

Оригинальная статья / Original article

<https://doi.org/10.21869/2223-1560-2023-27-4-79-97>

Поиск треппин-сетов методом смешанного целочисленного линейного программирования с использованием априорного списка кодовых вершин

В.С. Усатюк ¹, С.И. Егоров ² ✉¹ ООО «Т8»

ул. Краснобогатyrская, 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 в кодах на графах, обеспечивающего полноту поиска.

Методы. Существует два подхода к поиску треппин-сетов. Первый на основе метода Монте-Карло со смещенной оценкой вероятности при помощи выборки по значимости (Importance Sampling) предусматривает использование декодера. Достоинством этого подхода является высокое быстродействие. Недостатками – зависимость от параметров декодера и характеристик канала и конечная вероятность пропуска треппин-сетов. Второй подход основан на применении методов линейного программирования. Достоинством этого подхода является полнота полученного списка треппин-сетов, обусловленная его независимостью от параметров декодера и характеристик канала. Недостаток подхода заключается в его большой вычислительной сложности. В статье в рамках второго подхода предложен новый метод поиска треппин-сетов с меньшей вычислительной сложностью. Метод предусматривает решение задачи смешанного целочисленного линейного программирования с использованием априорного списка кодовых вершин, участвующих в кратчайших (коротких) циклах в графе кода.

Результаты. С использованием предложенного метода был выполнен поиск треппин-сетов в нескольких низкоплотностных кодах. При этом применялся математический пакет линейного программирования IBM CPLEX версии 12.8, который запускался на 32 потоках 16-ядерного процессора AMD Ryzen 3950X с 32GB ОЗУ (DDR4). В коде Маргулиса (2640, 1320) с помощью предложенного метода был найден треппин-сет TS(6,6) за время 0.53 с. Ускорение, обеспечиваемое предложенным в статье методом, по сравнению с методом Веласкеса-Субрамани составляет 8252.415 раза. Благодаря высокому быстродействию и полноте поиска впервые были найдены треппин-сеты TS(62,16) и TS(52,14) в коде Маргулиса (4896, 2474).

Заключение. В статье предложен метод поиска треппин-сетов на основе смешанного целочисленного линейного программирования с использованием априорного списка кодовых вершин. Метод обладает высоким быстродействием и обеспечивает полноту поиска.

Ключевые слова: Коды на графах; LDPC-коды; треппин-сеты; методы смешанного целочисленного программирования; IBM CPLEX.

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

Для цитирования: Усатюк В.С., Егоров С.И. Поиск трэппин-сетов методом смешанного целочисленного линейного программирования с использованием априорного списка кодовых вершин // Известия Юго-Западного государственного университета. 2023; 27(4): 79-97. <https://doi.org/10.21869/2223-1560-2023-27-4-79-97>.

Поступила в редакцию 27.10.2023

Подписана в печать 06.12.2023

Опубликована 21.12.2023

Trapping Sets Search Using the Method of Mixed Integer Linear Programming with a Priori List of Variable Nodes

Vasily S. Usatjuk 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 trapping sets in graph codes, ensuring the completeness of the search.

Methods. There are two approaches to finding trapping sets. The first, based on the Monte Carlo method with a biased probability estimation using Importance Sampling, involves the use of a decoder. The advantage of this approach is its high performance. The disadvantages are the dependence on decoder parameters and channel characteristics and the finite probability of missing trapping sets. The second approach is based on the use of linear programming methods. The advantage of this approach is the completeness of the resulting list of trapping sets, due to its independence from the decoder parameters and channel characteristics. The disadvantage of this approach is its high computational complexity. In the article, within the framework of the second approach, a new method for searching trapping sets with less computational complexity is proposed. The method involves solving a mixed integer linear programming problem using an a priori list of code vertices participating in the shortest cycles in the code graph.

Results. Using the proposed method, a search for trapping sets was performed for several low-density codes. For this purpose, the mathematical linear programming package IBM CPLEX version 12.8 was used, which was run on 32 threads of a 16-core AMD Ryzen 3950X processor with 32GB of RAM (DDR4). In the Margulis code (2640, 1320), using the proposed method, the trapping set TS(6,6) was found in a time of 0.53 s. The speedup provided by the method proposed in the paper compared to the Velazquez-Subramani method is 8252.415 times. Thanks to the high speed and completeness of the search, trapping sets were found for the first time TS(62,16) and TS(52,14) in the Margulis code (4896, 2474).

Conclusion. The paper proposes a new method for searching trapping sets by solving a mixed integer linear programming problem with an a priori list of code. The method is fast and provides completeness of the search.

Keywords: graph codes; LDPC codes; trapping sets; mixed integer programming method; IBM CPLEX.

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., Egorov S. I. Trapping Sets Search Using the Method of Mixed Integer Linear Programming with a Priori List of Variable Nodes // *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta = Proceedings of the Southwest State University*. 2023; 27(4): 79-97 (In Russ.). <https://doi.org/10.21869/2223-1560-2023-27-4-79-97>.

Received 27.10.2023

Accepted 06.12.2023

Published 21.12.2023

Введение

Использование кодов на графах привело к бурному развитию систем: квантового кодирования источников [1, 2], квантовых вычислений [2, 3], хранения информации (на магнитных, твердотельных, оптических носителях), беспроводной, оптической, спутниковой и других видов связи [4-7].

Коды на графах, использующие мягкое декодирование для коррекции ошибок, обеспечивают существенный энергетический выигрыш (от 1,5 до 2,5 дБ в режиме ограничения мощности до более 4 дБ в режиме ограничения по полосе с учетом выигрыша за счет работы с модуляционным созвездием) при передаче (воспроизведении) сообщений через канал с шумом.

Мягкие метрики (вероятности или логарифмы правдоподобия) полученного сообщения используются при реализации различных методов *мягкого декодирования* с помощью проверочной матрицы кода. Наибольшую помехоустойчивость обеспечивают методы, реализующие *декодирование по максимальному правдоподобию*, позволяющие найти максимально правдоподобное кодовое слово. Однако применение этих методов на практике существенно ограничено в силу экспоненциального роста сложности декодирования при увеличении длины кода. Например, в беспроводной связи используются коды с длиной порядка несколько десятков

тысяч бит (2^{32000} операций декодера), оптическая связь может потребовать коды с длиной порядка несколько сотен тысяч бит (2^{320000} операций).

Решением проблемы экспоненциального роста сложности является применение метода *распространения доверия* (Message Passing, Belief Propagation) и широко класса его аппроксимаций. Успешному применению этих методов мешают *асимметричные структуры в кодовом графе*: стоппин-сетов (Stopping Set, [8]) для двоичного стирающего канала, треппин-сетов (Trapping Sets) для АБГШ-канала, треппин-сетов (Trapping Sets, [9]) для Рэлеевского канала с замираниями (Rayleigh flat fading, [10]). Эти структуры приводят к получению в результате декодирования псевдокодовых слов, вызывающих ошибки, которые не возникли бы в случае применения оптимальных экспоненциальных по сложности методов декодирования по максимальному правдоподобию.

