УДК 004.056.55: 303.732.4: 519.1

О. В. Кузьмин[1], И. А. Зеленцов[1]

[1]Иркутский государственный университет, г. Иркутск, Российская Федерация

КОМБИНАТОРНЫЕ МЕТОДЫ ИССЛЕДОВАНИЯ СПЕЦИАЛЬНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ И СИМВОЛЬНОЕ КОДИРОВАНИЕ

Аннотация. *В данной статье рассматривается концепция универсального комбинаторного кодирования. Большинство методов кодирования похожи и имеют множество общих черт,собрав которые в одно целое, можно получить универсальный метод кодирования. Это может стать мостом, который соединит известные методы кодирования, и послужить толчком к развитию технологии кодирования. Универсальное комбинаторное кодирование без потерь основано на комбинаторике. Оно не зависит от энтропии источника информации и разделено на три характеристические ветви кодирования. Это разделение на ветви помогло исследовать отношения между универсальным комбинаторным кодированием и другими методами. Для данного кодирующего метода в статье была сделана оценка эффективности. Универсальное комбинаторное кодирование имеет теоретическуюзначимость и практическую ценность применения.*

Ключевые слова: *символьное кодирование, алгоритм, шифрование, комбинаторные методы, универсальный метод кодирования, ординал.*

O. V. Kuzmin[1], I.A. Zelentsov[1]

[1]IrkutskState University, Irkutsk, the Russian Federation

COMBINATORY METHODS OF RESEARCH OF SPECIAL SEQUENCES AND SYMBOL CODING

Annotation. *In this paper, we consider the concept of universal combinatorial coding. Most coding methods that can be found in one whole, you can get a universal coding method. It can become a bridge that will connect the known coding methods and will serve as an impetus to the development of coding technology. Universal combinatorial lossless coding is based on combinatorics. It does not depend on the entropy of the information source and is divided into three characteristic coding branches. This division into branches helped to investigate the relationship between universal combinatorial coding and other methods. For this coding method, an efficiency estimate was made in the article. Universal combinatorial coding has theoretical significance and practical value of application.*

Keywords: *Symbolic coding, algorithm, encryption, combinatorial methods, universal coding method, ordinal.*

Введение

Одной из центральных задач в теории информации является эффективное кодирование источников. Это в числе прочего объясняется теоретической важностью задачи и ее разнообразными практическими приложениями к сжатию данных разного типа [1].

Главная идея комбинаторного метода заключается в задании множества кодируемых сообщений не посредством перечисления всех его элементов, а путем определения процедуры вычисления определенного номера для конкретного сообщения, для чего достаточно знания алфавита и таблицы частот. Для каждой последовательности данных ставиться строго определенный комбинаторный номер и обратно, каждому комбинаторному номеру соответствует строго определенная последовательность данных [2].

Теорию кодирования можно условно разделитьна три ветви: кодирование источника, канальное кодирование и кодирование, обеспечивающее секретность [3]. Основной задачей кодирования источника является сжатие данных, такое как кодирование Хаффмана [4], арифметическое кодирование [5, 6, 7, 8, 9] и символьное кодирование [10, 11, 12]. Канальное кодирование может улучшить надежность связи;примером могут служить коды с обнаружением и исправлением ошибок. Чтобы гарантировать безопасность информации при передаче, используется кодирование, обеспечивающее секретность. Обычно это достигается путем шифрования и дешифрования данных[13, 14, 15].

Как правило, каждый существующий метод кодирования относят только к однойиз трех указанных ветвей кодирования. Однако, существуют методы кодирования, которые можно отнести, по крайней мере, к двумветвям. Это позволяет предположить, что универсальный метод кодирования объективно существует. Подобный способ кодирования может отражать различные функции кодирования с разных точек зрения и стать мостом, соединяющим многие методы кодирования, что послужит толчком в развитии технологии кодирования.

1. Теория комбинаторного кодирования

Универсальное комбинаторное кодирование является своего рода способом кодирования без потерь. Понятие «универсальный» в данном контексте имеет три значения: во-первых, метод кодирования не зависит от стохастическойхарактеристики источника информации; во-вторых, метод мультихарактеристичен в силу своих свойств; в-третьих, метод универсален.

Предположим, что последовательность $a_1 a_2 ... a_n$ с n элементами будет закодирована с помощью m различных кодовых элементов; частота каждого элемента равнаn_i. Если n элементов упорядочены до полной перестановки в соответствии с эталонной последовательностью, то может быть сформировано новое пространство словаря, которое имеет строгий порядок перестановки последовательности. Проблема заключается в том, как вычислить положение последовательности в пространстве словаря.

Все перечисленные способы комбинаторного кодирования строятся на основе числа возможных сочетаний или биномиальных коэффициентов. Если j-ый элемент из любой последовательности, подобранной нами, совпадает с i-ым элементом исходной последовательности, то это означает, что текущая последовательность включает в себя$i-1$ элементов. Номер перестановки каждого элемента x в $i-1$-элементах может быть выражена через $S_{j,x}$, когда он занимает j-ю позицию. (Позиция этого элемента в элементе $i-1 = x$.) Это показано в (1). Рассмотрим

$$\begin{aligned} S_{j,x} &= C_{n-j}^{\omega_1} \cdot C_{n-j-\omega_1}^{\omega_2} \cdot C_{n-j-(\omega_1+\omega_2)}^{\omega_3} \\ &\cdot ... \cdot C_{n-j-(\omega_1+\omega_2+...+\omega_{x-1})}^{\omega_{x-1}} \cdot C_{n-j-(\omega_1+\omega_2+...+\omega_{x-1})}^{\omega_{x-1}} \cdot C_{n-j-(\omega_1+\omega_2+...+\omega_x)}^{\omega_{x+1}} \\ &\cdot ... \cdot C_{n-j-\sum_{q=1}^{i'-2}\omega_q}^{\omega_{i'-1}} \cdot C_{n-j-\sum_{q=1}^{i'-1}\omega_q}^{\omega_{i'-1}} \cdot \end{aligned} \quad (1)$$

Каждое ω_k в (1) представляет количество соответствующих элементов, «i'»– количество ключевых элементов, а число каждого элемента в этих ключевых элементах не равно 0. Равенство (1) включает многочисленные вычисления перестановок и комбинаций. Вычислительная эффективность слишком низкая, поэтому (1) можно преобразовать к виду

$$S_{j,x} = \frac{(n-j)!}{(\prod_{q=1}^{x-1}(\omega_x - 1)) \cdot (\omega_x - 1)! \cdot (\prod_{q=x+1}^{i'}(\omega_q !))}. \quad (2)$$

Дляj-го символа общее число последовательностей, включающих элементы $i-1$ передj-м символом, показано в (2).Рассмотрим сумму всех значения перестановки

$$\sum_{x=1}^{i-1} S_{j,x}, \quad (3)$$

тогда положение последовательности с n элементами в словаре (ординал) может быть вычислено в виде

$$o = \sum_{j=1}^{n} \sum_{x=1}^{i-1} S_{j,x}. \quad (4)$$

Основная идея декодирования – это предположение о том, что измеренное положение является определенным элементом, основанным на порядке эталонной последовательности. Соответствующее значение перестановок p_1 рассчитывается на основе предположения. Пусть p_1 сравним с ординалом, если ординал равен p_1 или больше p_1. Затем вычисляем значение перестановок (p_2) следующего элемента в соответствии с эталонной последовательностью. А чтобы вычислить элемент следующей позиции, новый ординал можно получить из равенства

$$o_{new} = o_{old} - (p_1 + p_2 + ... + p_{r-1}). \quad (5)$$

Когда значение ординала равно 0, алгоритм заканчивается. Если у ординала все еще есть элементы, то эти элементы могут быть заполнены проанализированной последовательностью согласно эталонной последовательности.

