НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
январь–февраль 2024 Том 24 № 1 http://ntv.ifmo.ru/
SCIENTIFIC AND TECHNICAL JOURNAL OF INFORMATION TECHNOLOGIES, MECHANICS AND OPTICS
January–February 2024 Vol. 24 No 1 http://ntv.ifmo.ru/en/
ISSN 2226-1494 (print) ISSN 2500-0373 (online)

doi: 10.17586/2226-1494-2024-24-1-70-80
УДК 004.056.4

Исправление одиночных пакетов ошибок за пределами корректирующей способности кода с использованием информационных совокупностей

Мария Николаевна Исаева[1]✉, Андрей Анатольевич Овчинников[2]

[1,2] Национальный исследовательский университет «Высшая школа экономики», Санкт-Петербург, 190008, Российская Федерация

[1] misaeva@hse.ru✉, https://orcid.org/0009-0007-6228-0617
[2] a.ovchinnikov@hse.ru, https://orcid.org/0000-0002-8523-9429

Аннотация

Введение. Исправление ошибок, возникающих при хранении, обработке, передаче информации является важнейшим методом обеспечения целостности данных. Для борьбы с возникающими ошибками используются методы помехоустойчивого кодирования. В реальных системах шумовые процессы обычно являются коррелированными, однако традиционные методы кодирования и декодирования используют декорреляцию, при этом известно, что эта процедура снижает предельно достижимые характеристики кодирования. Таким образом, актуальной является задача построения вычислительно эффективных методов декодирования, которые позволяли бы бороться с группирующимися ошибками при использовании широкого класса кодов. **Метод.** Для борьбы с одиночными пакетами ошибок использован подход, основанный на декодировании по информационным совокупностям. Несмотря на то, что при исправлении независимых ошибок данный метод имеет экспоненциальную сложность, предложенный подход применяет количество информационных совокупностей, линейно растущее с длиной кода, и обеспечивает, таким образом, полиномиальную сложность декодирования. Дальнейшее уменьшение числа информационных совокупностей возможно с помощью предложенного метода использования плотных информационных совокупностей. Выполнен анализ векторов ошибки, корректно исправляемых предложенными методами. Анализ проведен для кодов небольшой длины на основе стандартной расстановки, позволяющей оценить как множество ошибок, потенциально исправляемых кодом, так и характеристики декодера. **Основные результаты.** Предложен метод декодирования одиночных пакетов на основе выбора линейного числа информационных совокупностей. Описано улучшение метода декодирования с помощью использования счетчика векторов ошибки, позволяющее в ряде случаев увеличить число исправляемых векторов ошибки. Представлен метод декодирования на основании плотных информационных совокупностей, который позволяет значительно снизить количество информационных совокупностей или повысить количество исправляемых векторов ошибок по критерию минимальной длины пакета. Выполненный анализ рассмотренных декодеров с помощью стандартной расстановки показал, что предложенные алгоритмы позволяют исправлять значительное число векторов ошибки сверх гарантированно исправляемой длины пакета. **Обсуждение.** Предложенные декодеры позволяют исправлять одиночные пакеты ошибок за полиномиальное время для произвольных линейных кодов, при этом результаты экспериментов продемонстрировали, что декодеры не только исправляют все ошибки в пределах корректирующей способности кода, но и значительное количество векторов ошибки сверх нее. Направлениями дальнейших исследований возможен анализ предложенных алгоритмов декодирования для длинных кодов, где метод анализа на основе стандартной расстановки неприменим. Также могут быть осуществлены разработка и анализ методов декодирования для множественных пакетов и совместного исправления пакетирующихся и независимых ошибок.

Ключевые слова

информационные совокупности, корректирующая способность, низкоплотностные коды, каналы с памятью, пакеты ошибок

Благодарности

Статья подготовлена в результате проведения исследования в рамках Программы фундаментальных исследований Национального исследовательского университета «Высшая школа экономики» (НИУ ВШЭ), лаборатория Интернета вещей и киберфизических систем НИУ ВШЭ в Санкт-Петербурге.

Ссылка для цитирования: Исаева М.Н., Овчинников А.А. Исправление одиночных пакетов ошибок за пределами корректирующей способности кода с использованием информационных совокупностей // Научно-технический вестник информационных технологий, механики и оптики. 2024. Т. 24, № 1. С. 70–80. doi: 10.17586/2226-1494-2024-24-1-70-80

Correction of single error bursts beyond the code correction capability using information sets

Maria N. Isaeva[1]✉, Andrei A. Ovchinnikov[2]

[1,2] HSE University, Saint Petersburg, 190008, Russian Federation

[1] misaeva@hse.ru✉, https://orcid.org/0009-0007-6228-0617

[2] a.ovchinnikov@hse.ru, https://orcid.org/0000-0002-8523-9429

Abstract

The most important method of ensuring data integrity is correcting errors that occur during information storage, processing or transmission. The error-correcting coding methods are used to correct errors. In real systems, noise processes are correlated. However, traditional coding and decoding methods use decorrelation, and it is known that this procedure reduces the maximum achievable characteristics of coding. Thus, constructing computationally efficient decoding methods that would correct grouped errors for a wide class of codes is an actual problem. In this paper the decoding by information sets is used to correct single bursts. This method has exponential complexity when correcting independent errors. The proposed approach uses a number of information sets linearly growing with code length, which provides polynomial decoding complexity. A further reduction of the number of information sets is possible with the proposed method of using dense information sets. It allows evaluating both the set of errors potentially corrected by the code and the characteristics of the decoder. An improvement of the decoding method using an error vector counter is proposed, which allows in some cases to increase the number of corrected error vectors. This method allows significantly reducing the number of information sets or increasing the number of corrected error vectors according to the minimum burst length criterion. The proposed decoders allow correction of single error bursts in polynomial time for arbitrary linear codes. The results of experiments based on standard array show that decoders not only correct all errors within the burst correcting capability of the code, but also a significant number of error vectors beyond of it. Possible directions of further research are the analysis of the proposed decoding algorithms for long codes where the method of analysis based on the standard array is not applicable; the development and analysis of decoding methods for multiple bursts and the joint correction of grouped and random errors.

Keywords

information sets, error correcting capability, low-density parity-check codes, channels with memory, error bursts

Acknowledgements

The article was prepared within the framework of the Basic Research Program at HSE University, Internet of Things and Cyber-Physical Systems Laboratory, St. Petersburg School of Physics, Mathematics, and Computer Science.

For citation: Isaeva M.N., Ovchinnikov A.A. Correction of single error bursts beyond the code correction capability using information sets. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2024, vol. 24, no. 1, pp. 70–80 (in Russian). doi: 10.17586/2226-1494-2024-24-1-70-80

Введение

С развитием цифровых технологий передача и обработка информации нашла применение во многих сферах деятельности человека. В современных сетях и инфокоммуникационных системах необходимо передавать огромные объемы информации на большие расстояния, обеспечивая требования по скорости передачи, надежности и достоверности. Передача информации осуществляется по каналам связи, особенности которых приводят к появлению ошибок. Вопросы надежной доставки сообщений могут решаться с помощью внесения избыточности и управления ею с помощью резервирования [1], другим подходом является использование методов помехоустойчивого кодирования [2, 3].