Центральным аспектом улучшения кодов на графах, необходимым для обеспечения большей надежности вычислительных систем, систем хранения данных, систем машинного обучения и систем связи, является поиск асимметричных структур в кодовом графе с последующей их нейтрализацией.

Поиск таких графовых структур является NP-сложной задачей [11]. Например, для поиска стоппин-сета размера 6 в коде Маргулиса (2640, 1320) без априорного знания структуры цик-

лов, автоморфизмов, индекса симметричной группы и прочего, потребуется анализ $\binom{2640}{6} \approx 4.67 \times 10^{17}$ точек в пространстве поиска решения.

Существует два подхода к поиску треппин-сетов. Первый на основе метода Монте-Карло со смещенной оценкой вероятности при помощи выборки по значимости (Importance Sampling) предусматривает использование декодера. Этот подход был представлен в работах Ричардсона, Коула, Банихашеми [12-16]. Достоинством этого подхода является высокое быстродействие. Недостатками – зависимость от параметров декодера и характеристик канала и конечная вероятность пропуска треппин-сетов.

Второй подход основан на применении методов линейного программирования. Декодер при этом не используется. Этому подходу посвящены работы Роснеса и Веласкеса-Субрамани [8, 10, 17, 18]. Достоинством этого подхода является полнота полученного списка треппин-сетов, обусловленная его независимостью от параметров декодера и характеристик канала. Недостаток подхода заключается в его большой вычислительной сложности.

Статья посвящена разработке декодеро-независимого метода поиска ассиметричных структур подграфов треппин-сетов при помощи смешанного линейного программирования, обладающего на порядки меньшей вычислительной сложностью.

Материалы и методы

Коды на графах – это математические графические модели, используемые для представления факторизованных форм распределений вероятностей и управления ими.

Коды на графах и их математические модели особенно ценны в таких приложениях, как помехоустойчивое кодирование, кодирование источников, шифрование, байесовские сети, марковские случайные поля и скрытые марковские модели, где графическое представление зависимостей играет ключевую роль в понимании и решении прикладных задач.

Эти коды описываются фактор-графом $G=(VUC,E)$, состоящим из n вершин переменных V и $(n-k)$ факторных вершин C (в помехоустойчивом кодировании называемых соответственно кодовыми и проверочными вершинами), где k – размерность кода. Вершины графа соединяются ребрами E . Пример фактор-графа приведен на рис. 1.

Соединения вершин описываются матрицей H (называемой в помехоустойчивом кодировании проверочной). Строки матрицы задают уравнения, определенные над конечным полем:

$$\sum_{l=0}^{n-1} v_l \cdot h_{j,l} = 0, \quad (1)$$

где $h_{j,l}$ – элемент матрицы H ; j – номер ее строки, соответствующей факторной вершине; l – номер ее столбца, соответствующего переменной вершине. Переменная вершина соединяется ребром с факторной вершиной в том случае, если $h_{j,l} \neq 0$.

Для кодов на графах должны быть заданы факторы — функции, представляющие зависимости или взаимодействия между переменными. Факторные функции принимают подмножество узлов переменных в качестве входных данных и выдают действительное число

в качестве выходных данных. Ребра (E) в фактор-графе соединяют вершины переменных с вершинами факторов в зависимости от того, какие переменные являются аргументами для каких факторных функций.

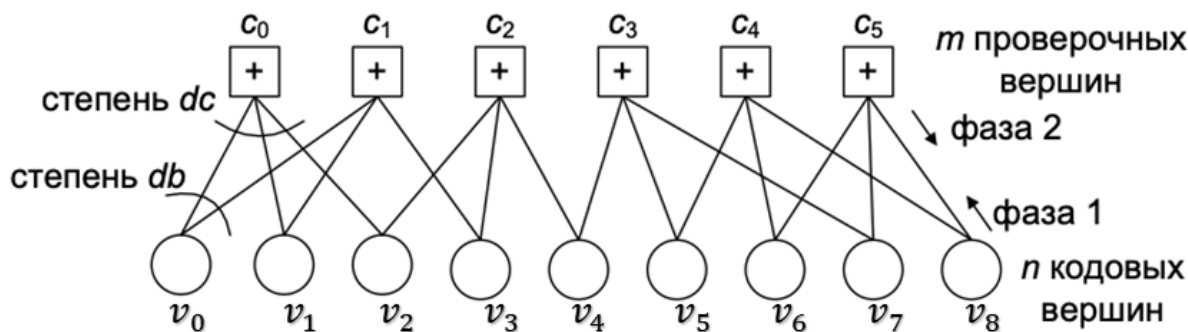


Рис. 1. Фактор-граф двоичного регулярного LDPC кода длины 9

Fig. 1. Factor-graph of binary regular LDPC code of length 9

Фактор-граф позволяет четко и кратко представить сложные вероятностные модели, упрощая выполнение логических выводов, оптимизацию, обучение, исправление ошибки в рамках некоторой модели (канала). Структура графа отражает отношения условной независимости между переменными в факторизованном распределении вероятностей, что имеет решающее значение для вероятностных рассуждений.

Особым классом кодов на графе, широко используемом в помехоустойчивом кодировании, являются помехоустойчивые линейные коды. Важным подмножеством этих кодов являются LDPC-коды (Low Density Parity Check Codes)). Низкая плотность отличных от нуля символов в проверочной матрице этих кодов позволяет осуществлять декоди-

рование с мягкими решениями с использованием метода распространения доверия (Belief Propagation, BP) с относительно небольшой сложностью. Метод BP обладает значительно большей корректирующей способностью по сравнению с методами декодирования с жесткими решениями.

Фактор-граф LDPC-кода G показан на рис.1. Он является графом Таннера с двумя множествами вершин. Одно состоит из $m=n-k$ факторных (проверочных) вершин $\{c_0, c_1, \dots, c_{m-1}\}$, соответствующих m строкам матрицы H , второе - из n переменных (кодовых) вершин $\{v_0, v_1, \dots, v_{n-1}\}$ соответствующих n столбцам матрицы H .

Эффективность декодирования методом BP уменьшается из-за наличия циклов в фактор-графе, образующих треппин-

сеты (Trapping set, TS). Треппин-сеты — это $TS(a, b)$ -подграфы, состоящие из a символьных (переменных) узлов, b из которых инцидентны проверочным узлам

с нечетными степенями [19]. Треппин-сеты обуславливают ошибку декодирования в методе ВР. Примеры треппин-сетов приведены на рис. 2.