2. Оптимизированные или параллельные вычисления

Принятие комбинаторного метода для вычисления ординала имеет более высокую вычислительную сложность; процесс вычисления ординала может быть оптимизирован.

В процессе вычисления (3) каждое значение перестановки и комбинации в *i-1* элементах *j*-й соответствующей группы имеет пропорциональное отношение. То есть пропорция между каждой величиной перестановки, и первое значение перестановки и комбинации $C_{j,1}$ равно пропорции между номером соответствующего элемента и номером первого элемента. Тогда (3) можно оптимизировать:

$$\sum_{x=1}^{i-1} S_{j,x} = \frac{(C_{j,1} \cdot \sum_{k=1}^{j-1} \omega_k)}{\omega_1}. \quad (6)$$

На самом деле существует пропорциональное соотношение между перестановкой и значением комбинации первого элемента в группе *(j+1)*-й группы и значением перестановки первого элемента в *j*-й группе элементов. Это показано ниже:

$$\frac{C_{j+1,1}}{C_{j,1}} = \frac{\omega_i}{n'}. \quad (7)$$

В (7) ω_i – частота *j*-го символа. Этот символ является *i*-м элементом в таблице частот. Предположим, что число оставшихся элементов равно *n'–j* (где *n'* -номер элемента последовательности). $C_{1,1}$ частный случай, он может быть специально обработан. Возьмем $C_{0,1}$ как число комбинаций всех элементов для кодируемой последовательности. *n'* – количество всех элементов для последовательности, подлежащей кодированию. ω_i – номер первого элемента в эталонной последовательности.

Конечно, он также должен обрабатывать некоторые другие конкретные ситуации, например, для непрерывных пустых элементов, которые являются первым элементом в эталонной последовательности. В группе, соответствующей*(j+empty+1)*-й элемент, первое значение перестановки элементов может быть выражено как

$$C_{j+empty+1,1} = C_{j,1} \cdot \frac{t \cdot (t-1) \cdot ... \cdot (t-empty)}{n' \cdot (n'-1) \cdot ... \cdot (n'-empty)}. \quad (8)$$

В (8), *t* представляет собой номер первого элемента в эталонной последовательности, когда обрабатывается первый *j*-й шаг. Это текущее число первого элемента в эталонной последовательности. Когда «*empty*» равен 0, (8) эквивалентно (7).

В процессе вычисления первое значение перестановки, соответствующее первому элементу, вычисляется по всей величине перестановки *Max*. *Max* можно рассчитать

$$Max = \frac{n!}{n_1! \cdot n_2! \cdot n_m!}. \tag{9}$$

По методу пропорций, вычислительная скорость ординала будет значительно улучшена.

Процесс декодирования аналогичен.

Когда определяется значение n , чтобы ускорить вычислительную скорость *Max*. *Max* может быть рассчитан заранее, когда все n_i равны друг другу. В это время *Max* является самым большим и называется «*Whole*», тогда «*Whole*» хранится в файле. Когда он используется, его можно напрямую прочитать из документа. Затем соответствующий отрегулированный *Max*получается в соответствии с различным n_i в реальной последовательности. Очевидно, что «*Whole*» может быть рассчитан на основе

$$Whole = \frac{n!}{\left(\left(\frac{n}{m}\right)!\right)^m}. \tag{10}$$

Кроме того, он также может разбивать последовательность. Таким образом, порядковое число может быть вычислено параллельно, и эффективность расчета будет улучшена [16].

Графические и параллельные вычисления быстро развиваются, и их можно использовать в различных областях [17,18].Тысячи многопоточных процессов значительно ускорять вычислительную скорость универсального комбинаторного метода кодирования. Но существующие методы кодирования трудно использовать. Технологию параллельных вычислений с графическим процессором, такую как арифметическое кодирование или словарь кодирования. Поскольку в этой статье будет описано универсальное комбинаторное кодирование с точки зрения теории, параллельному методу GPU не нужно давать лишние детали. Среднее сравнение времени вычисления всего ординала на разных длинах между процессором и графическим процессором выполняется на рисунке 1.

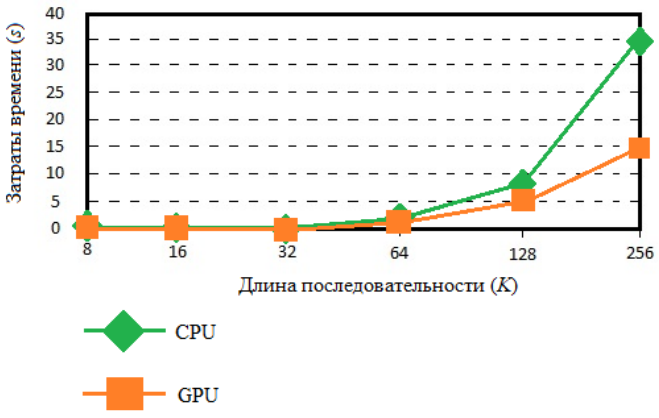

Рис.1. Среднее сравнение времени вычисления всего ординала на разных длинах между CPU и GPU.

На рисунке 1 видно, что наклон кривой последовательного вычисления ЦП больше, а наклон кривой параллельного вычисления графического процессора меньше. Этим объясняется, что при увеличении длины последовательности затраты на последовательное

вычисление ЦП растут быстро. Стоимость параллельных вычислений графического процессора растетмедленно. Когда длина последовательностей составляет 8 К, 16 К и 32 К, стоимость параллельного вычисления графического процессора больше, чем затраты времени на последовательные вычисления ЦП. По той причине, что, по мере того как длина последовательности не является достаточной длинной, время сохранения параллельных вычислений GPU не может компенсировать время, затрачиваемое на передачу данных между CPU и GPU. Когда длина последовательности равна 64 К, затраты времени на параллельные вычисления GPU становится меньше, чем затраты времени на последовательные вычисления ЦП. Это указывает на то, что время, затрачиваемое на параллельное вычисление графических процессоров, больше, чем затратное время, затрачиваемое на передачу данных между процессором и графическим процессором. С этого момента преимущество параллельных вычислений GPU все больше возрастает с длиной последовательности.

Увеличение ускорения показано на рисунке 2.

Рис.2. Ускорение параллельных вычислений GPU.

Согласно тенденции кривой на рисунке 2, чем длиннее длина последовательности, тем больше ускорение.

Очевидно, что при анализе эксперимента скорость целых порядковых вычислений может быть улучшена с помощью параллельной технологии GPU, а скорость работы увеличивается с увеличением длины последовательности.