Корректирующая способность кодов может быть формально определена, например, как исправление любой комбинации из заданного числа ошибок (декодирование в сфере Хэмминга), либо декодирование всех лидеров смежных классов (полное декодирование) [3]. Вместе с тем для большинства методов декодирования современных кодов (таких как турбо-коды, полярные коды, коды с малой плотностью проверок на четность) множество исправляемых ошибок не имеет простого описания, поэтому вероятность ошибки, обеспечиваемая такими декодерами, определяется путем интенсивного компьютерного моделирования, и актуальным является вопрос о возможности как кодов исправлять ошибки сверх корректирующей способности, так и декодеров реализовывать эти возможности.

В реальных каналах ошибки имеют тенденцию группироваться, образуя так называемые пакеты. Эффект памяти может вызываться многолучевостью распространения данных, архитектурой систем хранения, в том числе распределенных, медленной флюктуацией параметров канала и т. п. [4–6]. Для борьбы с данными пакетами может быть применено перемежение, приводящее к росту задержки обработки информации на передатчике и приемнике, а также ухудшающие предельные возможности кодирования. Другим подходом является построение специальных кодов для таких каналов и декодеров, исправляющих пакеты ошибок. Коды и декодирование для каналов с памятью исследованы крайне мало, что не позволяет говорить не только

о приближении к предельным характеристикам, но и о превышении характеристик традиционных кодов для независимых ошибок в сочетании с перемежением.

В теории кодирования одним из классических способов борьбы с группированием ошибок являются коды Рида–Соломона, в том числе с использованием информации о надежностях принятых символов [7], однако декодирование таких кодов достаточно вычислительно затратно по сравнению с современными методами. Способность исправлять ошибки в каналах с памятью исследовалась для кодов с малой плотностью проверок на четность [8–10], в том числе для моделей каналов с двумя состояниями [11, 12], а также для полярных кодов [13]. Отметим, что в большинстве научных работ часто рассматриваются очень специфические конфигурации ошибок [8] или проводится оптимизация кодовых конструкций, к которым затем применяются стандартные методы декодирования для независимых ошибок. В работе [14] предложен декодер для исправления одиночных пакетов, теоретически применимый к любым линейным кодам, однако он вычислительно эффективен только для кодов с малой плотностью проверок на четность.

Целью данной работы является разработка алгоритма декодирования для исправления пакетов ошибок, который может быть применен к любым линейным кодам, и проведение анализа способности исправлять группирующиеся ошибки сверх корректирующей способности. Характеристики декодирования оцениваются с помощью анализа стандартной расстановки для кодов небольшой длины, для экспериментов выбраны блочно-перестановочные (БП) коды с малой плотностью проверок на четность, широко используемые в современных стандартах связи, а также случайные линейные коды, чья корректирующая способность близка к предельной при исправлении одиночных пакетов ошибок.

Модель ошибок и стандартная расстановка

Рассмотрим модель дискретного двоичного канала, ошибки в котором описываются не с помощью переходных вероятностей, а комбинаторно. Таким образом, изучим не вероятности ошибки декодирования, а способность декодера исправлять конкретные комбинации ошибок. При этом рассмотрим передачу информации блоками по n бит. Введем обозначение одиночных пакетов длиной b как число позиций от первой до последней ненулевой компоненты в векторе из n элементов. Корректирующей способностью кода при исправлении одиночных пакетов обозначим величиной $b_{\max}$, при которой любой вектор ошибки, представляющий собой пакет длиной $b \le b_{\max}$, будет корректно исправлен. Заметим, что определение количества исправляемых независимых ошибок является NP-трудной задачей [15], в то время как максимальная длина исправляемого одиночного пакета может быть вычислена с помощью полиномиальной процедуры [14].

В общем случае множество ошибок, исправляемых линейным кодом при передаче по дискретным каналам связи, может быть описано (хотя и с экспоненциальной сложностью) с помощью стандартной расстановки, представляющей собой таблицу всех возможных двоичных векторов длиной n, в которой первая строка состоит из 2^k кодовых слов, все остальные строки называются смежными классами и представляют собой сумму первой строки с некоторым вектором, не являющимся кодовым словом.

Линейный (n,k)-код — k-мерное подпространство n-мерного пространства двоичных векторов. Базис $\mathbf{G}$ n-мерного пространства называется порождающей матрицей, а базис $\mathbf{H}$ ортогонального пространства размерности $r = n - k$ — проверочной матрицей кода. Для любого вектора $\mathbf{y}$ длиной n вектор $\mathbf{S} = \mathbf{y}\mathbf{H}^T$ называется синдромом.

Все векторы в смежном классе имеют одинаковый синдром, у любых векторов из разных смежных классов синдром различен. Из каждого смежного класса может быть выбран один представитель, который называется лидером смежного класса, таким образом, существует взаимно однозначное соответствие между синдромом и лидером смежного класса, и синдром может рассматриваться как номер смежного класса в стандартной расстановке. Лидер смежного класса считается вектором ошибки для данного синдрома при декодировании по стандартной расстановке [3], другими словами, если $\mathbf{y}$ — принятое из канала слово, синдром которого равен $\mathbf{S}$, то результатом декодирования считается кодовое слово $\mathbf{a} = \mathbf{y} - \mathbf{e}_{\mathbf{S}}$, где $\mathbf{e}_{\mathbf{S}}$ — лидер смежного класса с номером $\mathbf{S}$. Таким образом, если расположить все лидеры в начале соответствующих смежных классов, то первый столбец стандартной расстановки представляет собой список исправляемых векторов ошибок для данного кода. Это означает, что если корректирующая способность кода при исправлении пакетов ошибок равна $b_{\max}$, то все векторы ошибок, представляющие собой пакеты длиной $b \le b_{\max}$, могут быть выбраны в качестве лидеров своих смежных классов (т. е. никакие два таких вектора не лежат в одном классе).

Исправление векторов ошибок в рамках корректирующей способности кода называют декодированием в сфере. Вместе с тем код может исправлять значительное количество векторов ошибок за пределами сферы (т. е. с длиной пакетов, превышающей $b_{\max}$), так как код может исправить все ошибки, являющиеся лидерами смежных классов. Декодирование, при котором любой лидер смежного класса может быть исправлен, называется полным.

В качестве лидеров смежных классов выбирают самые вероятные векторы в канале (тогда полное декодирование называется декодированием по максимуму правдоподобия) или по минимуму веса в некоторой метрике (что соответствует декодированию по минимальному расстоянию). В настоящей работе предположим, что в качестве лидеров выбраны векторы смежного класса, образующие одиночные пакеты наименьшей длины (заметим, что длина пакета не является метрикой, так как для нее не выполняется неравенство треугольника, что не позволяет связать длину исправляемого пакета с расстоянием между кодовыми словами, как это происходит в случае независимых ошибок и метрики Хемминга).

Отметим, что практически все используемые на сегодняшний день декодеры не обеспечивают полного декодирования, таким образом, не реализуя потенциальные возможности кодов. Некоторые декодеры могут гарантировать декодирование в сфере (например, алгебраические декодеры циклических кодов) или не иметь простого описания исправляемых векторов ошибок. Оценка множества исправляемых декодером ошибок является важной задачей, позволяющей как получить более точное понимание свойств декодера, так и проводить аналитические оценки вероятностей ошибки без интенсивного компьютерного моделирования.

Декодирование по информационным совокупностям при исправлении пакетов ошибок

Декодирование по информационным совокупностям является лучшим (по сложности) из известных способов декодирования случайных линейных кодов [16]. Тем не менее, при исправлении независимых ошибок сложность такого декодирования экспоненциальна и определяется числом информационных совокупностей, необходимых для достижения малых вероятностей ошибки. Вместе с тем сложность такого декодирования может быть существенно меньше (полиномиальна) при исправлении пакетов ошибок. Это связано с тем, что для такого декодирования требуется существенно меньше информационных совокупностей.