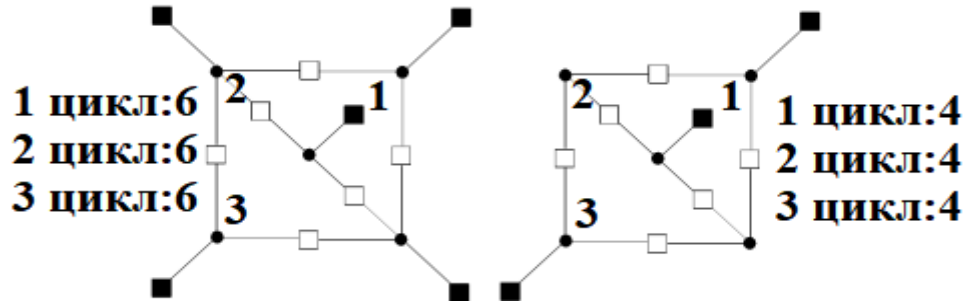


Рис. 2. Подграфы треппин-сетов $TS(5,5)$ (слева) и $TS(5,3)$ (справа): ● – символьный узел; □ – проверочный узел с четной степенью; ■ – проверочный узел с нечетной степенью

Fig. 2. Trapping set subgraphs: $TS(5,5)$ (left) and $TS(5,3)$ (right): ● – symbolic node; □ – check node with even degree; ■ – check node with odd degree

В работе [18] Веласкес и Субрама-ни предложили метод поиска треппин-сетов в фактор-графе $G=(VUC,E)$ на основе использования целочисленного линейного программирования.

Целочисленное линейное программирование позволяет найти экстремум целочисленной функции с учетом ограничений на переменные. Чаще всего для этого используется симплекс метод – алгоритм решения оптимизационной задачи линейного программирования путём перебора вершин выпуклого многогранника в многомерном пространстве.

В соответствии с методом [18] ищется минимум целевой функции:

$$\min \sum_{v_i \in V} x_i, \quad (2)$$

где x_i – битовые флаги кодовых вершин, образующих треппин-сеты, $x_i \in \{0,1\}$, $\forall i \in \{1,2,\dots,n\}$; при выполнении следующих двух условий.

Первое условие:

$$\sum_{c_j \in C} I_j^{\text{odd}} = b, \quad (3)$$

где I_j^{odd} – фиктивная переменная, равная 1, если символьный узел инцидентен нечетному числу проверок, равная 0 в противном случае, $\forall j \in \{1,2,\dots,m\}$, $b < b_{\max}$, и $b_{\max}$ задает максимальный параметр нечетной степени инцидентности $TS(a,b)$.

Второе условие:

$$2\alpha_j + I_j^{\text{odd}} - \sum_{(v_i, c_j) \in E} x_i = 0, \quad \forall j \in \{1,2,\dots,m\}, \quad (4)$$

где α_j – вспомогательная переменная для проверки условия степени инцидентности фиктивной переменной I_j^{odd} , $\alpha_k \in \{0, \dots, \lfloor \frac{\rho_{\max}}{2} \rfloor\}$, $\rho_{\max}$ – наибольшая степень инцидентности проверки в TS -подграфе (вес строки), $\forall k \in \{1,2,\dots,m\}$.

В результате решения целочисленной программы с $n+2m$ переменными и $m+1$ условиями находятся x_i (битовые

флаги кодовых вершин, образующих треппин-сеты).

В работе [18] Веласкес и Субрама-ни искали треппин-сеты для кода Маргулиса (2640, 1320) [26] и кода Маргулиса (4896, 2474) на графе Рамануджана [27]. При этом применялся математический пакет линейного программирования Gurobi platform версии 7.5.1, запускаемый на 32 потоках 64 ядерного процессора (Intel Xeon Phi Processor 7210, 64 GB DDR4). В частности, для кода Маргулиса (4896, 2474) за 700451 секунд был найден подграф с 48 кодовыми вершинами.

Недостатком метода является экспоненциальный рост сложности вычислений с ростом параметра a треппин-сета. Полный перебор требует анализ $\binom{n}{a}$ точек в пространстве поиска решений.

Нами предложена модификация метода поиска треппин-сетов с использованием линейного программирования, основанная на факте наличия кодовых вершин в кратчайших (коротких) циклах графа. Эта модификация, используя множество кодовых вершин $cycle_{vns}$, позволяет на ранних этапах работы линейного решателя – пресолвера, осуществить отсечения максимального числа невыполнимых условий.

Новый метод предназначен для поиска треппин-сетов $TS(a, b)$ с минимальным значением a , ($\min(a)_{cycle_{vns} \in TS(a, b)}$) путем использования смешанного целочисленного линейного программирования (mixed integer linear programming

problem). При этом конкретный вид целевой функции, ограничения переменных и неравенств определяются с учетом наличия априорных узлов. В проверочной матрице $H^{m \times n} \in GF(2)$ ищутся кодовые вершины, образующие $TS(a, b)$: $\min(a)_{TS(a, b) \in H}$ при наличии списка кодовых узлов $cycle_{vns}$, содержащихся в кратчайшем (коротком) цикле, образующем $TS(a, b)$.

Для поиска треппин-сетов используется следующая целевая функция:

$$\min(f^T * x), \quad (5)$$

где x_i – битовые флаги кодовых вершин, образующих треппин-сеты, $x_i \in \{0, 1\}$, $\forall i \in \{2*m+1, \dots, 2*m+n\}$, m – число проверочных вершин, $x_i \in \mathbb{Z}$, $\forall i \in \{1, \dots, 2*m\}$, $f = (0^{1 \times m} 1^{1 \times m} 0^{1 \times n})$.

Решения должны удовлетворять следующим условиям.

Первое условие:

Переменные решения x_i должны попадать в интервал:

$$lb \leq x \leq ub, \quad (6)$$

где $lb = 0^{(2m+n) \times 1}$, $ub(1:m) = \left\lfloor \frac{1}{2} \max \text{degree}_{CN} \right\rfloor$, $ub(m+1:2*m+n) = 1$, и $\max \text{degree}_{CN}$ – максимальный вес строки в проверочной матрице.

Второе условие:

$$Aeq * x = beq, \quad (7)$$

$$\text{где } Aeq = \begin{pmatrix} 2*I_m & I_m & -H \\ 0^{1 \times m} & 0^{1 \times m} & I^{1 \times m} \\ & Neq & \end{pmatrix}, \quad (8)$$

$$beq = \begin{pmatrix} 0^{m \times 1} \\ a \\ Ceq \end{pmatrix}, \quad (9)$$

и $\text{Ceq} = \text{card}(\text{cycle}_{vns})$, $\text{Neq}(2*m + \text{cycle}_{vns}) = 1$, где a – параметр треппин-сета $\text{TS}(a, b)$, card – число элементов в списке, Neq – маска позиций априорных кодовых вершин.