3. Связь между универсальным комбинаторным кодированием и другими методами кодирования

Универсальное комбинаторное кодирование похоже на разные классические методы кодирования. Многие существующие методы кодирования являются производными от этих классических методов кодирования и совершенствуются в соответствии со специальными приложениями [19-21]. Универсальное комбинаторное кодирование можно разделить на три характеристические ветви и представить в виде: дерево кодирования, словарь кодирования и арифметическое кодирование. Рассмотрим эти представления по отдельности.

3.1. Кодовое дерево Хаффмана

Предположим, что существует последовательность: $a_1a_2...a_n$, которая состоит из m различных элементов и содержит n элементов. В этом случае $m = 2^v$ т. е. каждый элемент занимает v бит. Кодирующая последовательность тестов содержит m различных элементов с полной перестановкой. Очевидно, что существует $m!$ в контрольных последовательностей. После того, как последовательность подтверждается, а m дерево может быть сгенерировано, и положение $a_1a_2...a_n$ в этом m дерево можно подтвердить. Другие пути от корня дерева переходящие в его узел, а затем в его лист m представляют собой последовательности, которые имеют один и тот же код элементов $a_1a_2...a_n$, но порядок будет отличатся. Общий порядковый номер *–Max,* а порядковый номер начинается с 0. Самый большой порядковый номер *Max-1*.

Например, предположим, что существует последовательность *«bdaca»*. Число различных элементов равно 4. Очевидно, *v* равно 2. Число элементов последовательности тестов равно 4! =24. Это означает, что есть 24 дерева, которые представляют собой квадратное дерево со строгим порядком. Предположим, что эталонная последовательность *«abcd»*, тогда каждый элемент кода может быть представлен как: *a*: 00, *b*: 01, *c*: 10 и *d*: 11. Таким образом, последовательность может быть выражена как 0110001100 и занимать 10 бит. Если эта последовательность использует древовидную форму, то полученное дерево Хаффманаможно представить, как на рисунке 3.

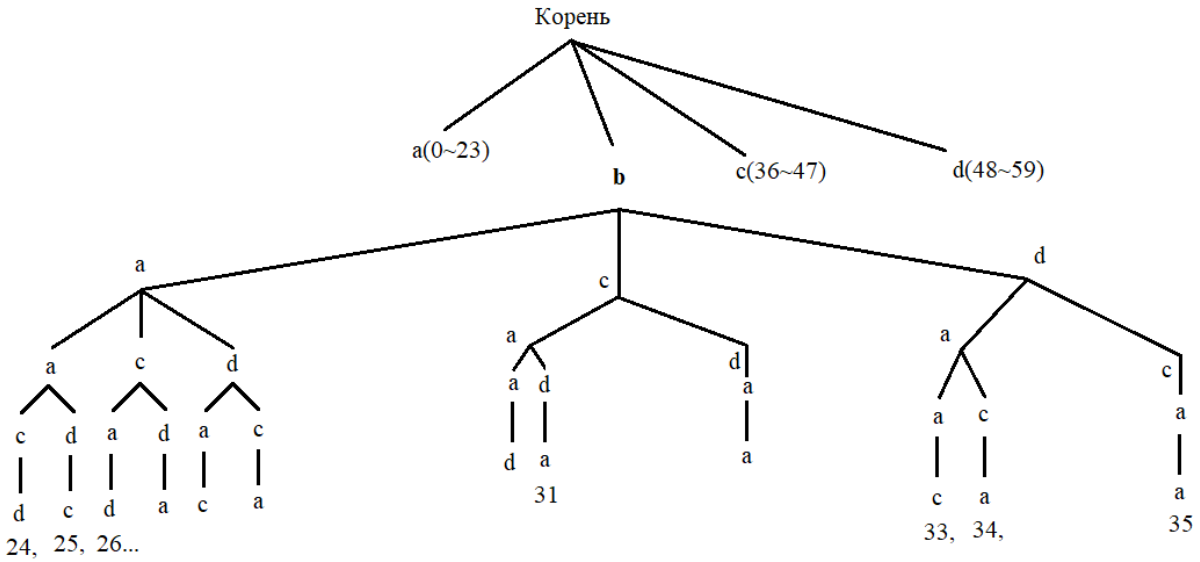

Рис.3. Квадратное дерево, которое имеет «bdaca».

Квадратное дерево на рисунке 3 состоит из последовательностей, в которых каждая последовательность содержит два *«a»*, один *«b»*, один *«c»* и один *«d»*. Сумма последовательностей $C_5^2 \cdot C_3^1 \cdot C_2^1 \cdot C_1^1 = 60$. Эти последовательности строго упорядочиваются в соответствии с порядком. Серийный номер от 0 до самого большого порядкового номера 59. Ординал, соответствующий последовательности *«bdaca»*, равен 34; этот ординал обозначает количество последовательностей перед *«bdaca»* и имеет один и тот же элемент кода, но имеет разное ранговое место. Таким образом, положение *«bdaca»* в квадратном дереве также может отражать свойство комбинации. Процесс декодирования будет выполнятся в обратном порядке.

Таким образом, комбинаторное кодирование может быть выражено как древовидная структура. Он вычисляет положение кодирующей последовательности в структуре*m*-дереваХаффмана на основе принципа комбинаторики.

3.2. Словарькодирования

Если упорядочить путь *m*-дерева Хаффмана в соответствии с порядком строго слева направо, может быть получен словарь. Его можно записать следующим образом: предположим, что существует последовательность: $a_1 a_2 ... a_n$, которая состоит из m разных элементов. В этом случае $m = 2^v$ и каждый элемент кода занимает *v* бит. Число присутствия элементов кода: $n_1 n_2 ... n_m$. Очевидно, $n_1 + n_2 + ... + n_m = n$. И эталонная последовательность состоит из m различных элементов кода, число эталонных последовательностей равна *m!*. После того, как тест последовательности подтверждается, пространство словаря может быть определено, и положение (номер ординала) $a_1 a_2 ... a_n$ словаре пространстве также определяется. Словарные пространство хранит все последовательности, которые имеют один и тот же код элемента и другой порядок перестановки с $a_1 a_2 ... a_n$. Число последовательностей можно рассчитать согласно (9).

Ординал кодирующей последовательности рассчитывается в соответствии с соответствующими уравнениями в следующих разделах. Все пространство словаря и положение кодирующей последовательности в словаре можно показать в виде таблицы 1.

Табл.1. Пространство словаря и положение кодированной последовательности.

Ординал	Последовательность
0	*aabcd*
1	*aabdc*
2	*aacbd*
…	...
33	*bdaac*
34	*bdaca*
35	*bdcaa*
…	...
57	*dcaab*
58	*dcaba*
59	*dcbaa*

Каждая последовательность словаря имеет строгий порядок с ограничениями эталонной последовательности. Конечно, в этом виде комбинационного словаря не существует. Он скрыт в частотной таблице последовательности. Это не требует фактического занятия пространства и времени. Позиция (номер ординала) кодирующей последовательности в словаре может быть рассчитана, а размер порядкового номера меньше размера пространства последовательности. Воспользовавшись этой характеристикой, универсальное комбинаторное кодирование может использоваться для сжатия данных. Словарное пространство универсального комбинаторного кодирования является фиксированным и объективно существует. Этот словарь содержит все последовательности, которые имеют одинаковый код с кодированной последовательностью.