Множество вида $\gamma = \{1 \le j_1 < j_2 < \ldots < j_k \le n\}$ называется информационной совокупностью, если компоненты кодового слова с номерами из γ однозначно определяют все кодовое слово [5]. Информационная совокупность называется свободной от ошибок, если на ее позициях не произошли ошибки (в векторе ошибок нет ненулевых элементов). В таком случае, принятое слово может быть продекодировано корректно, однако проблемой является определение того, что информационная совокупность свободна от ошибок. Таким образом, декодирование по информационной совокупности состоит в генерировании достаточно большого набора совокупностей и перебора по этому набору в поиске информационной совокупности, свободной от ошибок.

При декодировании пакетов ошибок необходимо учитывать их расположение: генерировать информационную совокупность на позициях, на которых нет позиций вектора ошибки. Пусть код с k информационными символами задан порождающей матрицей $\mathbf{G}$ размера $(k \times n)$. Приведем алгоритм декодирования по информационным совокупностям при исправлении пакетов ошибок с использованием порождающей матрицы $\mathbf{G}$.

Этап 1. Инициализация.

Шаг 1.1. Генерируется множество информационной совокупности $\Gamma = \{\gamma_1, \ldots, \gamma_N\}$.

Шаг 1.2. Формируется набор матриц $\mathbf{G}_{\gamma_i} = \mathbf{M}_{\gamma_i}^{-1}\mathbf{G}$, где $\mathbf{M}_{\gamma_i}$ — это подматрица матрицы $\mathbf{G}$, составленная из столбцов с номерами из множества γ_i.

Этап 2. Декодирование принятого слова $\mathbf{b} = \mathbf{a} \oplus \mathbf{e}$, где $\mathbf{a}$ — кодовое слово; $\mathbf{e}$ — пакет ошибок.

Шаг 2.1. Установить $b_{\min} = n$ и $\mathbf{z}_{\min} = \mathbf{b}$.

Шаг 2.2. Для всех $i = 1 \ldots N$:

— вычислить $\mathbf{z}_i = \mathbf{b}(\gamma_i)\mathbf{G}_{\gamma_i}$ где $\mathbf{b}(\gamma_i)$ — элементы из принятого слова $\mathbf{b}$ на позициях γ_i;

— вычислить $\mathbf{e}_i = \mathbf{b} \oplus \mathbf{z}_i$ и длину b_i пакета в векторе $\mathbf{e}_i$;

— если $b_i < b_{\min}$, то $\mathbf{z}_{\min} = \mathbf{z}_i$ и $b_{\min} = b_i$.

Этап 3. Принятие решения о декодированном слове: $\hat{\mathbf{a}} = \mathbf{z}_{\min}$.

Отметим, что в представленном алгоритме декодирования решение принимается в пользу кодового слова, для которого минимальна длина произошедшего одиночного пакета ошибок. Это обусловлено тем, что несмотря на то, что не была введена вероятностная модель ошибок, наиболее вероятны более короткие пакеты, чем более длинные — данное соображение согласуется со многими вероятностными моделями каналов с памятью, например моделями с конечным числом состояний (Гилберта или Гилберта–Эллиотта).

Сложность декодирования в рассмотренном алгоритме определяется числом информационных совокупностей N. Чтобы для каждого возможного расположения пакета длиной b (т. е. позиций начала пакета от 1 до $n - b + 1$) нашлась информационная совокупность, свободная от ошибок, необходимо сформировать множество Γ, состоящее из информационных совокупностей на позициях, не вошедших в пакет. Тогда $N = n - b + 1$ и сложность алгоритма декодирования является полиномиальной (кроме перебора по линейному от n количеству информационных совокупностей, необходимо хранить множество Γ, матрицы $\mathbf{G}_{\gamma_i}$ и вычислить $\mathbf{z}_i$). Более точно, вычислительная сложность этапа 1 не учитывается при оценке сложности декодирования, так как для выбранного кода выполняется один раз. При этом при случайном выборе информационной совокупности в случайном коде вероятность ее нахождения составляет около 0,3 [17]. Таким образом, шаг 1.1 выполняется ожидаемо за $3n$ попыток, при $n \approx N$. Шаг 1.2 требует $O(n)$ обращений матриц и столько же матричных умножений. Результатом этапа 1 является хранение $O(n)$ порождающих матриц размером $k \times n$, т. е. $O(kn^2)$ бит, а также kn ячеек памяти на хранение множества Γ. Цикл на этапе 2 шага 2.1 выполняется $O(n)$ раз, каждая итерация цикла требует $k + 1$ операций «исключающее ИЛИ» векторов длиной n, а также определения длины пакета со сложностью $O(n)$, что дает вычислительную сложность $O(kn)$ битовых операций. Таким образом, можно без применения оптимизации оценить сложность декодера как $O(kn^2)$ памяти и операций. Для сравнения, асимптотическое время выполнения декодера кода Рида–Соломона составляет $O(n^2)$, однако операции проводятся в конечном поле $GF(q)$, что более затратно при практической реализации.

Параметр b наиболее естественно выбрать, исходя из корректирующей способности кода, $b \le b_{\max}$, однако при выборе больших значений b алгоритм будет пытаться продекодировать сверх корректирующей способности.

Заметим, что в рассмотренном алгоритме в процессе декодирования формируется список возможных векторов ошибки для каждой информационной совокупности, и выбор происходит в пользу вектора ошиб-

ки, представляющего собой пакет наименьшей длины. Если пакет с такой длиной один, то решение декодера однозначно, однако пакетов ошибок с одинаковой минимальной длиной может быть несколько. В этом случае декодер может выбрать произвольный пакет ошибок; сигнализировать о невозможности принять однозначное решение или воспользоваться каким-либо дополнительным критерием выбора.

Рассмотрим следующую модификацию декодера, установив дополнительный критерий отбора вектора ошибки при декодировании, кроме длины пакета — счетчик встречаемости вектора ошибки в результирующем списке пакетов минимальной длины. Разные информационные совокупности, на которых отсутствуют ошибки, будут восстанавливать одно и то же кодовое слово, и следовательно — один и тот же вектор ошибки. Тогда можно ожидать, что верный вектор будет встречаться чаще других. Таким образом, если после декодирования получилось несколько векторов ошибки с одинаковой минимальной длиной пакета, выберем среди них тот, который встречается в списке чаще всего. При этом в случае, если есть несколько векторов с одинаковым значением счетчика, снова будет получена неоднозначность результата декодирования.

Рассмотрим подход к уменьшению количества используемых информационных совокупностей. Пусть b_1 и b_2 — номера начала и конца пакета в некотором векторе ошибок. Предположим, что происходит циклическое движение вправо по позициям вектора ошибки, начиная от позиции $b_2 + 1$ (при достижении конца вектора движение продолжится с начала вектора). Первую встреченную при таком обходе позицию информационной совокупности обозначим j_1, последнюю — j_k. При такой нумерации назовем плотной информационную совокупность, у которой минимизируется разность $(j_k - j_1)\mathrm{mod}n$, т. е. минимизируется разброс от первой до последней позиций информационной совокупности с учетом цикличности номеров позиций.