Предлагаемый метод поиска треппин-сетов методом смешанного целочисленного линейного программирования с априорным списком кодовых вершин можно представить в виде следующего алгоритма:

Вход: Проверочная матрица LDPC-кода $H^{m \times n}$, список кодовых вершин, содержащихся в кратчайшем(коротком) цикле cycle_{vns}

Выход: Список узлов треппин-сета $\text{TS}_{\text{list}} = \{\text{TS}_1(a, b): v_1, v_2, \dots, v_a\}$, содержащий параметры треппин-сета (a, b) и список символьных узлов, его образующих.

Формируем вектор-функцию цели $f = (0^{1 \times m} \ 1^{1 \times m} \ 0^{1 \times n})$.

Формируем матрицу линейной задачи, $\text{Neq}(2*m + \text{cycle}_{vns}) = 1$

$$A_{eq} = \begin{pmatrix} 2*I_m & I_m & -H \\ 0^{1 \times m} & 0^{1 \times m} & I^{1 \times m} \\ \text{Neq} & & \end{pmatrix}.$$

Формируем правую часть системы

$$b_{eq} = \begin{pmatrix} 0^{m \times 1} \\ a \\ \text{Ceq} \end{pmatrix}.$$

Формируем граничные условия для решения

$$lb \leq x \leq ub.$$

Вызываем функцию `cplexmilp` ($f, A_{eq}, b_{eq}, lb, ub$) для решения задачи смешанного линейного программирования, за-

данного уравнением $A_{eq} * x = b_{eq}$ на отрезке $lb \leq x \leq ub$.

Получаем искомое решение – вектор x , определяющий кодовые вершины треппин-сета $v_1, v_2, \dots, v_a$.

Согласно определению треппин-сета, находим кодовые вершины нечетной инцидентности, $\text{TS}(a, b)$.

Возвращаем $\text{TS}(a, b)$.

Результаты и их обсуждение

Для сравнения с методом Веласкеса-Субрамани был выполнен поиск треппин-сетов в кодах Маргулиса (2640, 1320) [26] и (4896, 2474) [27] с использованием предложенного метода. При этом применялся математический пакет линейного программирования IBM CPLEX версии 12.8, который запускался на 32 потоках 16 ядерного процессора AMD Ryzen 3950X с 32GB ОЗУ (DDR4).

Для кода Маргулиса (2640, 1320) была выбрана пара кодовых вершин в графе $\text{cycle}_{vns} = \{1, 1321\}$. При этом использовался метод поиска кратчайшего цикла в графе путем возведения в степень матрицы инцидентности. После выбора решалась задача смешанного линейного программирования для поиска минимального треппин-сета $\text{TS}(a, b)$, $\min(a)_{\text{cycle}_{vns} \in \text{TS}(a, b)}$.

В результате был найден треппин-сет $\text{TS}(6, 6)$, состоящий из кодовых вершин $x_i = \{1, 250, 901, 1321, 1883, 2454\}$ и проверочных вершин $c_i = \{5, 8, 253, 283, 326, 467, 497, 956, 1023, 1100, 1137, 1167\}$, заданный следующей проверочной подматрицей $H_{\text{TS}(6, 6)}$.

$$H_{TS(6,6)} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

Графическое представление треппин-сета TS(6,6) приведено на рис. 3.

Время поиска этого треппин-сета составило 0.53 секунды (252.39 ticks). Время решения аналогичной задачи по методу Веласкеса-Субрамани потребовало 4373.78 секунды ([18, табл. 2]) при использовании более быстродействующего пакета линейного программирования (Gurobi platform версии 7.5.1). Ускорение, обеспечиваемое предложенным в статье методом, по сравнению с методом Веласкеса-Субрамани составляет 8252.415 раза.

В работе [18] для кода Маргулиса (4896, 2474) на графе Рамануджана [27] был найден подграф стоппин-сета размера 48 за 700451 секунд. Нами были найдены подграфы размером 48 за 432 секунды и 52 за 459 секунд, соответственно. Ускорение по сравнению с методом Веласкеса-Субрамани составляет более 1621 раза.

Подграф TS(52,14) размера 52 состоит из кодовых вершин $x_i = \{32, 125, 135, 140, 329, 394, 556, 580, 745, 928, 969, 981, 1015, 1047, 1051, 1128, 1130,$

1141, 1295, 1345, 1396, 1491, 2005, 2199, 2276, 2366, 2556, 2700, 2823, 2824, 2884, 2921, 3087, 3094, 3140, 3149, 3269, 3349, 3601, 3707, 3744, 4103, 4164, 4189, 4473, 4489, 4499, 4696, 4727, 4837, 4867, 4888} и проверочных вершин, $c_i = \{31, 92, 149, 182, 189, 195, 196, 214, 241, 343, 354, 359, 360, 384, 386, 407, 446, 525, 541, 566, 575, 641, 734, 737, 767, 798, 859, 880, 887, 925, 943, 981, 1001, 1052, 1069, 1080, 1095, 1122, 1165, 1174, 1189, 1196, 1264, 1314, 1328, 1332, 1352, 1423, 1432, 1445, 1452, 1472, 1485, 1487, 1512, 1545, 1551, 1556, 1586, 1629, 1647, 1679, 1719, 1752, 1851, 1873, 1875, 1911, 1918, 2000, 2022, 2052, 2059, 2089, 2120, 2128, 2151, 2192, 2304, 2320, 2330, 2410, 2412, 2415, 2447\}$, (рис. 4).

Замечание. Структура треппин-сета и выбор априорных узлов влияют на эффективность CPLEX решателя, позволяя ускорять поиск за счет более эффективных методов решения задачи смешанного целочисленного линейного программирования функции `cxplexmilp`, [28]. Полный перебор для поиска треппин-сета TS(62, b) в коде Маргулиса (4896, 2474) требует анализа $\binom{4896}{62}$ точек в пространстве поиска решения.

Однако при определенных параметрах целочисленного программирования метод отсечения (метод Гомори) [29], релаксация задачи путем смешанных целочисленных округлений (обобщений неравенств целочисленного округления Хватала) [30] и другие методы, реализованные в пакете CPLEX, существенно ускоряют поиск треппин-сетов.

