УДК 681.3

Статистический алгоритм сжатия информации

С. В. Лобанов

В данной статье представлено описание алгоритма кодирования перестановок с повторениями, являющегося статистическим алгоритмом сжатия информации. Рассматриваются два варианта метода: двухпроходный, требующий априорно знания статистики кодируемой последовательности, и однопроходный, формирующий статистику в процессе работы. Показывается асимптотическая оптимальность алгоритма при увеличении длины последовательности. Приводится зависимость времени кодирования от длины сжимаемой последовательности. Сообщается о практических результатах моделирования алгоритма на ЭВМ.

Описание алгоритма

При удалении избыточности из сообщения, порождаемого дискретным источником информации, различают случаи, когда априорно известна или неизвестна его статистика. Рассматриваемый в данной статье алгоритм нумерационного кодирования (АНК) можно одинаково эффективно использовать в обоих случаях, т.к. имеются две модификации алгоритма, использующих одну концептуальную модель [1, 2, 3]. Первая модификация алгоритма требует перед кодированием наличия статистики сжимаемого блока данных, вторая ее не требует, а формирует по мере поступления элементов последовательности. В любом случае процедура кодирования состоит в том, что любой последовательности некоторой длины букв дискретного источника информации ставится в соответствие кодовое слово, состоящее из двух частей: в одной указывается состав последовательности – код состава (КС), в другой – номер последовательности с данным составом – код расположения (КР) [3]. АНК относится к группе алгоритмов без потери информации, т.е. является универсальным.

При нумерационном кодировании дискретный источник информации S задается как произвольное подмножество множества слов длины N ($N \geq 1$) в алфавите $X = \{x_1, x_2, \ldots, x_n\}$ ($n \geq 2$). Для описания алгоритма упорядочим как-либо буквы из алфавита X и для упрощения обозначений положим, что X={1,2,…,n}.

Любую последовательность букв дискретного источника информации

$\pi = \{x(1), x(2), \ldots, x(j), \ldots, x(N)\}$,

можно представить в виде перестановки составляющих ее элементов. Здесь $x(j)$ – элемент с порядковым номером x ($x \in \overline{1, n}$), стоящий на j-ой ($j \in \overline{1, N}$) позиции перестановки. Если в перестановке

π содержится N_i элементов с номером i ($i \in \overline{1,n}$), то, очевидно, выполняется следующее соотношение

$$\sum_{i=1}^{n} N_i = N, \qquad (1)$$

где N - количество элементов в последовательности (длина последовательности).

Общее число перестановок заданного состава N_i, т.е. мощность множества S, определяется известным в комбинаторике выражением для числа перестановок с повторениями

$$M = \frac{N!}{\prod_{i=1}^{n} N_i!} \qquad (2)$$

Несложно показать, что мощность источника S можно выразить через вероятности $P_i(j)$ появления символов алфавита на соответствующих позициях перестановки π следующим образом

$$M = \prod_{j=1}^{N} \frac{1}{P_{x(j)}(j)}$$

если определить, что $P_i(j) = L_i(j)/j$ или $P_i(j) = (N_i - R_i(j))/(N - j + 1)$, где $L_i(j)$, $R_i(j)$ - количество символов с порядковым номером i среди первых j и ($j-1$) элементов последовательности соответственно.

Следует отметить, что, т.к. приход одного из n символов алфавита является обязательным событием, то на каждом шаге кодирования они образуют полную группу несовместных событий с вероятностями $P_1(j), P_2(j), \ldots P_i(j), \ldots P_n(j)$. Следовательно, на каждой j-ой позиции ($j \in \overline{1,N}$) перестановки π соблюдается равенство

$$\sum_{i=1}^{n} P_i(j) = 1$$

Таким образом, математической моделью последовательности при кодировании по АНК является перестановка элементов 1,2,…,n с повторениями $N_1, N_2, \ldots, N_n$ раз соответственно. Это означает, что каждой перестановке из некоторого множества S надо поставить в соответствие ее номер из натурального ряда чисел $0 \div (M-1)$, где M- мощность множества S.

