and high rate applications // The IMA Workshop on Codes, System and Graphical Models. 2001. P. 113–130.

24. Smarandache R., Vontobel P. O. Quasi-cyclic LDPC codes: Influence of proto- and Tanner-graph structure on minimum Hamming distance upper bounds // IEEE Transactions on Information Theory. 2012. Vol. 58, no. 2. P. 585-607.

25. Butler B. K., Siegel P. H. Bounds on the Minimum Distance of Punctured Quasi-Cyclic LDPC Codes // IEEE Transactions on Information Theory. 2013. Vol. 59, no. 7. P. 4584-4597.

26. Huang Y., Vontobel P. O. Bounding the Permanent of a Non-negative Matrix via its Degree- M Bethe and Sinkhorn Permanents // IEEE International Symposium on Information Theory. 2023. P. 2774-2779.

27. Smarandache R. Pseudocodewords from Bethe permanents // IEEE International Symposium on Information Theory. 2013. P. 2059-2063.

28. Vontobel P. O. Counting in Graph Covers: A Combinatorial Characterization of the Bethe Entropy Function // IEEE Transactions on Information Theory. 2013. Vol. 59, no. 9. P. 6018-6048.

29. Dehghan A., Banihashemi A. H. On the Tanner Graph Cycle Distribution of Random LDPC, Random Protograph-Based LDPC, and Random Quasi-Cyclic LDPC Code Ensembles // IEEE Transactions on Information Theory. 2018. Vol. 64, no. 6. P. 4438-4451.

30. Усатюк В.С., Егоров С.И. Построение LDPC-кодов с использованием модифицированного метода выборки по значимости Коула // Известия Юго-Западного государственного университета. 2023; 27(1): 92-110. https://doi.org/10.21869/2223-1560-2023-27-1-92-110.

31. XILINX SDAccel Development Environment Release Notes, Installation, and Licensing Guide UG1238 (v2019.1) July 26, 2019. P. 7-9. URL: https://download.amd.com/docnav/documents/aem/xilinx2019_1-ug1238-sdx-rnil.pdf. (accessed: 15.06.2024).

32. MacWilliams F., Sloane N. The Theory of Error-Correcting Codes. North-holland Publishing Company, 1977. 778 p.

33. Cole S. A., Wilson E. H., Giallorenzi T. A general method for finding low error rates of LDPC codes. URL: arxiv.org/abs/cs/0605051. (accessed: 15.06.2024).

34. Richardson T. Error floors of LDPC codes // The 41st Annu. Allerton Conf., Alleton, USA. 2003. P. 1426-1435.

35. Fossorier M., et al. Reduced complexity iterative decoding of low density parity check codes based on belief propagation // IEEE Transactions on Communications. 1999. Vol. 47, no. 5. P. 673-680.

36. Chen J., Fossorier M. Near optimum universal belief propagation based decoding of low-density parity check codes // IEEE Transactions on Communications. 2002. Vol. 50, no. 3. P. 406-414.

References

1. Forney G. D. Codes on graphs: normal realizations. *IEEE Transactions on Information Theory*. 2001; 47(2): 520-548.

2. Huang B., Jebara T. Approximating the permanent with belief propagation. Available at: arxiv.org/abs/0908.1769. (accessed: 15.06.2024).

3. Vontobel P. O. A factor-graph approach to Lagrangian and Hamiltonian dynamics. *IEEE International Symposium on Information Theory.* 2011. P. 2183-2187.

4. Glasser I., Pancotti N., Cirac J. I. From Probabilistic Graphical Models to Generalized Tensor Networks for Supervised Learning. *IEEE Access*. 2020; 8: 68169-68182.

5. Ikeda S., Tanaka T., Amari S. Information geometry of turbo and low-density parity–check codes. *IEEE Transactions on Information Theory*. 2004; 50(6): pp. 1097-1114.

6. Usatyuk V., Sapozhnikov D., Egorov S. Topology-Aware Exploration of Energy-Based Models Equilibrium: Toric QC-LDPC Codes and Hyperbolic MET QC-LDPC Codes. Available at: arxiv.org/abs/2401.14749. (accessed: 15.06.2024).

7. Welling M., Hinton G. E. A new learning algorithm for mean field boltzmann machines. *Int. Conf. on Artificial Neural Networks*, 2002. P. 351–357.

8. Fischer A., Igel Empirical analysis of the divergence of gibbs sampling based learning algorithms for restricted boltzmann machines. *Int. Conf. on Artificial Neural Networks*, 2010. P. 208–217.

9. Dinh L., Krueger D., Bengio Y. NICE: Non-linear independent components estimation. Available at: arxiv.org/abs/1410.8516. (accessed: 15.06.2024).

10. Sohl-Dickstein J., et al. Deep unsupervised learning using nonequilibrium thermodynamics. Int. Conf. on Machine Learning, 2015. P. 2256–2265.

11. Gao R., et al. Learning generative convnets via multi-grid modeling and sampling. *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. P. 9155–9164.

12. Bose A. J., Ling H., Cao Y. Adversarial contrastive estimation. *The 56-th Annual Meeting of the Association for Comput. Linguistics*. 2018; 1: 1021–1032.

13. Ceylan C., Gutmann M. U. Conditional noise-contrastive estimation of unnormalised models. *International Conference on Machine Learning*, 2018. P. 726–734.

14. Dai B., et al. Exponential family estimation via adversarial dynamics embedding. *33rd International Conference on Neural Information Processing Systems*, 2019. P.10979–10990.

15. Polyanskii N., Usatyuk V., Vorobyev I. Floor Scale Modulo Lifting for QC-LDPC codes. Available at: arxiv.org/abs/1701.07521. (accessed: 15.06.2024).