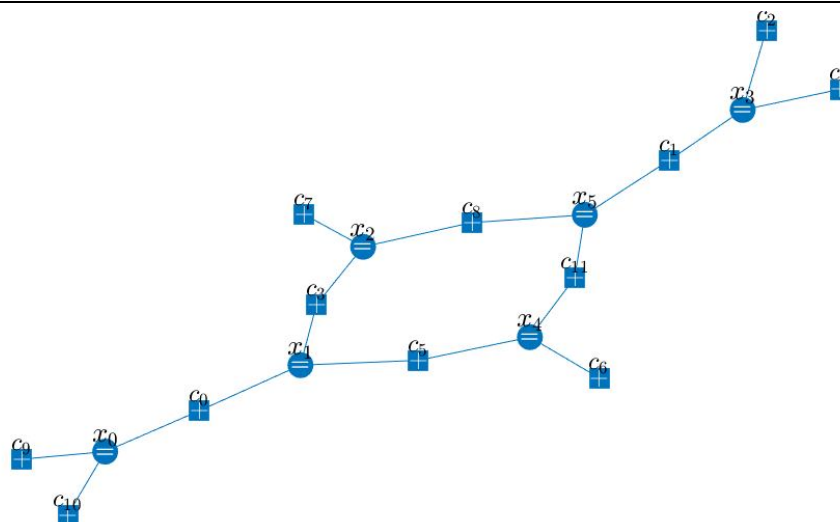


Рис. 3. Графическое представление $TS(6,6)$ в коде Маргулиса (2640, 1320)

Fig. 3. Graph representation of $TS(6,6)$ in Margulis code (2640, 1320)

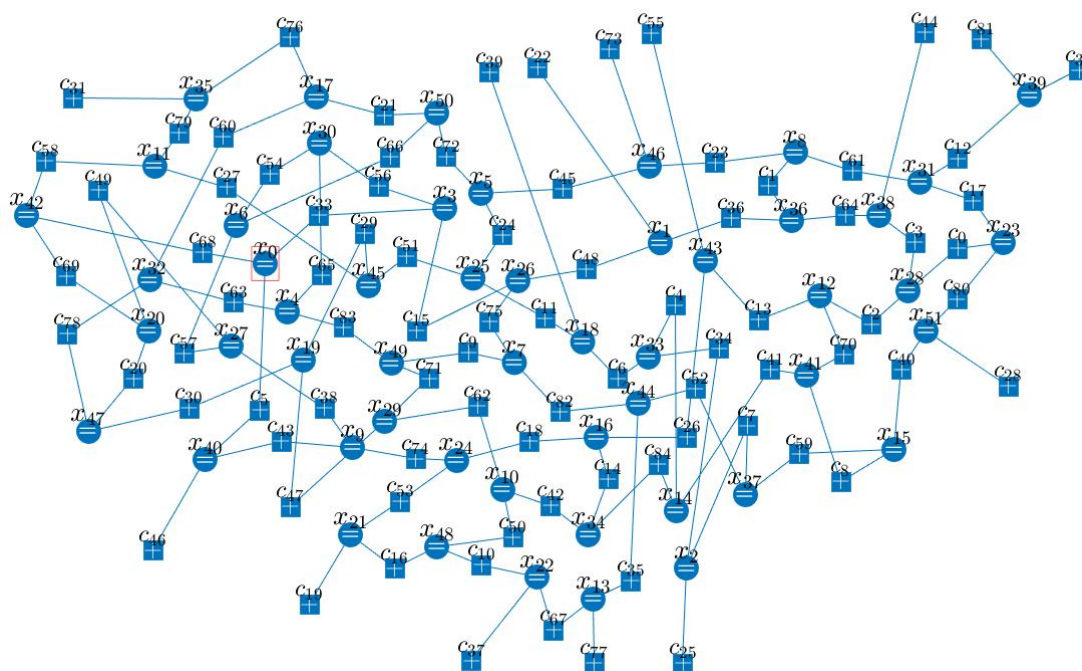


Рис. 4. Графическое представление $TS(52,14)$ в коде Маргулиса (4896, 2474)

Fig. 4. Graph representation of $TS(52,14)$ in Margulis code (4896, 2474)

Например, в случае поиска треппинг-сети в коде Маргулиса (4896, 2474) удачный выбор узлов в цикле длины 12, $cycle_{vns} = \{6, 41, 1132, 1140, 1508, 1561, 1988, 2506, 3154, 3182, 3444, 3575,$

$4141, 4710\}$ с последующим расширением списка до $cycle_{vns} = \{6, 41, 65, 159, 364, 719, 769, 880, 970, 981, 1097, 1132, 1140, 1164, 1274, 1279, 1385, 1508, 1561, 1625, 1681, 1716, 1823, 1979, 1988,$

2194, 2216, 2369, 2460, 2506, 2738, 2765, 2795, 2855, 2950, 2976, 3024, 3154, 3182, 3271, 3444, 3566, 3575, 3836, 3940, 4012, 4109, 4141, 4162, 4355, 4396, 4401, 4463, 4547, 4581, 4671, 4710, 4800, 4859, 4864} позволил выполнить поиск треппин-сета $TS(62,16)$, состоящего из кодовых вершин $x_i = \{6, 41, 65, 159, 364, 546, 719, 769, 880, 970, 981, 1097, 1132, 1140, 1164, 1274, 1279, 1385, 1508, 1561, 1625, 1681, 1716, 1819, 1823, 1979, 1988, 2194, 2216, 2369, 2460, 2506, 2738, 2765, 2795, 2855, 2950, 2976, 3024, 3154, 3182, 3271, 3444, 3566, 3575, 3836, 3940, 4012, 4109, 4141, 4162, 4355, 4396, 4401, 4463, 4547, 4581, 4671, 4710, 4800, 4859, 4864\}$ и проверочных вершин $c_i = \{8, 15, 23, 53, 86, 113, 122, 123, 132, 133, 204, 225, 234, 261, 377, 460, 486, 514,$

540, 541, 559, 587, 706, 723, 744, 751, 756, 774, 777, 814, 849, 874, 876, 880, 882, 934, 935, 963, 995, 1009, 1073, 1083, 1098, 1106, 1119, 1133, 1148, 1225, 1233, 1244, 1269, 1342, 1348, 1377, 1390, 1402, 1465, 1502, 1525, 1536, 1562, 1586, 1589, 1625, 1646, 1658, 1666, 1670, 1708, 1746, 1763, 1764, 1766, 1774, 1801, 1806, 1807, 1832, 1893, 1894, 1896, 1908, 1912, 1942, 1959, 1967, 1975, 1994, 2003, 2013, 2078, 2090, 2190, 2242, 2271, 2281, 2320, 2336, 2339, 2370\} (рис. 5) за 3.67 секунды.

Для сравнения с модифицированным методом выборки по значимости Коула [16] с помощью предложенного метода был выполнен поиск кратчайшего треппин-сета в нерегулярном квази-циклическом LDPC-коде (2560, 2048) (коде Б) с обхватом 6, EMD 5.