Вообще при фиксированной величине n алфавита дискретного источника информации и длине последовательности π равной N общее число групп перестановок с различным составом определяется по формуле комбинаторики для числа сочетаний с повторениями

$$f_n^N = C_{N+n-1}^{n-1} = C_{N+n-1}^{N} = \frac{(N+n-1)!}{(n-1)! \cdot N!}$$

В основу алгоритма нумерации перестановок заложен принцип цепного разбиения множества перестановок на подмножества по каждой из позиций перестановки. Процедура разбиения множества перестановок на подмножества, позволяющая установить взаимно-однозначное соответствие

между любой перестановкой π из множества S и ее номером из натурального ряда чисел $0 \div (M-1)$, представлена на рис.1, где имеют место следующие обозначения:

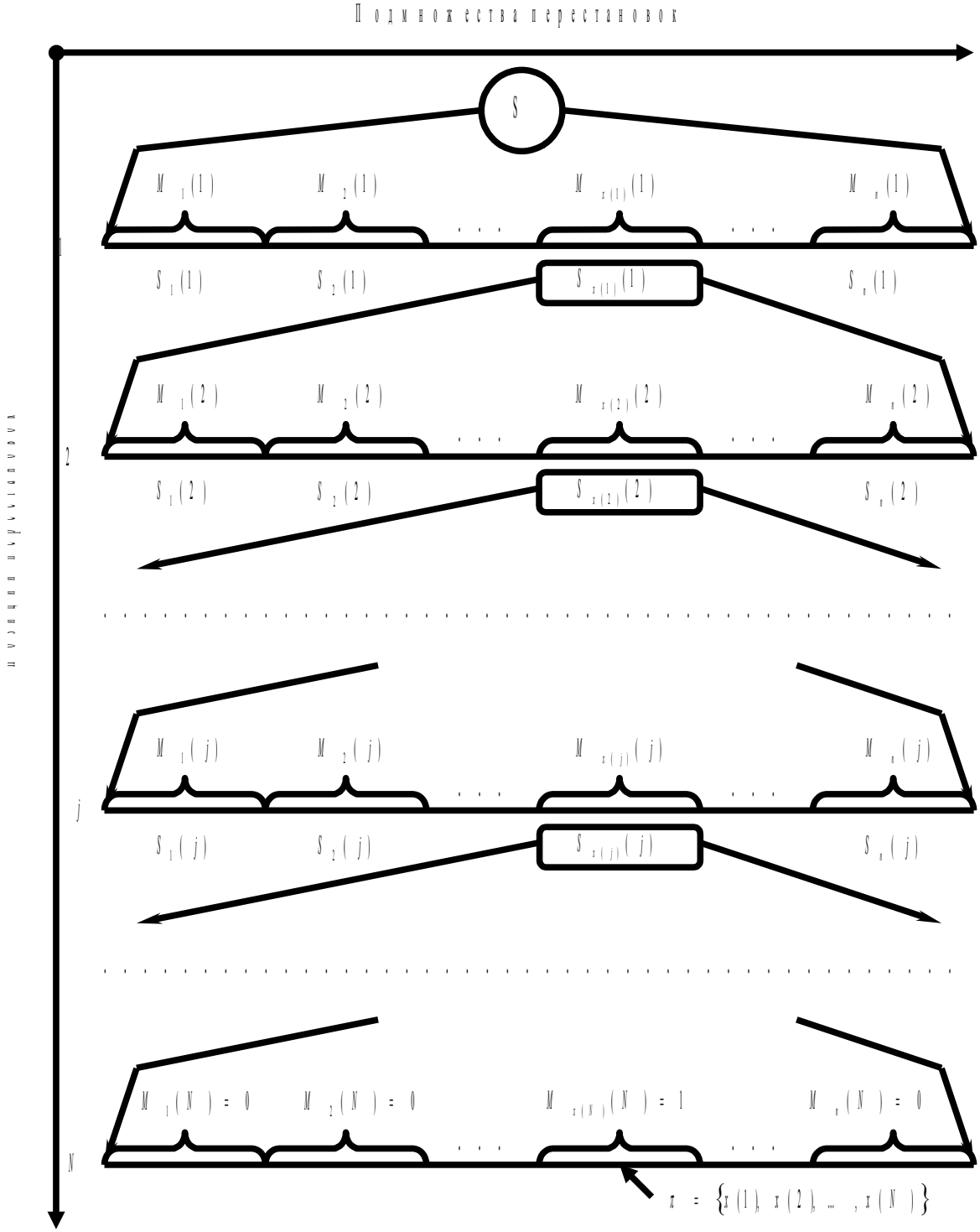

Рис. 1. Процедура разбиения множества перестановок на подмножества.

$M_i(1)$ - мощность *i*-го подмножества $S_i(1)$, полученного при разбиении исходного множества S по первой позиции перестановки;

$M_i(2)$ - мощность *i*-го подмножества $S_i(2)$, полученного при разбиении подмножества $S_{x(1)}(1)$ по второй позиции перестановки;

$M_i(j)$ - мощность *i*-го подмножества $S_i(j)$, полученного при разбиении подмножества $S_{x(j-1)}(j-1)$ по *j*-ой позиции перестановки.

Таким образом, *N* строк разбиения соответствуют *N* позициям перестановки π, причем каждая строка состоит из *n* подмножеств.

Для установления взаимно-однозначного соответствия между любой перестановкой π из множества *S* и ее номером *K(*π*)* из натурального ряда $0 \div (M-1)$ достаточно в качестве номера перестановки взять сумму мощностей всех подмножеств, предшествующих выделенным в разбиении по каждой из позиции [4]

$$K(\pi) = \sum_{j=1}^{N} \sum_{i=1}^{x(j)-1} M_i(j) \qquad (3)$$