3.3. Арифметическое кодирование

Универсальное комбинаторное кодирование использует абсолютное значение позиции для выражения последовательности в пространстве словаря, и это значение является целым числом; поэтому его можно рассматривать как арифметическое кодирование. Фактически, традиционное арифметическое кодирование выражает последовательность в виде цифры, которая является реальными данными 0 (нет данных) и 1 (есть данные).

Другим методом, который ближе к универсальному комбинаторному кодированию, является метод кодирования диапазона. Метод кодирования диапазона также можно рассматривать по существу, как арифметическое кодирование. Но метод кодирования диапазона должен иметь достаточно большое положительное целое число. Фактически, так называемое, достаточное большое положительное целое число играет роль самого большого порядкового числа, но оно недостаточно точное и оно больше фактических потребностей самого большого порядкового числа. Другими словами, порядковый номер универсального комбинаторного кодирования на самом деле является наименьшим и самым точным «положительным целым числом, которое достаточно велико».

Самое главное, что как кодирование диапазона, так и арифметическое кодирование 0-1 основаны на вероятности, и мы всегда предполагаем, что вероятность постоянна. Это неизбежно приведет к ошибке. Хуже того, во многих случаях элементы последовательности

не могут быть предугаданы. Конечно, адаптивное арифметическое кодирование может не зависеть от вероятности, но его коэффициент сжатия соответственно уменьшается.

Напротив, универсальное комбинаторное кодирование основано не на вероятности, а на частоте. Он может своевременно корректировать частоту в процессе вычислений в соответствии с реальной ситуацией. Таким образом, порядковый номер может быть обеспечен точностью. Конечно, недостатком является то, что значение частоты каждого элемента должно быть записано.

На рисунке 4 представлена схематичная диаграмма вычисленных результатов экземпляра *«bdaca»* в соответствии с комбинаторным кодированием, кодированием области (от 0 до 10000) и 0-1 арифметическим кодированием.

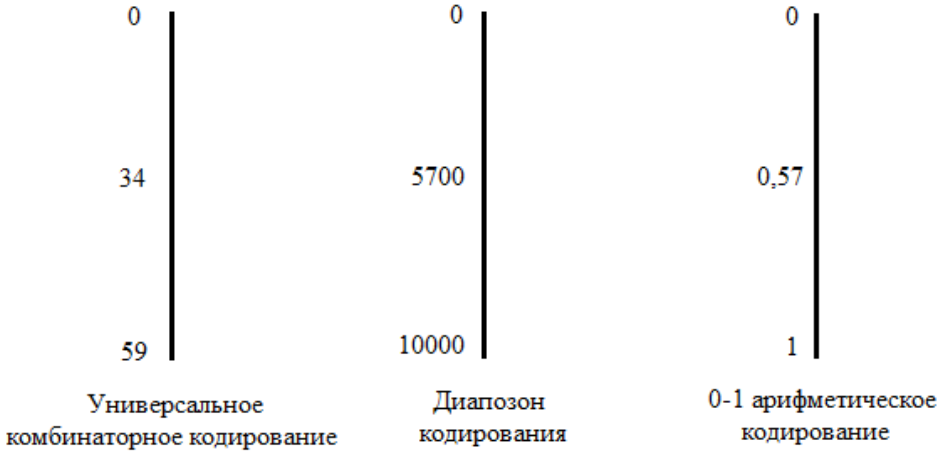

Рис.4. Результат каждого арифметического кодирования.

Таким образом, универсальное комбинаторное кодирование использует целочисленное значение для более точного представления последовательности, тогда как традиционные методы арифметического кодирования принимают относительное положение или аналогичную позицию для выражения последовательности. Фактически, если он принимает значение пропорции для представления относительного положения последовательности во всем пространстве словаря в универсальном комбинаторном кодировании, то пропорция аналогична результату 0-1 арифметического кодирования: $34/59 \approx 0{,}576$.

Комбинационная особенность универсального комбинаторного кодирования делает этот метод зависимым от многих других методов кодирования. Это означает, что метод имеет многокодовые возможности подключения. Это означает, что универсальное комбинаторное кодирование может стать инструментом для измерения характеристик различных методов кодирования.

4. Многократное применение универсального комбинаторного кодирования

Универсальное комбинаторное кодирование также имеет более специальные свойства помимо дерева кодирования Хаффмана, словаря кодирования и арифметического кодирования. Эти свойства делают универсальное комбинаторное кодирование применимым во многих отношениях.

4.1. Оценка размера ординала

Используя связь между всей порядковой длиной и характеристиками частот, размер ординала может быть предварительно оценен.

Текущие исследования показывают, чем больше разных частот среди символов в последовательности, тем меньше словарь, максимальный порядковый номер и средний порядковый номер. Когда разность частот между символами меньше 1, можно получить глобальный максимальный порядковый номер. Глобальный максимальный порядковый номер связан только с длиной последовательности. Размеры последовательности и всех видов ординалов удовлетворяют следующему неравенству:

$$len_{sequence} \geq len_{Whole} \geq len_{Max} \geq len_{ordinal}. \quad (11)$$

Длина (length) $len_{sequence} - len_{Whole}$ растет с ростом длины последовательности n, но увеличивающийся диапазон становится меньше. Это видно на рисунке 5.

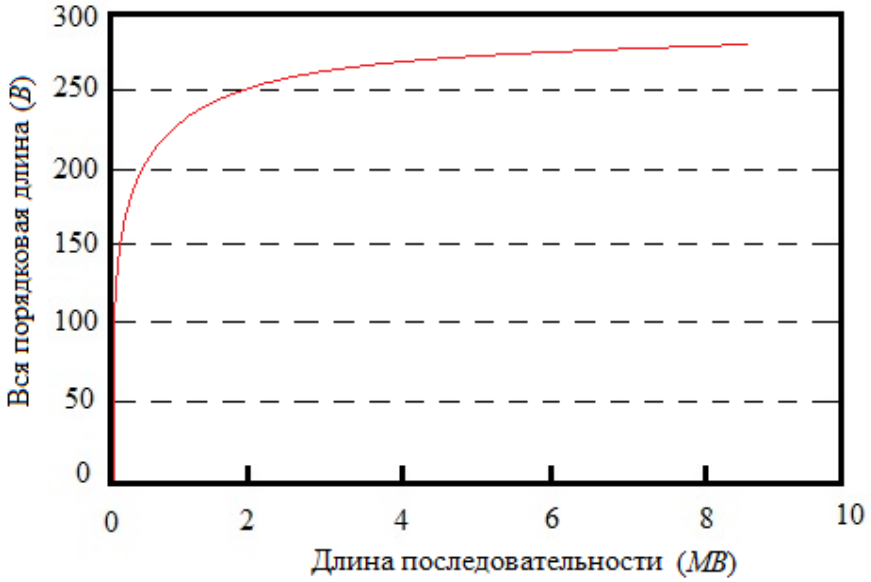

Рис. 5. Разница длины последовательности и всей порядковой длины.