16. Price A., Hall J. A Survey on Trapping Sets and Stopping Sets. Available at: arxiv.org/abs/1705.05996. (accessed: 15.06.2024).

17. Myung S., Yang K. Extension of quasi-cyclic LDPC codes by lifting. *IEEE International Symposium on Information Theory*, 2005. P. 2305-2309.

18. Myung S., Yang K., Kim Y. Lifting methods for quasi-cyclic LDPC codes. *IEEE Comm. Lett.* 2006; 10(6): 489-491.

19. Koetter R., Li W.-C. W., Vontobel P. O., Walker J. L. Characterizations of pseudo-codewords of (low-density) parity-check codes. *Advances in Mathematics*. 2007; 213 (1): 205-229.

20. Pusane A. E., Smarandache R., Vontobel P. O., Costello D. J. On Deriving Good LDPC Convolutional Codes from QC LDPC Block Codes. *IEEE International Symposium on Information Theory*, Nice, France, 2007. P. 1221-1225.

21. Pusane A. E., Smarandache R., Vontobel P. O., Costello D. J. Deriving Good LDPC Convolutional Codes from LDPC Block Codes. *IEEE Transactions on Information Theory*. 2011; 57 (2): 835-857.

22. Tanner R. M., Sridhara D., Sridharan A., Fuja T. E., Costello D. J. LDPC block and convolutional codes based on circulant matrices. *IEEE Transactions on Information Theory*. 2004; 50 (12): 2966-2984.

23. MacKay D. J., Davey M. C. Evaluation of Gallager codes for short block length and high rate applications. The IMA Workshop on Codes, System and Graphical Models, 2001. P. 113–130.

24. Smarandache R., Vontobel P. O. Quasi-cyclic LDPC codes: Influence of proto- and Tanner-graph structure on minimum Hamming distance upper bounds. *IEEE Transactions on Information Theory*. 2012; 58(2): 585-607.

25. Butler B. K., Siegel P. H. Bounds on the Minimum Distance of Punctured Quasi-Cyclic LDPC Codes. *IEEE Transactions on Information Theory*. 2013; 59(7): 4584-4597.

26. Huang Y., Vontobel P. O. Bounding the Permanent of a Non-negative Matrix via its Degree- M Bethe and Sinkhorn Permanents. *IEEE International Symposium on Information Theory*, 2023/ P. 2774-2779.

27. Smarandache R. Pseudocodewords from Bethe permanents. *IEEE International Symposium on Information Theory*, 2013. P. 2059-2063.

28. Vontobel P. O. Counting in Graph Covers: A Combinatorial Characterization of the Bethe Entropy Function. *IEEE Transactions on Information Theory*. 2013; 59(9): 6018-6048.

29. Dehghan A., Banihashemi A. H. On the Tanner Graph Cycle Distribution of Random LDPC, Random Protograph-Based LDPC, and Random Quasi-Cyclic LDPC Code Ensembles. *IEEE Transactions on Information Theory*. 2018; 64(6): 4438-4451.

30. Usatjuk V. S., Egorov S. I. Construction of LDPC Codes Using Cole's Modified Importance Sampling Method. *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta = Proceedings of the Southwest State University*. 2023; 27(1): 92-110 (In Russ.). https:// doi.org/10.21869/2223-1560-2023-27-1-92-110.

31. XILINX SDAccel Development Environment Release Notes, Installation, and Licensing Guide UG1238 (v2019.1) July 26, 2019, p. 7-9. Available at: https:// download.amd.com/docnav/documents/aem/xilinx2019_1-ug1238-sdx-rnil.pdf. (accessed: 15.06.2024).

32. MacWilliams F., Sloane N. The Theory of Error-Correcting Codes. North-holland Publishing Company, 1977. 778 p.

33. Cole S. A., Wilson E. H., Giallorenzi T. A general method for finding low error rates of LDPC codes. Available at: arxiv.org/abs/cs/0605051. (accessed: 15.06.2024).

34. Richardson T. Error floors of LDPC codes. The 41st Annu. Allerton Conf., Alleton, USA, 2003. P. 1426-1435.

35. Fossorier M., et al. Reduced complexity iterative decoding of low density parity check codes based on belief propagation. *IEEE Transactions on Communications*. 1999; 47(5): 673-680.

36. Chen J., Fossorier M. Near optimum universal belief propagation based decoding of low-density parity check codes. *IEEE Transactions on Communications*. 2002. 50(3). 406-414.

Информация об авторах / Information about the Authors

Усатюк Василий Станиславович, кандидат технических наук, главный инженер департамента исследований и разработок, ООО «Т8», г. Москва, Российская Федерация, e-mail: usatiuk@t8.ru

Vasily S. Usatjuk, Cand. of Sci. (Engineering), Head Engineer, R&D department, Company T8, Moscow, Russian Federation, e-mail: usatiuk@t8.ru

Кузнецов Юрий Олегович, кандидат физико-математических наук, ведущий инженер-разработчик департамента исследований и разработок, ООО «Т8», г. Москва, Российская Федерация, e-mail: kuznetsov.y@t8.ru

Yuri O. Kuznetsov, Cand. of Sci. (Physico-Mathematical), Leading Engineer, Company T8, Moscow, Russian Federation, e-mail: kuznetsov.y@t8.ru

Егоров Сергей Иванович, доктор технических наук, доцент, профессор кафедры вычислительной техники, Юго-Западный государственный университет, г. Курск, Российская Федерация, e-mail: sie58@mail.ru, ORCID: https://orcid.org/0000-0001-5859-1024

Sergey I. Egorov, Dr. of Sci. (Engineering), Associate Professor, Southwest State University, Kursk, Russian Federation, e-mail: sie58 @mail.ru, ORCID: https://orcid.org/0000-0001-5859-1024