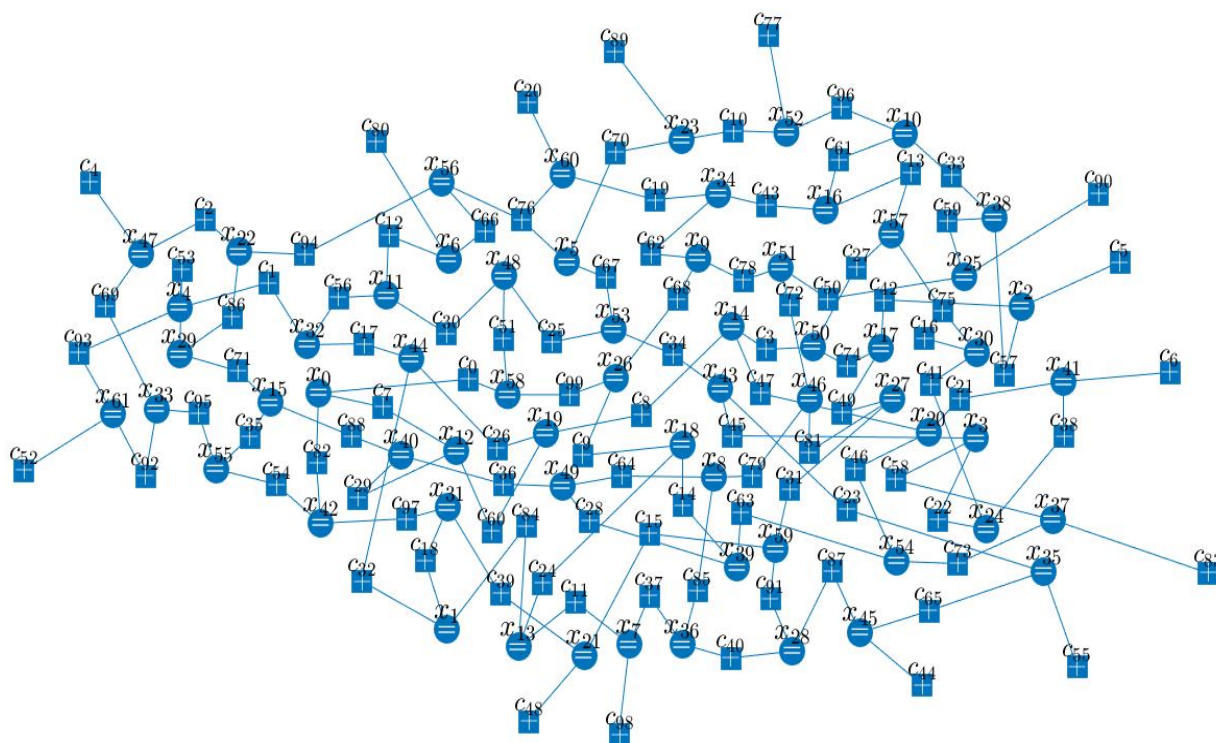


Рис. 5. Графическое представление $TS(62,16)$ в коде Маргулиса (4896, 2474)

Fig. 5. Graph representation of $TS(62,16)$ in Margulis code (4896, 2474)

С использованием 6 априорных кодовых вершин, содержащихся в кратчайшем цикле, менее чем за 0.3 секунды был найден $TS(20, 2)$, состоящий из кодовых вершин $x_i = \{257, 748, 790, 909, 946, 993, 1149, 1168, 1193, 1284, 1638, 1683, 1827, 2048, 2227, 2232, 2242, 2279, 2346, 2545\}$ и проверочных вершин $c_i = \{10, 17, 31, 61, 87, 93, 98, 102, 108, 127, 130, 140, 150, 168, 177, 223, 224, 242, 253, 276, 278, 311, 318, 320, 357, 375, 381, 387, 421, 435, 440, 450, 465, 474, 487, 490, 502\}$ (рис. 6), заданных следующей проверочной подматрицей $H_{TS(20,2)}$.

Треппин-сет $TS(20, 2)$ не был найден в работе [16] по причине большого обобщенного веса Хэмминга для примененных параметров поиска [16]. Это показывает большую полноту поиска предложенного метода, вытекающую из его декодеро-независимости, по сравнению с методами выборки по значимости.

Результаты применения предложенного метода для некоторых LDPC-кодов в сравнении с методом Веласкеса-Субрамани приведены в табл. 1.

Детальные результаты работы предложенного метода поиска треппин-сетов, отчеты результатов предварительной оптимизации решателя CPLEX, подматрицы и графы приведены в [31].

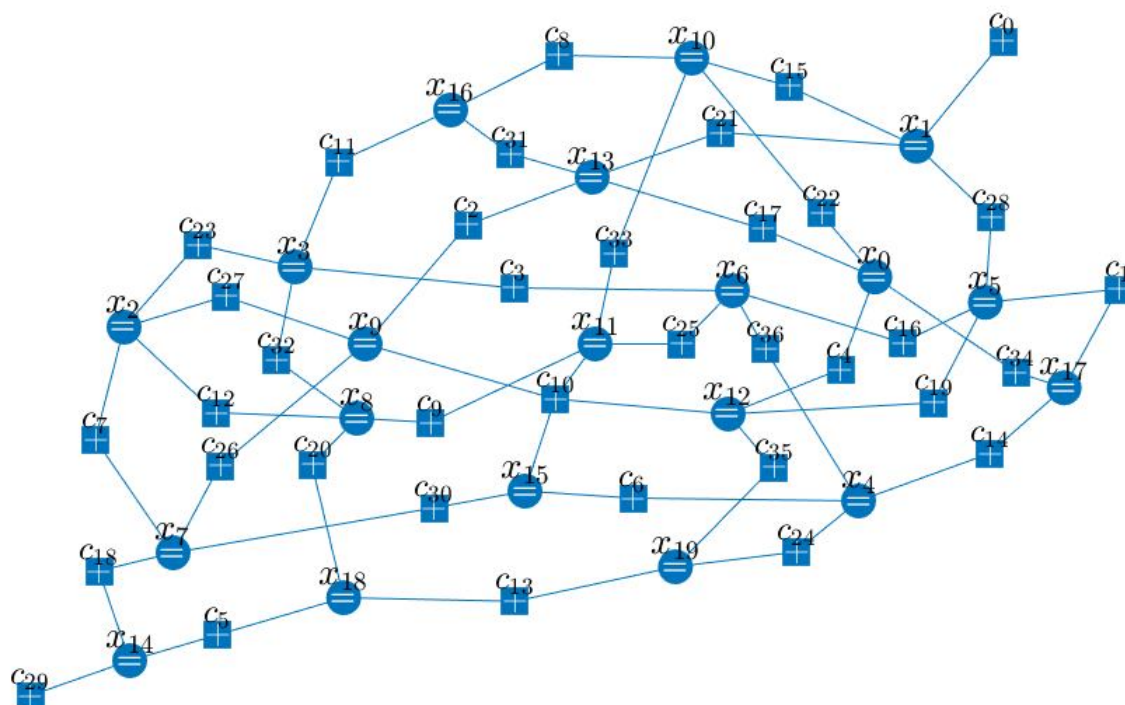


Рис. 6. Графическое представление $TS(20, 2)$ в QC-LDPC кода (2560, 2048)