Мощность $M_i(j)$ подмножества перестановок $S_i(j)$ равна произведению мощности $M(j)$ источника *S* на *j*-ом шаге кодирования на вероятность появления $P_i(j)$ на *j*-ой позиции последовательности элемента с *i*-ым порядковым номером

$$M_i(j) = M(j) \cdot P_i(j) \qquad (4)$$

Вообще дальше возможны два пути трактовки того, что из себя представляют подмножества $S_i(j)$. В зависимости от этого получаются разные выражения, определяющие $M(j)$, $P_i(j)$ и, как следствие, $K(\pi)$.

Первый вариант состоит в следующем. В подмножество $S_i(j)$ входят перестановки, у которых первые $(j-1)$ элементов совпадают с первыми $(j-1)$ элементами перестановки π, а на *j*-ой позиции стоит элемент x_i $(i = \overline{1,n})$. Следовательно, поскольку первые $(j-1)$ элементов перестановок подмножества $S_i(j)$ зафиксированны, то вероятность появления на *j*-ой позиции перестановки π элемента x_i определяется их количеством $N_i - R_i(j)$ среди всех $N-j+1$ оставшихся элементов, т. е.

$$P_i(j) = \frac{N_i - R_i(j)}{N-j+1}, \qquad (5)$$

где $R_i(j)$ - количество элементов с порядковым номером *i* среди первых $(j-1)$ элементов перестановки π. Мощность $M(j)$ источника *S* на *j*-ой позиции перестановки можно определить следующим образом

$$M(j) = \prod_{k=j}^{N} \frac{1}{P_{x(k)}(k)}$$

Подставив в $M(j)$ выражение для $P_i(j)$ из (5) получим формулу

$$M(j)=\prod_{k=j}^{N}\frac{N-k+1}{N_{x(k)}-R_{x(k)}(k)},$$

которой эквивалентна следующая форма записи

$$M(j)=\frac{(N-j+1)!}{\prod_{m=1}^{n}(N_m-R_m(j))!} \qquad (6)$$

в силу очевидного соотношения

$$\prod_{m=1}^{n}(N_m-R_m(j))!=\prod_{k=j}^{N}(N_{x(k)}-R_{x(k)}(k))$$

Вообще выражение (2) является частным случаем выражения (6) при *j*=1. Таким образом, подставляя в (4) выражения (5) и (6) получаем

$$M_i(j)=\frac{(N-j)!}{\prod_{m=1}^{n}(N_m-R_m(j))!}\cdot(N_i-R_i(j)) \qquad (7)$$