Использование таких плотных информационных совокупностей позволяет сократить количество информационных совокупностей при декодировании, так как одна информационная совокупность может оказаться свободной от ошибок для большего количества расположений пакетов ошибок.

При таком подходе можно значительно уменьшить количество информационных совокупностей (эксперименты показывают, что в ряде случаев — до двух или трех, в зависимости от параметров кода и максимальной исправляемой длины пакета). При использовании плотных информационных совокупностей можно считать $N = \mathrm{const}$, тогда декодирование требует $O(kn)$ памяти и битовых операций.

Отметим, что декодирование с дополнительным использованием счетчика может не дать заметного эффекта для плотных информационных совокупностей, так как значение счетчика может просто не накопиться из-за малого количества плотных информационных совокупностей. В связи с этим рассмотрим дополнительный режим декодирования, при котором построим плотные информационные совокупности для каждого возможного расположения пакета. В этом случае цель использования плотных информационных совокупностей — не минимизация числа информационных совокупностей, а накопление значения счетчика для правильного слова при декодировании по информационным совокупностям, свободным от ошибок.

Описание экспериментов и классификация результатов декодирования

В разделе «Модель ошибок и стандартная расстановка» полное декодирование определено как исправление всех лидеров смежных классов, выбранных по какому-либо критерию. Однако декодер, обеспечивающий полное декодирование, можно определить при другом условии: пусть для каждого возможного синдрома декодер верно декодирует какой-либо вектор ошибки из соответствующего смежного класса. Другими словами, это означает, что существует такой способ выбора списка лидеров смежного класса, что рассматриваемый декодер будет их верно декодировать, если они произошли в канале. При этом, если лидеры смежных классов выбраны традиционно в соответствии с некоторым критерием (например, минимальным весом Хемминга, минимальной длиной пакета, максимальной вероятностью и т. д.), такое декодирование, являясь полным в смысле данного определения, может не обеспечивать полное декодирование по заданному критерию.

Оценим множество векторов ошибок, исправляемых описанным декодером по информационным совокупностям. Выполним оценку экспериментально, на примере кодов небольшой длины, для которых можно результаты декодирования сравнить с расположением векторов в стандартной расстановке. Для проведения экспериментов выберем следующие коды.

БП-коды с малой плотностью проверок на четность БП(γ, ρ) задаются $(m\gamma \times m\rho)$-проверочной матрицей, состоящей из блоков — степеней матрицы циклической перестановки [18, 19], т. е. имеющей вид:

$$\mathbf{H} = \begin{bmatrix} \mathbf{C}^{t_{11}} & \dots & \mathbf{C}^{t_{1\rho}} \\ \dots & \dots & \dots \\ \mathbf{C}^{t_{\gamma 1}} & \dots & \mathbf{C}^{t_{\gamma\rho}} \end{bmatrix},$$

где $\mathbf{C}$ — матрица циклической перестановки; ρ — количество блоков в строке; γ — количество блоков в столбце; m — размер блока; степени t_{ij} — целые неотрицательные числа. Для задания такой матрицы достаточно указать (γ, ρ)-матрицу степеней t_{ij}, называемую базовой матрицей.

Для исследования выбраны пять кодов с параметрами, приведенными в табл. 1.

Для БП-кодов максимальная длина исправляемого пакета $b_{\max} \le m - 1$ [14]. Выбран код 1 с базовой матрицей

$$\mathbf{H}_{\mathbf{base}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 2 \end{bmatrix} \qquad (1)$$

с размером блоков $m = 7$. Получена для кода 1 скорость $R = 0{,}38$ и $b_{\max} = 6$, и для которого $N_{\min} = 2$ плотные информационные совокупности покрывают все возможные расположения пакета. Код, задаваемый матрицей (1), является кодом Гилберта, его способность

Таблица 1. Параметры выбранных кодов
Table 1. Parameters of selected codes

Номер кода	Конструкция	n	k	R	$N_{\min}$	$b_{\max}$	Граница Рейгера
1	БП(2,3)	21	8	0,38	2	6	6
2	БП(2,4)	20	11	0,55	3	4	4
3	случайный	20	7	0,35	3	5	6
4	случайный	20	10	0,5	4	3	5
5	случайный	20	13	0,65	5	2	3

исправлять одиночные пакеты ошибок рассмотрены в работе [20].

Рассмотрим более высокоскоростной код 2 со скоростью $R = 0{,}55$, для которого базовая матрица имеет вид

$$\mathbf{H_{base}} = \begin{bmatrix} 0 & 2 & 1 & 1 \\ 2 & 3 & 4 & 1 \end{bmatrix} \qquad (2)$$

и размер блоков $m = 5$. Для этого кода $b_{\max} = 4$, $N_{\min} = 3$. Особенности нахождения информационных совокупностей в БП-кодах приведены в [17].

Выберем три случайных линейных кода, сгенерировав порождающие матрицы с вероятностью элементов в них 0,5: код 3 ($R = 0{,}35$, $b_{\max} = 5$, $N_{\min} = 3$), код 4 ($R = 0{,}5$, $b_{\max} = 3$, $N_{\min} = 4$) и код 5 ($R = 0{,}65$, $b_{\max} = 2$, $N_{\min} = 5$).

Отметим, что в соответствии с границей Рейгера для линейных (n, k)-кодов, максимальная длина одиночного исправляемого пакета не превышает $b_{\max} \le \lfloor (n-k)/2 \rfloor$ [3]. При этом для выбранных кодов их корректирующая способность лежит на границе или близка к ней.

Для проведения эксперимента построим стандартную расстановку для выбранного кода, отбирая в лидеры смежных классов пакеты наименьшей длины. Поставим на вход декодера все возможные векторы ошибок (не только лидеры смежных классов), т. е. будем декодировать каждый вектор стандартной расстановки. Предположим, что декодер в качестве результата выдает вектор ошибки, учитывая, что результатом декодирования также может быть список таких векторов (в случае, если несколько векторов ошибки имеют одинаковую минимальную длину пакета), приводящий к неоднозначности принятия решения. Пусть «*F*» — обозначение того, что декодером получен однозначный ответ, тем не менее, не являющийся правильным вектором ошибки (т. е. однозначное и неверное декодирование); «*T*» — получен однозначный ответ, совпадающий с правильным вектором ошибки (однозначно верно); «*T**» — получено несколько векторов ошибки с минимальной длиной пакета, но среди них есть верный; «*F**» — получено несколько векторов минимальной длины пакета, и среди них нет верного.

Выполним анализ декодирования с точки зрения способности декодера исправлять пакеты в пределах корректирующей способности, а также за ее пределами. Для этого введем следующую классификацию результатов декодирования.

Назовем смежный класс декодируемым (однозначно декодируемым), если в нем есть единственный вектор ошибок, однозначно исправляемый данным декодером. Назовем смежный класс вероятностно декодируемым, если исправляемых векторов ошибки в нем нет, но в некоторых случаях декодер возвращает список, содержащий правильный вектор ошибки. Для классификации декодера выделим следующие результаты декодирования для каждого смежного класса.

Случай I (декодируемый смежный класс по выбранному критерию): смежный класс декодируется верно и однозначно в пакет минимальной длины. Это означает, что для векторов ошибки из данного смежного класса один раз декодер вернул результат «*T*», и этот результат был получен для лидера.