Fig. 6. Graph representation of $TS(20, 2)$ in QC-LDPC code (2560, 2048)

[illegible]

Таблица 1. Время, необходимое для поиска треппин-сетов TS(a, b) в соответствующих LDPC-кодах, в секундах**Table 1.** Trapping set search time for different LDPC-codes, sec

TS(a, b)	LDPC-код Маккея PEG(1008, 504)		LDPC-код Маргулиса (2640, 1320)	
	ВС метод, [18]	Предложенный метод	ВС метод, [18]	Предложенный метод
a=5	53.42	5.22	520.95	6.83
a=6	31.74	2.06	4373.78	0.53
a=7	984.51	1.34	716.04	2.14
a=8	4.10	1.08	18843.19	0.22
a=9	4864.37	1.27	13504.13	0.20
a=10	8143.11	1.34		0.92
a=11	485919.5	4.74		1.21
a=12	245.1	3.63		4.02
...				
a=18		5.92		3.86

Выводы

В статье предложен метод поиска треппин-сетов на основе смешанного целочисленного линейного программирования с использованием априорного списка кодовых вершин. Метод обладает высоким быстродействием и обеспечивает полноту поиска.

В коде Маргулиса (2640, 1320) с помощью предложенного метода был найден треппин-сет TS(6,6) за время 0.53 с. Ускорение, обеспечиваемое предложенным в статье методом, по сравнению

с методом Веласкеса-Субрамани составляет 8252.415 раза.

Ключевое отличие предложенного метода поиска треппин-сетов по сравнению с методом выборки по значимости Коула заключается в полноте получаемого множества треппин-сетов, что обусловлено независимостью метода линейного программирования от параметров декодера.

Благодаря высокому быстродействию и полноте поиска впервые были найдены треппин-сеты TS(62,16) и TS(52,14) в коде Маргулиса (4896, 2474).

Список литературы

1. Djordjevic I. B. Quantum Communication, Quantum Networks, and Quantum Sensing / Elsevier/Academic Press, 2022. 608 p.
2. Milicevic M., Feng C., Zhang L.M. et al. Quasi-cyclic multi-edge LDPC codes for long-distance quantum cryptography // Quantum Inf. 2018. Vol.4. 21 p.

3. Panteleev P., Kalachev G. Degenerate Quantum LDPC Codes with Good Finite Length Performance // *Quantum*. 2021. Vol 5, p. 585.
4. Mézard M., Montanari A. Information, Physics, and Computation. Oxford Graduate Texts / Oxford University Press. 2009. 569 p.
5. Richardson T., Urbanke R. Modern Coding Theory / Cambridge University Press. 2008. 590 p.
6. Ryan W., Shu Lin Channel Codes: Classical and Modern / Cambridge University Press. 2009. 710 p.
7. Djordjevic I. B. Advanced Optical and Wireless Communications Systems, 2nd Edition / Springer Nature Switzerland AG. 2022. 992 p.
8. Rosnes E., Ytrehus O. An Algorithm to Find All Small-Size Stopping Sets of Low-Density Parity-Check Matrices // *IEEE International Symposium on Information Theory, Nice, France*. 2007. Pp. 2936-2940.
9. Richardson T. J. Error floors of LDPC codes// in 41st Annual Allerton Conference on Comm., Control and Computing, Oct. 2003. P. 1426–1435.
10. Rosnes E. "On the Effects of Pseudo-Codewords on Independent Rayleigh Flat-Fading Channels," *IEEE Inform. Theory Workshop on Information Theory for Wireless Networks, Bergen, Norway*. 2007. Pp. 1-5.
11. Velasquez A., Subramani K., Wojciechowski P. On the complexity of and solutions to the minimum stopping and trapping set problems // *Theor. Comput. Sci*. 2022. Vol. 915. Pp. 26-44.
12. Cole C. A. Error floor analysis for an ensemble of easily implementable irregular (2048, 1024) LDPC codes// *MILCOM - 2008*, pp. 1-5.
13. 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. (date of treatment: 14.04.2022).
14. Butler B. K., Siegel P. H. Error Floor Approximation for LDPC Codes in the AWGN Channel // *ITIT*. 2014. Vol. 60. № 12. Pp. 7416-7441.
15. Karimi M., Banihashemi A. H. Efficient Algorithm for Finding Dominant Trapping Sets of LDPC Codes // *IEEE Transactions on Information Theory*, 2012. Vol. 58. № 11. Pp. 6942-6958.
16. 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>.
17. Sariduman A., Pusane A.E., Taşkın Z.C. An integer programming based trapping set search technique // 2012 20th Signal Processing and Communications Applications Conference, Mugla, Turkey. 2012. Pp. 1-4.
18. Velasquez A., et al. Finding Minimum Stopping and Trapping Sets: An Integer Linear Programming Approach / In: Lee J., Rinaldi G., Mahjoub A. (eds) *Comb. Optim. ISCO 2018. Lect. Notes in Comp. Science*, V. 10856. Pp. 402–415.

19. Vasić B., et al Trapping set ontology // 47th Annual Allerton Conference on Communication, Control, and Computing (Allerton). 2009. Pp. 1-7.
20. Improved min-sum decoding algorithms for irregular LDPC codes / J. Chen, R. M. Tanner, C. Jones, Y. Li // ISIT., 2005. Pp. 449-453.
21. Two-dimensional correction for min-sum decoding of irregular LDPC codes / J. Zhang, M. Fossorier, D. Gu, J. Zhang // in IEEE Communications Letters. March 2006. Vol. 10. № 3. Pp. 180-182.
22. Zhang S., Schlegel C. Controlling the Error Floor in LDPC Decoding // in IEEE Transactions on Comm. 2013. Vol. 61(9). Pp. 3566-3575.
23. Tian T., et al Selective avoidance of cycles in irregular LDPC code construction // IEEE Trans. on Comm. 2004. Vol. 52. № 8. Pp. 1242-1247.
24. Усатюк В.С., Егоров С.И. Построение квазициклических недвоичных низкоплотностных кодов на основе совместной оценки их дистантных свойств и спектров связности //Телекоммуникации. 2016. №8. С. 32-40.
25. Усатюк В.С., Егоров С.И. Устройство для оценки кодового расстояния линейного блочного кода методом геометрии чисел // Известия Юго-Западного государственного университета. Серия: Управление, вычислительная техника, информатика. Медицинское приборостроение. 2017. № 4 (25). С. 24-33.
26. Margulis G. A. Explicit Constructions of Graphs Without Short Cycles and Low Density Codes // Combinatorica. 1982. Vol. 2. № 1. Pp. 71-78.
27. Rosenthal J., Vontobel P. O. Constructions of regular and irregular LDPC codes using Ramanujan graphs and ideas from Margulis // Proceedings 2001 IEEE International Symposium on Information Theory. 2001. Pp. 4.
28. IBM CPLEX function documentation. URL: <https://www.ibm.com/docs/en/icos/12.9.0?topic=functions-cplexmip>
29. Gilmore P. C., Gomory R. E. A linear programming approach to the cutting stock problem". Operations Research. 9 (6), (1961): 849–859.
30. Chvatal V. Edmonds polytopes and a hierarchy of combinatorial problems // Discrete Math. 1973. № 4. Pp. 305–337.
31. Usatyuk V.S. Support material for Mixed Linear Programming Method for Trapping Sets Search. URL: <https://github.com/Lcrypto/trapping-sets-enumeration/tree/master/LP>