В итоге подстановкой $M_i(j)$ из (7) в (3) формируем окончательно соотношение для определения номера перестановки π

$$K(\pi)=\sum_{j=1}^{N}\left\{\frac{(N-j)!}{\prod_{m=1}^{n}(N_m-R_m(j))!}\right\}\sum_{i=1}^{x(j)-1}(N_i-R_i(j)) \qquad (8)$$

Такая схема кодирования обладает следующим недостатком: перед сжатием по выражению (8) необходимо знать априорно статистику последовательности, т.е. ее КС. Поэтому алгоритм является двухпроходным.

Возможен другой вариант АНК, который лишен этого недостатка, т.к. он формирует КС по мере поступления символов из входного потока, если определить подмножество $S_i(j)$ исходя из других соображений. Пусть в подмножество $S_i(j)$ входят перестановки, у которых последние $(N-j)$ элементов зафиксированы, а на *j*-ой позиции стоит элемент x_i $(i=\overline{1,n})$. Тогда вероятность появления на *j*-ой позиции последовательности *i*-го элемента алфавита равна

$$P_i(j)=\frac{L_i(j)}{j}, \qquad (9)$$

где $L_i(j)$ - количество элементов x_i среди первых *j* элементов перестановки π. Мощность $M(j)$ источника *S* на *j*-ой позиции перестановки можно определить следующим образом

$$M(j)=\prod_{k=1}^{j}\frac{1}{P_{x(k)}(k)}$$

Подставляя в предыдущее выражение значение $P_i(j)$ из (9), получим мощность множества *S* на *j*-ом шаге кодирования

$$M(j) = \prod_{k=1}^{j} \frac{k}{L_{x(k)}(k)}$$

Используя следующее тождество, справедливое для любой последовательности,

$$\prod_{m=1}^{n} L_m(j)! = \prod_{k=1}^{j} L_{x(k)}(k),$$

выражение для $M(j)$ запишем несколько иным образом

$$M(j) = \frac{j!}{\prod_{m=1}^{n} L_m(j)!} \qquad (10)$$

Следует обратить внимание, что выражение (2) является частным случаем соотношения (10) при *j=N*. Тогда мощность множества $S_i(j)$, определяемая по формуле (4), с учетом (9) и (10) будет следующей

$$M_i(j) = \frac{(j-1)!}{\prod_{m=1}^{n} L_m(j)!} \cdot L_i(j) \qquad (11)$$

В результате подставляя (11) в (3) окончательно получаем

$$K(\pi) = \sum_{j=1}^{N} \left\{ \frac{(j-1)!}{\prod_{m=1}^{n} L_m(j)!} \right\} \sum_{i=1}^{x(j)-1} L_i(j) \qquad (12)$$

Такая схема кодирования обладает следующим недостатком: восстановленная последовательность будет иметь порядок расположения элементов обратный по отношению к исходной. По данному варианту алгоритма написана программа на языке ассемблера для компьютеров типа IBM PC/AT, которая позволяет менять размер КР от 4 байт до 64 кБайт.

При реализации алгоритмов сжатия на ЭВМ очень удобно иметь рекуррентную форму записи процедуры кодирования и декодирования. Обе разновидности нумерационного кодирования имеют рекуррентную форму записи, которая, например, приведена в [1].

Процедура восстановления для двух приведенных выше вариантов нумерационного кодирования одинакова [3]. Она требует наличия перед декодированием наряду с $K(\pi)$, также КС последовательности. Следовательно, его необходимо передавать по каналу связи (сохранять на носителе информации) наряду с КР, что вводит дополнительную избыточность и уменьшает коэффициент сжатия алгоритма.