Глобальный максимальный порядковый номер используется для предварительных оценок или теоретического анализа порядкового номера. Чтобы вычислить максимальный порядковый номер, он может быть предварительно рассчитан и помещен в файл. Максимальный порядковый номер может использоваться для вычисления порядкового номера последовательности, подлежащей кодированию.

4.2. Комбинаторное свойство сжатия

Порядковая длина должна быть меньше длины последовательности; это может быть использовано для комбинаторного сжатия.

Универсальное комбинаторное кодирование в основном использует порядковый номер для сокращения частотной избыточности. Он кодирует всю последовательность. Пространство *A* можно определить по длине кодирующей последовательности и преобразовать в меньшее пространство *B* через ограничение таблицы частот последовательности. Пространство *B* составлено из последовательности, которая имеет такую же таблицу кодирования частот с кодирующей последовательностью. Положение кодирующей последовательности в пространстве *B* меньше, чем позиция в пространстве *A*.Например, для последовательности *«bcada»* число различных элементов равно 4, каждый код занимает 2 бита (*a:* 00; *b:* 01; *c:* 10; d: 11). Предположим, что эталонная последовательность *«abcd»*, тогда позиция в пространстве *A* равна 0110001100 (396 в десятичной форме). После ограничения таблицей частот (*a:* 2; *b:* 1; *c:* 1; *d:* 1) создается пространство словаря *B*, а положение последовательности в пространстве *B* равно 34. Это можно показать на рисунке 6.

Пространство A

Последовательность	Двоичный код	Позиция
aaaaa	0000000000	0
aaaab	0000000001	1
aaaac	0000000010	2
aaaad	0000000011	3
.....	.....	...
.....	.....	...
bdaca	**01110010000**	**456**
.....	.....	...
.....	.....	...
ddddc	1111111110	1022
ddddd	1111111111	1023

Таблица частот

Код элемента	Частота
a	2
b	1
c	1
d	1

Пространство B

Последовательность	Ординал
a a b c d	0
a a b d c	1
a a c b d	2
.....	...
b d a c a	**34**
.....	...
d c a b a	58
d c b a a	59

Рис.6. Эталонная последовательность «abcd».

Из рисунка 6 видно, что пространство *B* через ограничение таблицы частот последовательности меньше исходного пространства *A*. В то же время среднее число позиций (номер ординала) кодирующей последовательности в пространстве *B* также становится меньше.

4.3. Комбинаторное шифрование

Пространство ключа шифрования представляет собой комбинацию различных элементов кода в эталонной последовательности (предположим, что каждая длина элемента кода равна *m*). Номер ключа шифрования в пространстве ключа шифрования равен ()!. Когда *m* достигает определенной длины, например, *m* равно 6, это может обеспечить конфиденциальное требование. В это время пространство ключа шифрования равно , а пространство ключа шифрования больше, чем текущий существующий алгоритм шифрования. (Они, как правило, используют в качестве бита блока шифрования [22-24]. Когда длина ключа равна 256 битам, ключ шифрования пространство равно). Он может достичь конфиденциального требования.

Расположение ключа в комбинированном кодировании принимает метод замены данных и положения. То есть ключ (*k+1*)-го раунда может быть выведен из ключа *k*-го раунда. Метод должен сначала выяснить данные *j* в *i*-ом местоположении (считая от 0) ключа *k*-го раунда. Затем возьмём данные *j* как данные *i*-го местоположения в ключ (*k+1*)-ых раундов. Чтобы предотвратить появление мертвой циркуляции местоположения и данных, результат (*k+1*)-го раунда ключа может быть смещен вправо. Наконец, финал (*k+1*)-го раунда. Это можно показать на рисунке 7 (для удобного описания возьмем $m = 4$).

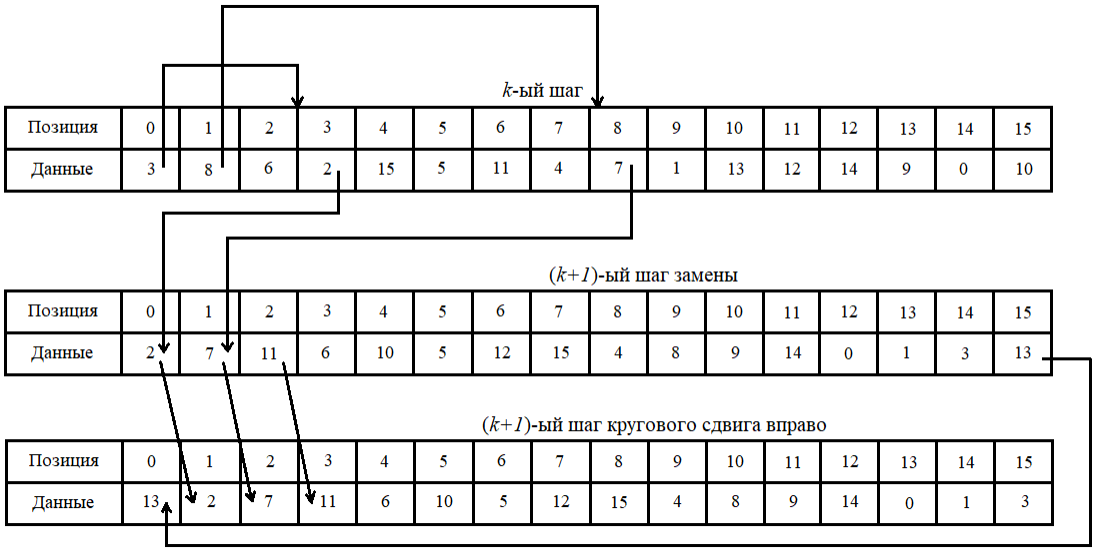

Рис.7. Метод замены данных и позиции.

Метод замены данных и позиции делает каждый ключ уникальными, и не только круглым ключом, но также и групповым ключом в каждом раунде. Метод создания группового ключа аналогичен методу создания круглого ключа, только поочередно выполняет правый переход к предыдущему ключу, а затем заменяет его между данными и позицией. На рисунке 8 показана взаимосвязь между основным ключом, круглым ключом и групповым ключом.

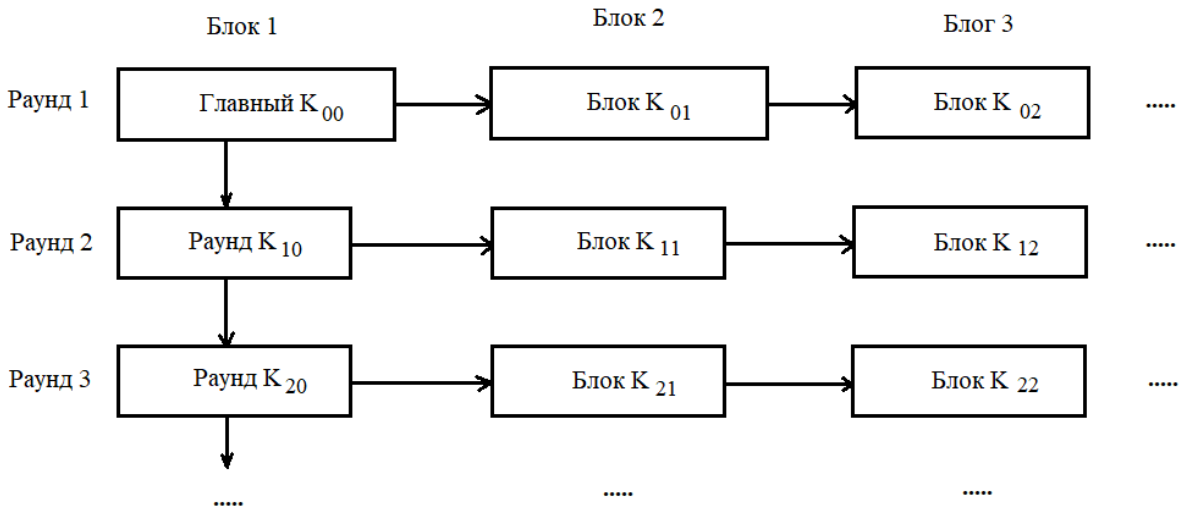

Рис.8.Связь всех видов ключей.