References

1. Djordjevic I. B. Quantum Communication, Quantum Networks, and Quantum Sensing. / Elsevier/Academic Press, 2022, 608 p.
2. Milicevic, M., Feng, C., Zhang, L.M. et al. Quasi-cyclic multi-edge LDPC codes for long-distance quantum cryptography. *Quantum Inf*, 2018, vol.4, 21 p.
3. Panteleev P., Kalachev G. Degenerate Quantum LDPC Codes with Good Finite Length Performance. *Quantum*, 2021, vol. 5, 585 p.

4. Mézard M., Montanari A. *Information, Physics, and Computation*. Oxford Graduate Texts / Oxford University Press, 2009, 569 p.
5. Richardson T., Urbanke R. *Modern Coding Theory*. Cambridge University Press. 2008, 590 p.
6. Ryan W., Shu Lin. *Channel Codes: Classical and Modern*. Cambridge University Press, 2009, 710 p.
7. Djordjevic I. B. *Advanced Optical and Wireless Communications Systems*, 2nd Edition. Springer Nature Switzerland AG, 2022, 992 p.
8. Rosnes E., Ytrehus O. An Algorithm to Find All Small-Size Stopping Sets of Low-Density Parity-Check Matrices. *IEEE International Symposium on Information Theory*, Nice, France, 2007, pp. 2936-2940.
9. Richardson T. J. Error floors of LDPC codes. *In 41st Annual Allerton Conference on Comm., Control and Computing*, Oct. 2003, pp. 1426–1435.
10. Rosnes E. "On the Effects of Pseudo-Codewords on Independent Rayleigh Flat-Fading Channels," *IEEE Inform. Theory Workshop on Information Theory for Wireless Networks*, Bergen, Norway, 2007, pp. 1-5.
11. Velasquez A., Subramani K., Wojciechowski P. On the complexity of and solutions to the minimum stopping and trapping set problems. *Theor. Comput. Sci.*, 2022, vol. 915, pp. 26-44.
12. Cole C. A. Error floor analysis for an ensemble of easily implementable irregular (2048, 1024) LDPC codes. *MILCOM - 2008*, pp. 1-5.
13. 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: 14.04.2022).
14. Butler B. K., Siegel P. H. Error Floor Approximation for LDPC Codes in the AWGN Channel. *ITIT*, 2014, vol. 60, no. 12, pp. 7416-7441.
15. Karimi M., Banihashemi A. H. Efficient Algorithm for Finding Dominant Trapping Sets of LDPC Codes. *IEEE Transactions on Information Theory*, 2012, vol. 58, no. 11, pp. 6942-6958.
16. 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>.
17. Sarıduman A., Pusane A.E., Taşkın Z.C. An integer programming based trapping set search technique. *2012 20th Signal Processing and Communications Applications Conference*, Mugla, Turkey, 2012, pp. 1-4.
18. Velasquez A., et al. Finding Minimum Stopping and Trapping Sets: An Integer Linear Programming Approach; In: Lee J., Rinaldi G., Mahjoub A. (eds) *Comb. Optim. ISCO Lect. Notes in Comp. Science*, 2018. V. 10856., pp. 402–415.

19. Vasić B., et al Trapping set ontology. *47th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2009, pp. 1-7.
20. Chen J., Tanner R. M., Jones C., Li Y. Improved min-sum decoding algorithms for irregular LDPC codes. *ISIT*, 2005, pp. 449-453.
21. Zhang J., Fossorier M., Gu D., Zhang J. Two-dimensional correction for min-sum decoding of irregular LDPC codes. in *IEEE Communications Letters*, March 2006, vol. 10, no. 3, pp. 180-182.
22. Zhang S., Schlegel C. Controlling the Error Floor in LDPC Decoding. In *IEEE Transactions on Comm.*, 2013, vol. 61(9), pp. 3566-3575.
23. Tian T., et al Selective avoidance of cycles in irregular LDPC code construction. *IEEE Trans. on Comm.*, 2004, vol. 52, no. 8, pp. 1242-1247.
24. Usatjuk V. S., Egorov S. I. Postroenie kvazitsiklicheskikh nedvoichnykh nizkoplotnostnykh kodov na osnove sovmestnoi otsenki ikh distantnykh svoistv i spektrov svyaznosti [Quasi-Cyclic Non-binary LDPC Codes Construction based on joint optimization the code distance properties and ACE=EMD Spectrum code's graph properties]. *Telekommunikatsii = Telecommunications*, 2016, no 8, pp. 32-40.
25. Usatjuk V. S., Egorov S. I. [An Apparatus for Estimation of Linear Block Codes Minimum Distance Based on Number Geometry Method]. *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta. Seriya: Upravlenie, vychislitel'naya tekhnika, informatika. Meditsinskoe priborostroenie = Proceedings of the Southwest State University. Series: Control, Computing Engineering, Information Science. Medical Instruments Engineering*, 2017, no 4 (25), pp. 24-33 (In Russ.).
26. Margulis G. A. Explicit Constructions of Graphs Without Short Cycles and Low Density Codes. *Combinatorica*, 1982, vol. 2, no. 1, pp. 71-78.
27. Rosenthal J., Vontobel P. O. Constructions of regular and irregular LDPC codes using Ramanujan graphs and ideas from Margulis. *Proceedings 2001 IEEE International Symposium on Information Theory*, 2001, pp. 4.
28. IBM CPLEX function documentation <https://www.ibm.com/docs/en/icos/12.9.0?topic=functions-cplexmilp>
29. Gilmore P. C., Gomory R. E. A linear programming approach to the cutting stock problem". *Operations Research*. 1961. 9 (6): 849–859.
30. Chvatal V. Edmonds polytopes and a hierarchy of combinatorial problems. *Discrete Math*, 1973, no. 4, pp. 305–337.
31. Usatyuk V.S. Support material for Mixed Linear Programming Method for Trapping Sets Search. Available at: <https://github.com/Lcrypto/trapping-sets-enumeration/tree/master/LP>

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

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>