Продекодированный вектор ошибок может не находиться в лидерах смежного класса, выбранных по некоторому принципу, в рассматриваемом случае — по минимальной длине пакета. При этом возможны два варианта:

вариант 1 — в смежном классе есть несколько пакетов одинаковой минимальной длины (это возможно только для пакетов длиной, большей $b_{\max}$), и продекодирован пакет минимальной длины, но не тот, который был выбран в лидеры;

вариант 2 — верно и однозначно продекодирован пакет, более длинный, чем лидер смежного класса.

При варианте 1 выполним переупорядочение стандартной расстановки (смежного класса) по результатам декодирования, и выберем верно продекодированный пакет в лидеры, при этом не меняется распределение длин исправляемых пакетов, находящихся в лидерах. Этот вариант не будем отличать от случая I, когда верно продекодированным вектором ошибок сразу оказался лидер. При варианте 2 получим следующий случай.

Случай I* (декодируемый смежный класс): при однозначном и верном декодировании смежного класса правильно продекодированный вектор ошибок не является лидером и не является пакетом минимальной длины. Это означает, что для данного смежного класса (и соответствующего синдрома) декодер не обеспечивает декодирование по выбранному критерию, однако смежный класс имеет единственный верно декодируемый вектор ошибки.

Случай II (вероятностно декодируемый смежный класс по выбранному критерию): лидеру смежного класса соответствует результат декодирования «*T**», т. е. декодер возвратил список векторов ошибки, среди которых есть и правильный, и при случайном выборе есть вероятность верного декодирования, зависящая от размера списка (можно ожидать, что он не будет боль-

шим). При этом предполагается, что результатов «*T*» в смежном классе нет.

Случай II* (вероятностно декодируемый смежный класс): аналогично случаю I*, смежный класс содержит результаты декодирования «*T**», однако они не соответствуют векторам с минимальной для данного смежного класса длиной пакета.

Случай III (не декодируемый смежный класс): результаты декодирования векторов смежного класса равны «*F*» или «*F**», т. е. вероятность правильного декодирования любого вектора ошибки из данного смежного класса равна нулю, вне зависимости от того, принял декодер однозначное решение либо нет.

Схематически описанные случаи представлены на рис. 1. Основываясь на выделенных результатах декодирования смежных классов, декодирование может быть классифицировано следующим образом:

— полное декодирование по выбранному критерию: декодирование каждого смежного класса относится к случаю I;
— полное декодирование: декодирование каждого смежного класса относится к случаям I или I*;
— вероятностное полное декодирование по выбранному критерию: декодирование каждого смежного класса относится к случаям I или II;
— вероятностное полное декодирование: декодирование каждого смежного класса относится к случаям I, I*, II или II*;
— декодирование в сфере по выбранному критерию: декодирование смежных классов с лидерами, имеющими длину пакета, не превышающую b_{max}, соответствует случаю I;
— декодирование в сфере: декодирование смежных классов с лидерами, имеющими длину пакета, не превышающую b_{max}, соответствует случаям I или II;
— вероятностное декодирование в сфере по выбранному критерию: декодирование смежных классов с лидерами, имеющими длину пакета, не превышающую b_{max}, соответствует случаям I или I*;
— вероятностное декодирование в сфере: декодирование смежных классов с лидерами, имеющими длину пакета, не превышающую b_{max}, соответствует случаям I, I*, II или II*.

Заметим, что декодирование по информационным совокупностям является вероятностным в том смысле, что при одинаковых параметрах (числе информационных совокупностей *N*) результаты декодирования могут зависеть от конкретного множества информационных совокупностей, которое строится случайно.

При использовании декодера с дополнительным принятием решения по счетчику может быть достигнуто улучшение полученных характеристик, так как повысится доля декодируемых смежных классов. Для этого декодера введем дополнительные обозначения результатов декодирования: «T_c» (*True with counter*) — однозначное и верное декодирование после использования значения счетчика (т. е. результат «*T**» стал результатом «*T*»); «F_c» — однозначное неверное декодирование после выбора по счетчику; «T_c*» — неоднозначность не была разрешена (среди векторов ошибки с одинаковой минимальной длиной пакета есть несколько, имеющих одинаковое значение счетчика), но в списке присутствует верный вектор; «F_c*» — неоднозначность не была разрешена, и верный вектор в списке отсутствует. Теперь однозначно, верно, декодированным векторам соответствуют результаты декодирования «*T*» или «T_c». В табл. 2 представлены параметры декодирования, которые будут использованы в ходе экспериментов.

Результаты экспериментов

С учетом введенных в разделе «Описание экспериментов и классификация результатов декодирования» обозначений проведем эксперименты с кодом с базовой матрицей (1) (код 1, табл. 1). Так как код 1 задается (21,8)-матрицей, то его стандартная расстановка содержит 8191 смежных класса (не считая самого множества

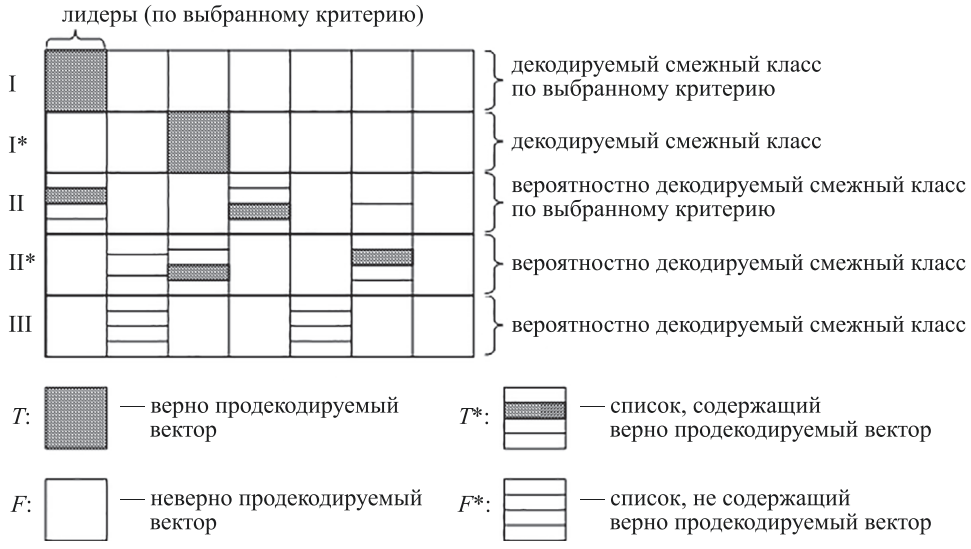

Рис. 1. Классификация результатов декодирования по смежным классам

Fig. 1. Classification of decoding results by cosets

Таблица 2. Параметры декодирования
Table 2. Decoding parameters

Обозначение	Параметры декодирования
A	$N = n - b + 1$ информационная совокупность для каждого возможного начала пакета;
B	$N = n - b + 1$ информационная совокупность для каждого возможного начала пакета и использование счетчика;
C	$N_{\min}$ плотных информационных совокупностей;
D	$N_{\min}$ плотных информационных совокупностей и использование счетчика;
E	$N = n - b + 1$ плотных информационных совокупностей для каждого возможного начала пакета;
F	$N = n - b + 1$ плотных информационных совокупностей для каждого возможного начала пакета и использование счетчика.

кодовых слов), каждый смежный класс содержит 256 векторов.

На рис. 2 представлены результаты декодирования векторов из стандартной расстановки. Базовый алгоритм декодирования помечен как (1), с использованием счетчика — как (2). Представлена доля декодируемых смежных классов (с выделением случая декодирования в рамках корректирующей способности), вероятностно декодируемых и не декодируемых.