Метод генерации ключей комбинаторного шифрования делает: основной ключ, круглый ключ, а групповой ключ имеет такое же большое пространство; поэтому сложность расшифровки возрастает. Кроме того, комбинаторный метод шифрования использует элемент кода как блок обработки информации, а не бит, поэтому длина группы может быть более длинной, и она подходит для параллельных вычислений.

Ключевое пространство можно сравнить между комбинаторным методом шифрования и существующим методом шифрования (обычно они используют бит в качестве блока шифрования), как показано на рисунке 9. Для того, чтобы удобней выразить, есть только уровень экспоненты ключевого пространства, выраженный в оси *y* на рисунке 9.

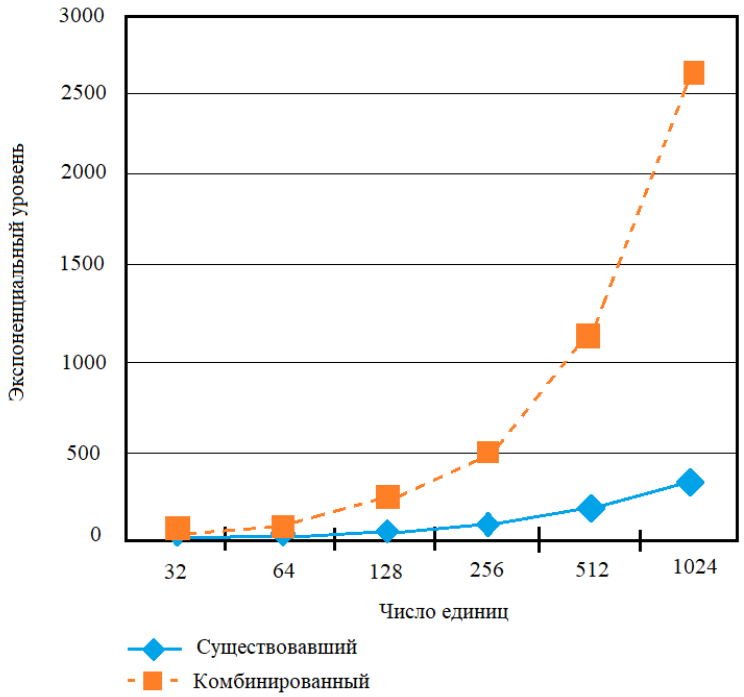

Рис. 9. Основное пространство различных методов шифрования.

На рисунке 9 видно, что ключевое пространство комбинаторного метода шифрования быстро расширяется вместе с увеличением битов элемента кода. Таким образом, безопасность увеличивается.

4.4. Другое применение комбинаторного кодирования

Универсальное комбинаторное кодирование также может выполнять простойанализданных, используя порядковый номер в качестве кода обнаружения ошибок. Он может использоваться для проверки данных передачи или хранения информации. Универсальное комбинаторное кодирование также может быть использовано для обработки абстрактной информации путем включения метода комбинаторной проверки и секретного ключа. В соответствии с ключевой последовательностью для последовательности можно выполнить несколько комбинационных вычислений. Другой подобный порядковый расчет может быть выполнен для ординала, полученного каждый раз, до тех пор, пока длина ординала не будет соответствовать требованиям пользователя. В этот момент последний порядковый номер используется в качестве основного. Конечно, время ординального расчета должно быть записано. Для метода информационных вычислений посредством универсального комбинаторного кодирования; отношение коллизии в теории очень мало.

5. Оценка эффективности универсального комбинаторного кодирования

Универсальное комбинаторное кодирование не зависит от статистических свойств источника информации, поэтому энтропия Шенонна не может использоваться для оценки качества кодирования для универсального комбинаторного кодирования. Но большинство методов оценки все еще используют вероятностную оценку. Поэтому в этой статье по-прежнему принимается во внимание оценка универсальной эффективности комбинаторного кодирования путем приблизительного расчета. Чтобы показать отличие от теории Шеннона, предположим, что источник является гладким без последовательности памяти из q элементов, а длина последовательности равна n. Кроме того, распределение вероятностей исходных символов равно $p_i (i = 1,...,q)$. Источник информации делится на $C(n)$ сегментов для обработки. Когда длина последовательности n и длина сегмента K очень велики, средняя длина кода может быть рассчитана следующими способами:

$$n = KC(n). \qquad (12)$$

Для каждого сегмента, длина которого равна K, число порядковых чисел может быть получено универсальным комбинаторным кодированием в виде следующей формулы:

$$N_k = \frac{K!}{\left(\prod (p_i K)!\right)}. \tag{13}$$

Таким образом, длина двоичного кода, занимаемого каждым порядком, равна

$$l = \left[\log N_k\right] = \left[\log\left(\frac{K!}{\left(\prod (p_i K)!\right)}\right)\right]. \tag{14}$$

Хотя частота каждого сегмента также занимает пространство, она очень мала по сравнению с пространством ординала. Таким образом, общая длина кода последовательности с n элементами равна

$$C(n)\left[\log\left(\frac{K!}{\left(\prod (p_i K)!\right)}\right)\right], \tag{15}$$

а средняя длина кода каждого символа источника, соответственно,

$$\bar{L} = \frac{\left(C(n)\left[\log(K!/(\prod p_i K)!)\right]\right)}{n}. \tag{16}$$

Из соотношения (16) следуют неравенства

$$\bar{L} \geq \frac{\left(C(n)\left[\log(K!/(\prod p_i K)!)\right]\right)}{n}, \bar{L} \prec \frac{\left(C(n)\left[\log(K!/(\prod p_i K)!)+1\right]\right)}{n}. \tag{17}$$

Из соотношения (12) и неравенства (17) следует

$$\frac{\log(K!/\left(\prod (p_i K)!\right))}{K} \leq \bar{L}, \bar{L} \prec \frac{(\log(K!/\left(\prod (p_i K)!\right))+1)}{K}. \tag{18}$$

После логарифмического расчета результаты $N_k = \frac{K!}{\prod((p_i)K)!}$ аналогичны результату $\frac{K!}{\prod((p_i)K)!}$. Произведя несложные вычисления, получим:

$$\begin{aligned}
&\log\left(\frac{K!}{(\prod (p_i K)!)}\right) \approx \log\left(\frac{K!}{(\prod p_i K)!)}\right) = \\
&\log K^K - \log \prod (p_i K)^{p_i K} = K \log K \\
&-(p_1 K \log(p_1 K) + p_2 K \log(p_2 K) + \ldots + p_q K \log(p_q K)) = \\
&K \log K - (K \sum p_i \log K + (p_1 + p_2 + \ldots + p_q) \\
&\cdot K \log K)) = -K \sum p_i \log p_i = KH(S).
\end{aligned} \tag{19}$$

Подставив (19) в неравенство (18), имеем

$$H(S) < \bar{L} < H(S) + \frac{1}{K}. \tag{20}$$

Когда K достаточно велико, средняя длина кода может быть представлена в следующем виде:

$$\bar{L} \approx H(S). \tag{21}$$

Выражение (21) показывает, что средняя длина кода в кодировании универсальной комбинации постепенно приближается к пределу энтропии источника, когда K достаточно велико.

Но следует отметить, что в приведенном выше процессе оценки есть приблизительный расчет, таким образом, имеет место приближенное равенство:

$$\log\left(\frac{K!}{(\prod(p_iK)!)}\right)\approx\log\left(\frac{K!}{(\prod(p_iK)!)}\right). \quad (22)$$

Таким образом, результат вычисления увеличивается. Это означает, что фактическая эффективность кодирования лучше, чем приблизительный результат вычисления.

Заключение

В данной статье рассматривается концепция универсального комбинаторного кодирования. Выполнена оценка эффективности метода кодирования. Универсальное комбинаторное кодирование было разделено на три характеристические ветви и исследовано по каждой их них. Универсальное комбинаторное кодирование имеет большие перспективы применения в таких процессах, как: обнаружение и исправление ошибок, шифрование и дешифрование, комбинаторное сжатие. Теоретические исследования показали, что при большом значении *K*средняя длина кода в универсальном комбинаторном кодировании близка к пределу энтропии источника,но фактическая эффективность кодирования должна быть лучше, чем приблизительные результаты расчета.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

[1] Бакулина М. П. Эффективный метод универсального комбинаторного кодирования / М. П. Бакулина // Ползуновский вестник, 2014, № 2. – С. 58-61.

[2] Гришин М.Л. Комбинаторное кодирование информации. URL.http://www. arts-union.ru/node/20

[3] FuZ., InformationTheoryandCoding. / Z. Fu, J. Zhao // Beijing, China: Publishing House of Electronics Industry; 2008.

[4] Huffman D. A. A method for the construction of minimum redundancy codes / D. A.Huffman // Proceedings of the Institute of Radio Engineers. 1952.40(9):1098–1101.

[5] Rissanen J. Arithmetic coding / J. Rissanen, G. G. Langdon // IBM Journal of Research and Development. 1979. 23(2):149–162.

[6] Кузьмин О.В.Фрактальные свойства бинарных матриц, построенных при помощи арифметики треугольника Паскаля, и помехоустойчивое кодирование /О. В. Кузьмин, Б. А.Старков // Современные технологии. Системный анализ. Моделирование. – 2016. № 4 (52). – С. 138-142.

[7] Кузьмин О.В. Бинарные матрицы, построенные при помощи треугольника Паскаля, и помехоустойчивое кодирование /О. В. Кузьмин, Б. А.Старков // Современные технологии. Системный анализ. Моделирование. – 2016. № 1 (49). – С. 112-117.

[8] Зеленцов И. А. Псевдослучайные последовательности и кодирование информации / И. А.Зеленцов // Вопросы естествознания, 2017, № 2 (14). – С. 30-37.

[9] Langdon G. G.Compression of black-white images with arithmetic coding / G. G. Langdon, J. Rissanen // IEEE Transactions on Communications Systems. 1981. 29(6):858–867.

[10] Ziv J. Universal algorithm for sequential data compression / J. Ziv, A. Lempel // IEEE Transactions on Information Theory. 1977. 23(3):337–343.

[11] Кузьмин О. В. Бинарные матрицы с арифметикой треугольника Паскаля и символьные последовательности / О. В. Кузьмин, Б. А. Старков // Известия Иркутского государственного университета. Серия «Математика». – 2016. – Т. 18. – С. 38–47.

[12] Ziv J. Compression of individual sequences via variable-rate coding / J. Ziv, A. Lempel // IEEE Transactions on Information Theory. 1978. IT-24(5):530–536.

[13] Кузьмин О. В. Анализалгоритмовдекодированиястандартарадиосвязи MIL-STD-186-141B / О. В. Кузьмин, А. А. Тимошенко // Вестник Иркутского государственного технического университета. – 2015, № 2 (97). – С.188–192.

[14] Кузьмин О. В. Введение в перечислительную комбинаторику / О. В.Кузьмин // – Иркутск: Изд-во Иркут.ун-та, 1995. – 112 с.

[15] Кузьмин О. В. Кодирование звуковой информации с помощью алгоритма перестановок / О. В. Кузьмин, И. А. Зеленцов // Современные технологии. Системный анализ. Моделирование. – 2017. № 4 (56). – С. 151-158.

[16] Jun L. Research on parallel technology within section in combinatorics coding / L. Jun, D-X. Liu // Proceedings of the International Conference on Computer Application and System Modeling (ICCASM '10). October 2010. IEEE Press. pp. 52–56.

[17] Shu Z. CUDA of GPU High Performance Computing / Z. Shu, Z. Yan-li // Beijing, China: China Water & Power Press; 2009.

[18] Sanders J. CUDA By Example: An Introduction to General-Purpose GPU Programming / J. Sanders, Ed. Sanders // Beijing, China: China Machine Press; 2011.

[19] Han Z. Parametric model for context-based adaptive binary arithmetic coding / Z. Han, K. Tang, H. Cui //Journal of Tsinghua University. 2009;49(4):531–534.

[20] Deng H-G. VQ image compression algorithm Based on Huffman Coding /H-G. Deng, S-W.Guo, Z-J. Li //Computer Engineering. 2010;36(4):218–219.

[21] Sun C. Research on the optimization of lossless compression algorithm for network transmission / C. Sun, G-X. Zhou // Journal of Hefei University of Technology. 2012;35(6):762–766.

[22] Yang H. A new block cipher based on chaotic map and group theory / H. Yang, X. Liao, K-W. Wong, W. Zhang, P. Wei //Chaos, Solitons and Fractals. 2009;40(1):50–59.

[23] Wei J. A new chaotic cryptosystem / J. Wei, X. Liao, K-W. Wong, T. Xiang // Chaos, Solitons and Fractals. 2006;30(5):1143–1152.

[24] Toldinas J. Energy efficiency comparison with cipher strength of AES and Rijndael cryptographic algorithms in mobile devices / J. Toldinas, V. Stuikys, R. Damasevicius, G. Ziberkas, M. Banionis // Electronics and Electrical Engineering. 2011. 108(2):11–14.

REFERENCES

[1] Bakulina M. P. Effektivnyj metod universal'nogo kombinatornogo kodirovaniya [*Effective method of universal combinatorial coding*]. Polzunovskij vestnik [*Polzunovsky Herald*]. , 2014, № 2. – C. 58-61.