Заметим, что во всех случаях в смежных классах результаты декодирования содержат одну «*T*» и остальные «*F*» (т. е. соответствуют случаям I и I*) или содержат некоторое количество «*T**» и остальные «*F**» (что соответствует случаям II и II*). Случая III ни для какого декодера получено не было. Также отметим, что во всех случаях рис. 2 гарантированно исправляются все лидеры, представляющие собой пакеты длиной до $b_{\max}$. Таким образом, можно сделать вывод, что для

Таблица 3. Результаты декодирования для выбранных кодов, %
Table 3. Decoding results for selected codes, %

Номер кода	Параметры декодирования	Выделенные классы				
		случай I, вариант 1 ($b \le b_{\max}$)	случай I, вариант 2 ($b > b_{\max}$)	случай I*	случай II	случай II*
2	A	28,13	44,73	13,09	9,96	4,10
	B	28,13	51,96	14,47	2,34	3,13
	C	28,13	46,68	8,59	9,77	6,84
	D	28,13	46,68	8,59	8,98	7,62
	E	28,13	47,66	—	24,23	—
	F	28,13	47,66	24,23	—	—
3	A	3,32	70,21	7,50	13,88	5,09
	B	3,32	77,78	11,03	4,76	3,11
	C	3,32	58,82	36,11	1,04	0,71
	D	3,32	58,87	36,11	0,43	1,26
	E	3,32	71,37	—	25,31	—
	F	3,32	91,12	—	5,55	—
4	A	7,42	54,1	21,58	5,96	10,97
	B	7,42	57,52	27,44	2,15	5,47
	C	7,42	49,61	29,79	3,42	9,77
	D	7,42	49,61	29,78	3,22	9,96
	E	7,42	66,02	—	26,56	—
	F	7,42	86,04	—	6,54	—
5	A	31,25	34,38	19,53	3,91	10,94
	B	31,25	36,72	28,12	1,56	2,34
	C	31,25	42,97	7,81	11,72	6,25
	D	31,25	42,97	7,81	10,16	7,81
	E	31,25	42,97	—	25,78	—
	F	31,25	62,5	—	6,25	—

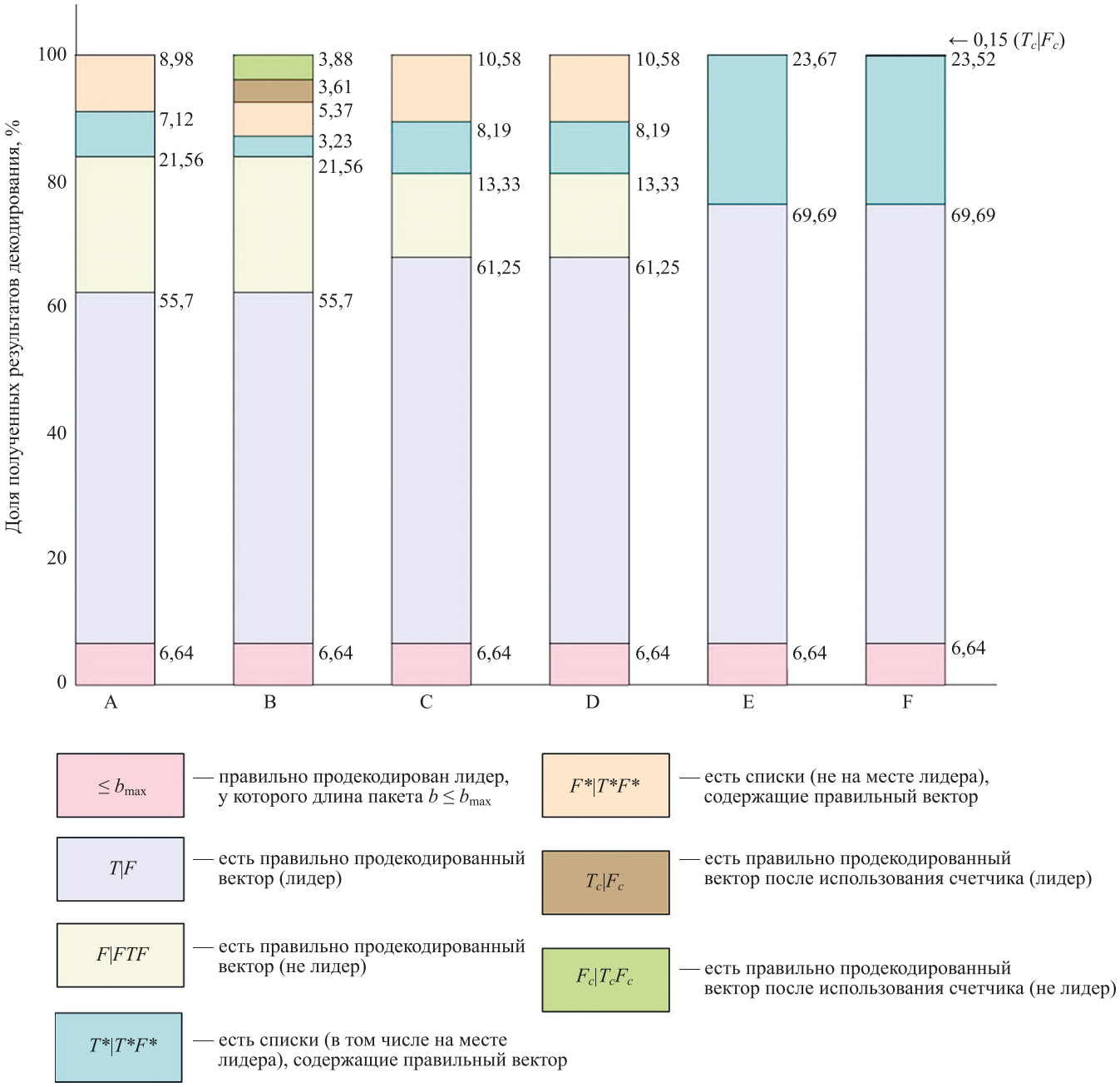

Рис. 2. Результаты декодирования по информационным совокупностям по минимальной длине пакета для блочно-перестановочного кода 1 (табл. 1) с использованием параметров декодирования из табл. 2

Fig. 2. Results of information set decoding by minimal burst length for block-permutation code 1 (Table 1) using decoding parameters from Table 2

всех рассмотренных декодеров обеспечивается декодирование в сфере по критерию минимальной длины пакета, а также вероятностное полное декодирование. Видно, что большая доля пакетов, имеющих длину сверх корректирующей способности, также могут быть однозначно исправлены.

В целом можно сделать вывод, что использование счетчика повышает долю однозначно декодируемых смежных классов для алгоритма с распределенными информационными совокупностями и практически не имеет эффекта при использовании плотных информационных совокупностей. При этом использование плотных информационных совокупностей имеет большую долю исправляемых лидеров смежных классов (еще большую при использовании плотных информационных совокупностей для всех расположений пакетов), но меньшую долю общего числа декодируемых смежных классов.

В табл. 3 представлены результаты декодирования для БП-кода и случайных кодов, номера и параметры которых представлены в табл. 1. Для БП-кода 2 (табл. 1) можно сделать выводы, аналогичные результатам для БП-кода 1 на рис. 2. Вместе с тем отметим, что при использовании плотных информационных совокупностей для каждого расположения пакета получено полное декодирование.