[2] Grishin M.L. Kombinatornoe kodirovanie informacii [*Combinatorial coding of information*]. URL http://www. arts-union.ru/node/20

[3] Fu Z., Zhao J. Information Theory and Coding. Beijing, China: Publishing House of Electronics Industry; 2008.

[4] Huffman D. A. A method for the construction of minimum redundancy codes. Proceedings of the Institute of Radio Engineers. 1952.40(9):1098–1101.

[5] Rissanen J., Langdon G. G. Arithmetic coding. IBM Journal of Research and Development. 1979. 23(2):149–162.

[6] Kuz'min O. V., Starkov B. A. Fraktal'nye svojstva binarnyh matric, postroennyh pri pomoshchi arifmetiki treugol'nika Paskalya, i pomekhoustojchivoe kodirovanie [*Fractal properties of binary matrices constructed using Pascal's triangle arithmetic, and noise-immune coding*]. Sovremennye tekhnologii. Sistemnyj analiz. Modelirovanie [*Modern technologies. System analysis. Modeling*]. – 2016. № 4 (52). – C. 138-142.

[7] Kuz'min O.V., Starkov B. A. Binarnye matricy, postroennye pri pomoshchi treugol'nika Paskalya, i pomekhoustojchivoe kodirovanie [*Binary matrices constructed using the Pascal triangle, and noise-immune coding*]. Sovremennye tekhnologii. Sistemnyj analiz. Modelirovanie [*Modern technologies. System analysis. Modeling*]. – 2016. № 1 (49). – C. 112-117.

[8] Zelentsov I. A. Psevdosluchajnye posledovatel'nosti i kodirovanie informacii [Pseudo-random sequences and information coding]. Voprosy estestvoznaniya [Questions of natural science], 2017, № 2 (14). – C. 30-37.

[9] Langdon G. G., Rissanen J. Compression of black-white images with arithmetic coding. IEEE Transactions on Communications Systems. 1981. 29(6):858–867.

[10] Ziv J., Lempel A. Universal algorithm for sequential data compression. IEEE Transactions on Information Theory. 1977. 23(3):337–343.

[11] Kuz'min O. V., Starkov B. A. Binarnye matricy s arifmetikoj treugol'nika Paskalya i simvol'nye posledovatel'nosti [*Binary matrices with Pascal triangle arithmetic and character sequences*]. Izvestiya Irkutskogo gosudarstvennogo universiteta. Seriya «Matematika» [*News of Irkutsk State University. Series "Mathematics"*]. – 2016. – T. 18. – C. 38–47.

[12] Ziv J., Lempel A. Compression of individual sequences via variable-rate coding. IEEE Transactions on Information Theory. 1978. IT-24(5):530–536.

[13] Kuz'min O. V., Timoshenko A. A. Analiz algoritmov dekodirovaniya standarta radiosvyazi [*Analyze of the algorithms of decoding of the standard-radio connection*]. MIL-STD-186-141B. Vestnik Irkutskogo gosudarstvennogo tekhnicheskogo universiteta [*News of Irkutsk State University. Series "Mathematics"*]. – 2015, № 2 (97). – C.188–192.

[14] Kuz'min O. V. Vvedenie v perechislitel'nuyu kombinatoriku [*Introduction to enumerative combinatorics*]. – Irkutsk: Izd-vo Irkut.un-ta [*Irkutsk University Press*], 1995. – 112 c.

[15] Kuz'min O. V., Zelentsov I. A. Kodirovanie zvukovoj informacii s pomoshch'yu algoritma perestanovok [*Coding of sound information using the permutation algorithm*]. Sovremennye tekhnologii. Sistemnyj analiz. Modelirovanie. [*Modern technologies. System analysis. Modeling*]. – 2017. № 4 (56). – C. 151-158.

[16] Jun L., Liu D-X. Research on parallel technology within section in combinatorics coding. Proceedings of the International Conference on Computer Application and System Modeling (ICCASM '10). October 2010. IEEE Press. pp. 52–56.

[17] Shu Z., Yan-li Z. CUDA of GPU High Performance Computing. Beijing, China: China Water & Power Press; 2009.

[18] Sanders J., Sanders Ed. CUDA By Example: An Introduction to General-Purpose GPU Programming. Beijing, China: China Machine Press; 2011.

[19] Han Z., Tang K., Cui H. Parametric model for context-based adaptive binary arithmetic coding. Journal of Tsinghua University. 2009;49(4):531–534.

[20] Deng H-G., Guo S-W., Li Z-J. VQ image compression algorithm Based on Huffman Coding. Computer Engineering. 2010;36(4):218–219.

[21] Sun C., Zhou G-X. Research on the optimization of lossless compression algorithm for network transmission. Journal of Hefei University of Technology. 2012;35(6):762–766.

[22] Yang H., Liao X., Wong K-W., Zhang W., Wei P. A new block cipher based on chaotic map and group theory. Chaos, Solitons and Fractals. 2009;40(1):50–59.

[23] Wei J., Liao X., Wong K-W., Xiang T. A new chaotic cryptosystem. Chaos, Solitons and Fractals. 2006;30(5):1143–1152.

[24] Toldinas J., Stuikys V., Damasevicius R., Ziberkas G., Banionis M. Energy efficiency comparison with cipher strength of AES and Rijndael cryptographic algorithms in mobile devices. Electronics and Electrical Engineering. 2011. 108(2):11–14.

Информация об авторах

Олег Викторович Кузьмин– д. ф.-м. н., профессор, заведующий кафедрой теории вероятностей и дискретной математики. Иркутский государственный университет, г. Иркутск, quzminov@mail.ru.

Иван Александрович Зеленцов– магистрант ИМЭИ Иркутского государственного университета, г. Иркутск, izelentsov.isu@gmail.com

Authors

Oleg Viktorovich Kuzmin– Doctor of Physical and Mathematical Sciences. Professor, Head of the Department of Probability and Discrete Mathematics. Irkutsk State University, Irkutsk, quzminov@mail.ru.

Ivan Alexandrovich Zelentsov– Graduate student of the IMEI of Irkutsk State University, Irkutsk, izelentsov.isu@gmail.com

Для цитирования

Кузьмин О. В. Комбинаторные методы исследования специальных последовательностей и символьное кодирование / О. В. Кузьмин, И. А. Зеленцов // «Информационные технологии и математическое моделирование в управлении сложными системами»: электрон. науч. журн. – 2018. – №1. – С. 48-63 – Режим доступа: http://ismm-irgups.ru/toma/11-2018, свободный. – Загл. с экрана. – Яз. рус., англ. (дата обращения: 01.10.2018)

For citation

Kuz'min O. V., Zelentsov I. A. Kombinatornye metody issledovaniya special'nyh posledovatel'nostej i simvol'noe kodirovanie [Combinatorial methods for studying special sequences and character coding]. // Informacionnye tehnologii i matematicheskoe modelirovanie v upravlenii slozhnymi sistemami: ehlektronnyj nauchnyj zhurnal [Information technology and mathematical modeling in the man-agement of complex systems: electronic scientific journal], 2018. No. 1. P. 48-63. [Accessed 01/10/18]