Рассмотрим теперь случайные коды. В табл. 3 представлены результаты декодирования для случайного кода 3 (табл. 1) с параметрами $n = 20$, $k = 7$ и $b_{\max} = 5$. По сравнению с БП-кодами, использование плотных информационных совокупностей снижает долю декодирований лидеров, однако существенно увеличивает количество декодируемых смежных классов, в которых исправляемый вектор не является лидером. Рассмотрим результаты для случайного кода 4 (табл. 1) с параметрами $n = 20$, $k = 10$, $b_{\max} = 3$ и результаты для случайного кода 5 $n = 20$, $k = 7$ с максимальной длиной исправляемого пакета $b_{\max} = 2$.

Подводя итог, можно сделать вывод, что для всех рассмотренных кодов все декодеры обеспечивают декодирование в рамках корректирующей способности. Также во всех случаях отсутствуют не декодируемые смежные классы, что с учетом используемой терминологии описано как вероятностное полное декодирование. Во всех случаях доля декодируемых векторов ошибки значительно превосходит корректирующую способность кода. Декодирование по критерию минимальной длины пакета лучше всего обеспечивается при использовании плотных информационных совокупностей для всех расположений пакета.

Заключение

В работе рассмотрено применение декодирования по информационным совокупностям для исправления одиночных пакетов ошибок. Введено понятие плотной информационной совокупности, которое может быть использовано как для сокращения их числа при декодировании в рамках корректирующей способности $b_{\max}$, так и для увеличения доли исправляемых пакетов сверх $b_{\max}$. Представлена модификация декодера с применением счетчика векторов ошибок, что позволяет уменьшить количество неоднозначных результатов декодирования. Приведена классификация декодирования, обобщающая понятие полного декодирования.

Выполнены эксперименты для кодов с небольшими параметрами: блочно-перестановочных кодов с малой плотностью проверок на четность, а также случайных линейных кодов, показывающие, что все рассмотренные декодеры обеспечивают для выбранных кодов декодирование в рамках корректирующей способности, а также исправление значительной доли векторов ошибок сверх корректирующей способности. Кроме этого, для рассмотренных кодов и декодеров не существует смежных классов, вероятность правильного декодирования в которых равнялась бы нулю, таким образом, данные декодеры обеспечивают ненулевую вероятность полного декодирования.

Возможными направлениями дальнейших исследований должен быть анализ предложенных алгоритмов декодирования для длинных кодов, где эффект памяти канала проявляется более выраженно, а также разработка и анализ методов декодирования для множественных пакетов и совместного исправления пакетирующихся и независимых ошибок. Необходимо принять во внимание, что это приведет к увеличению сложности декодирования.

Литература

1. Богатырев В.А., Богатырев С.В., Богатырев А.В. Оценка готовности компьютерной системы к своевременному обслуживанию запросов при его совмещении с информационным восстановлением памяти после отказов // Научно-технический вестник информационных технологий, механики и оптики. 2023. Т. 23. № 3. С. 608–617. https://doi.org/10.17586/2226-1494-2023-23-3-608-617
2. Moon T.K. Error Correction Coding: Mathematical Methods and Algorithms. Wiley, 2021. 992 p.
3. MacWilliams F.J., Sloane N.J.A. The Theory of Error-Correcting Codes. North-Holland Publishing Company, 1983. 762 p. (North-Holland Mathematical Library; Vol. 16).
4. Krouk E., Ovchinnikov A., Poikonen J. Channel models and reliable communication // Modulation and coding techniques in Wireless communications. Wiley, 2011. P. 1–20. https://doi.org/10.1002/9780470976777.ch1
5. Bogatyrev V.A., Bogatyrev A.V., Bogatyrev S.V. Multipath transmission of heterogeneous traffic in acceptable delays with packet replication and destruction of expired replicas in the nodes that make up the path // Communications in Computer and Information Science. 2023. V. 1748. P. 104–121. https://doi.org/10.1007/978-3-031-30648-8_9
6. Bogatyrev V., Bogatyrev S., Bogatyrev A. Timeliness of multipath redundant transmissions when all paths are not accessible for some request sources // Studies in Systems, Decision and Control. 2023. V. 457. P. 671–681. https://doi.org/10.1007/978-3-031-22938-1_46
7. Kulhandjian M., Kulhandjian H., D'Amours C. Improved soft decoding of Reed-Solomon codes on Gilbert-Elliott channels // Proc. of the IEEE International Symposium on Information Theory (ISIT). 2019. P. 1072–1076. https://doi.org/10.1109/ISIT.2019.8849456
8. Xie N., Zhang T., Haratsch E.F. Improving burst error tolerance of LDPC-centric coding systems in read channel // IEEE Transactions on Magnetics. 2010. V. 46. N 3. P. 933–941. https://doi.org/10.1109/TMAG.2009.2034012

References

1. Bogatyrev V.A., Bogatyrev S.V., Bogatyrev A.V. Assessment of the readiness of a computer system for timely servicing of requests when combined with information recovery of memory after failures. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2023, vol. 23, no. 3, pp. 608–617. (in Russian). https://doi.org/10.17586/2226-1494-2023-23-3-608-617
2. Moon T.K. *Error Correction Coding: Mathematical Methods and Algorithms*. Wiley, 2021, 992 p.
3. MacWilliams F.J., Sloane N.J.A. *The Theory of Error-Correcting Codes*. North-Holland Publishing Company, 1983, 762 p. North-Holland Mathematical Library, Vol. 16.
4. Krouk E., Ovchinnikov A., Poikonen J. Channel models and reliable communication. *Modulation and Coding Techniques in Wireless Communications*. Wiley, 2011, pp. 1–20. https://doi.org/10.1002/9780470976777.ch1
5. Bogatyrev V.A., Bogatyrev A.V., Bogatyrev S.V. Multipath transmission of heterogeneous traffic in acceptable delays with packet replication and destruction of expired replicas in the nodes that make up the path. *Communications in Computer and Information Science*, 2023, vol. 1748, pp. 104–121. https://doi.org/10.1007/978-3-031-30648-8_9
6. Bogatyrev V., Bogatyrev S., Bogatyrev A. Timeliness of multipath redundant transmissions when all paths are not accessible for some request sources. *Studies in Systems, Decision and Control*, 2023, vol. 457, pp. 671–681. https://doi.org/10.1007/978-3-031-22938-1_46
7. Kulhandjian M., Kulhandjian H., D'Amours C. Improved soft decoding of Reed-Solomon codes on Gilbert-Elliott channels. *Proc. of the IEEE International Symposium on Information Theory (ISIT)*, 2019, pp. 1072–1076. https://doi.org/10.1109/ISIT.2019.8849456
8. Xie N., Zhang T., Haratsch E.F. Improving burst error tolerance of LDPC-centric coding systems in read channel. *IEEE Transactions on Magnetics*, 2010, vol. 46, no. 3, pp. 933–941. https://doi.org/10.1109/TMAG.2009.2034012

9. Yang M., Pan Z., Djordjevic I.B. FPGA-based burst-error performance analysis and optimization of regular and irregular SD-LDPC codes for 50G-PON and beyond // Optics Express. 2023. V. 31. N 6. P. 10936–10946. https://doi.org/10.1364/OE.477546
10. Xiao X., Vasić B., Lin S., Li J., Abdel-Ghaffar K. Quasi-cyclic LDPC codes with parity-check matrices of column weight two or more for correcting phased bursts of erasures // IEEE Transactions on Communications. 2021. V. 69. N 5. P. 2812–2823. https://doi.org/10.1109/TCOMM.2021.3059001
11. Eckford A., Kschischang F., Pasupathy S. Analysis of low-density parity-check codes for the gilbert–elliott channel // IEEE Transactions on Information Theory. 2005. V. 51. N 11. P. 3872–3889. https://doi.org/10.1109/TIT.2005.856934
12. Ovchinnikov A., Veresova A., Fominykh A. Usage of LDPC codes in a gilbert channel // Труды учебных заведений связи. 2022. Т. 8. № 4. С. 55–63. https://doi.org/10.31854/1813-324X-2022-8-4-55-63
13. Ghaddar N., Kim Y.-H., Milstein L.B., Ma L., Yi B.K. Joint channel estimation and coding over channels with memory using polar codes // IEEE Transactions on Communications. 2021. V. 69. N 10. P. 6575–6589. https://doi.org/10.1109/TCOMM.2021.3098822
14. Ovchinnikov A.A., Veresova A.M., Fominykh A.A. Decoding of linear codes for single error bursts correction based on the determination of certain events // Информационно-управляющие системы. 2022. № 6. С. 41–52. https://doi.org/10.31799/1684-8853-2022-6-41-52
15. Barg A., Krouk E., van Tilborg H.C.A. On the complexity of minimum distance decoding of long linear codes // IEEE Transactions on Information Theory. 1999. V. 45. N 5. P. 1392–1405. https://doi.org/10.1109/18.771141
16. Both L., May A. Optimizing BJMM with nearest neighbors: full decoding in 22/21n and McEliece security // WCC Workshop on Coding and Cryptography. 2017. V. 214.
17. Исаева М.Н. Поиск информационных совокупностей при исправлении пакетов ошибок квазициклическими кодами // T-Comm: Телекоммуникации и транспорт. 2023. Т. 17. № 7. С. 4–12. https://doi.org/10.36724/2072-8735-2023-17-7-4-12
18. Fossorier M.P.C. Quasicyclic low-density parity-check codes from circulant permutation matrices // IEEE Transactions on Information Theory. 2004. V. 50. N 8. P. 1788–1793. https://doi.org/10.1109/TIT.2004.831841
19. Xiao X., Vasić B., Lin S., Abdel-Ghaffar K., Ryan W.E. Reed-Solomon based quasi-cyclic LDPC codes: designs, girth, cycle structure, and reduction of short cycles // IEEE Transactions on Communications. 2019. V. 67. N 8. P. 5275–5286. https://doi.org/10.1109/TCOMM.2019.2916605
20. Крук Е.А., Овчинников А.А. Точная корректирующая способность кодов Гилберта при исправлении пакетов ошибок // Информационно-управляющие системы. 2016. № 1. С. 80–87. https://doi.org/10.15217/issn1684-8853.2016.1.80

9. Yang M., Pan Z., Djordjevic I.B. FPGA-based burst-error performance analysis and optimization of regular and irregular SD-LDPC codes for 50G-PON and beyond. *Optics Express*, 2023, vol. 31, no. 6, pp. 10936–10946. https://doi.org/10.1364/OE.477546
10. Xiao X., Vasić B., Lin S., Li J., Abdel-Ghaffar K. Quasi-cyclic LDPC codes with parity-check matrices of column weight two or more for correcting phased bursts of erasures. *IEEE Transactions on Communications*, 2021, vol. 69, no. 5, pp. 2812–2823. https://doi.org/10.1109/TCOMM.2021.3059001
11. Eckford A., Kschischang F., Pasupathy S. Analysis of low-density parity-check codes for the gilbert–elliott channel. *IEEE Transactions on Information Theory*, 2005, vol. 51, no. 11, pp. 3872–3889. https://doi.org/10.1109/TIT.2005.856934
12. Ovchinnikov A., Veresova A., Fominykh A. Usage of LDPC codes in a gilbert channel. *Proceedings of Telecommunication Universities*, 2022, vol. 8, no. 4, pp. 55–63. https://doi.org/10.31854/1813-324X-2022-8-4-55-63
13. Ghaddar N., Kim Y.-H., Milstein L.B., Ma L., Yi B.K. Joint channel estimation and coding over channels with memory using polar codes. *IEEE Transactions on Communications*, 2021, vol. 69, no. 10, pp. 6575–6589. https://doi.org/10.1109/TCOMM.2021.3098822
14. Ovchinnikov A.A., Veresova A.M., Fominykh A.A. Decoding of linear codes for single error bursts correction based on the determination of certain events. *Information and Control Systems*, 2022, no. 6, pp. 41–52. https://doi.org/10.31799/1684-8853-2022-6-41-52
15. Barg A., Krouk E., van Tilborg H.C.A. On the complexity of minimum distance decoding of long linear codes. *IEEE Transactions on Information Theory*, 1999, vol. 45, no. 5, pp. 1392–1405. https://doi.org/10.1109/18.771141
16. Both L., May A. Optimizing BJMM with nearest neighbors: full decoding in 22/21n and McEliece security. *WCC Workshop on Coding and Cryptography*, 2017, vol. 214.
17. Isaeva M.N. Finding information sets when correcting error bursts with quasi-cyclic codes. *T-Comm*, 2023, vol. 17, no. 7, pp. 4–12. (in Russian). https://doi.org/10.36724/2072-8735-2023-17-7-4-12
18. Fossorier M.P.C. Quasicyclic low-density parity-check codes from circulant permutation matrices. *IEEE Transactions on Information Theory*, 2004, vol. 50, no. 8, pp. 1788–1793. https://doi.org/10.1109/TIT.2004.831841
19. Xiao X., Vasić B., Lin S., Abdel-Ghaffar K., Ryan W.E. Reed-Solomon based quasi-cyclic LDPC codes: designs, girth, cycle structure, and reduction of short cycles. *IEEE Transactions on Communications*, 2019, vol. 67, no. 8, pp. 5275–5286. https://doi.org/10.1109/TCOMM.2019.2916605
20. Krouk E.A., Ovchinnikov A.A. Exact burst-correction capability of gilbert codes. *Information and Control Systems*, 2016, no. 1, pp. 80–87. (in Russian). https://doi.org/10.15217/issn1684-8853.2016.1.80

Авторы

Исаева Мария Николаевна — младший научный сотрудник, Национальный исследовательский университет «Высшая школа экономики», Санкт-Петербург, 190008, Российская Федерация, sc 57243599200, https://orcid.org/0009-0007-6228-0617, misaeva@hse.ru

Овчинников Андрей Анатольевич — кандидат технических наук, доцент, ведущий научный сотрудник, Национальный исследовательский университет «Высшая школа экономики», Санкт-Петербург, 190008, Российская Федерация, sc 57207711162, https://orcid.org/0000-0002-8523-9429, a.ovchinnikov@hse.ru

Authors

Maria N. Isaeva — Junior Researcher, HSE University, Saint Petersburg, 190008, Russian Federation, sc 57243599200, https://orcid.org/0009-0007-6228-0617, misaeva@hse.ru

Andrei A. Ovchinnikov — PhD, Associate Professor, Leading Researcher, HSE University, Saint Petersburg, 190008, Russian Federation, sc 57207711162, https://orcid.org/0000-0002-8523-9429, a.ovchinnikov@hse.ru

Статья поступила в редакцию 23.10.2023
Одобрена после рецензирования 01.12.2023
Принята к печати 13.01.2024

Received 23.10.2023
Approved after reviewing 01.12.2023
Accepted 13.01.2024