Асимптотическая оптимальность нумерационного кодирования

Покажем, что нумерационное кодирование асимптотически оптимально при увеличении длины последовательности *N*. Очевидно, что код будет оптимальным, если сумма удельных длин КС

и КР, приходящихся на один входной элемент, будет всего лишь на малую величину больше, чем энтропия H дискретного источника информации без памяти.

Исследуем сначала на оптимальность КР. КР является номером перестановки данного состава, общее число которых определяется формулой (2). Удельная длина КР, приходящаяся на один элемент последовательности (далее и везде, если не оговорено особо, подразумевается двоичный логарифм)

$$L_{1KP} = \frac{L_{KP}}{N} = \frac{\log M}{N} = \frac{1}{N} \cdot \log\left(N! \Big/ \prod_{i=1}^{n} N_i! \right) \qquad (13)$$

Воспользовавшись формулой Стирлинга вида

$\log N! = N \log N - N \log e + O(N)$,

где

$\log(\sqrt{2\pi N} \cdot e^{\frac{1}{12 \cdot N+1}}) < O(N) < \log(\sqrt{2\pi N} \cdot e^{\frac{1}{12 \cdot N}})$,

преобразуем выражение (13) к следующему виду

$$L_{1KP} = \log N - \log e + \frac{O(N)}{N} - \frac{1}{N} \sum_{i=1}^{n} (N_i \log N_i - N_i \log e + O(N_i))$$

Используя равенство (1) и проведя дополнительные преобразования, предыдущее выражение можем записать следующим образом

$$L_{1KP} = -\sum_{i=1}^{n} \frac{N_i}{N} \log \frac{N_i}{N} + \frac{O(N)}{N} - \frac{1}{N} \sum_{i=1}^{n} O(N_i) \qquad (14)$$

Можно показать, что

$$\lim_{N \to \infty} \frac{O(N)}{N} = \lim_{N \to \infty} \frac{1}{N} \sum_{i=1}^{n} O(N_i) = 0$$

Следовательно, L_{1KP} при $N \to \infty$ сходится по вероятности к энтропии источника

$$H = -\sum_{i=1}^{n} \frac{N_i}{N} \log \frac{N_i}{N}$$

Значит КР является оптимальным.

Теперь рассмотрим оптимальность КС. При любом способе кодирования существует некоторый максимальный размер $N_{\max}$ последовательности, при котором происходит заполнение разрядной сетки КС в силу ее конечности. Пусть КС формируется простейшим способом, а именно: кодирование состава сводится к подсчету букв каждого типа в общем случае на n счетчиках разрядности $\log N_{\max}$ каждый. Такой способ формирования КС наиболее простой, но не самый оптимальный. Следовательно относительная длина КС, приходящаяся на один элемент входной последовательности

$$L_{1KC} = \frac{n}{N} \cdot \log N_{\max} \qquad (15)$$

Принимая во внимание, что $N_{\max} \le N$ и $n << N_{\max}$, получим

$$\lim_{N\to\infty} L_{1KC} = 0$$

Таким образом, описанный выше способ нумерационного кодирования является асимптотически оптимальным, т.к. удельная длина кода

$$L_1 = L_{1KP} + L_{1KC} = H$$

при $N \to \infty$.

Зависимость времени кодирования от длины последовательности

Время, затраченное на кодирование по алгоритму АНК последовательности длиной *N,* равно

$$T = \sum_{j=1}^{N} t_j , \qquad (16)$$

где t_j - время обработки элемента *x*(*j*), стоящего на *j*-ой позиции перестановки π. Время обработки элемента *x*(*j*) на каждом отдельном шаге *j* прямо пропорционально длине КР на этом шаге

$$t_j = k \cdot L_{KP}(j) \qquad (17)$$

Коэффициент пропорциональности *k* имеет размерность сек/бит. Он показывает, сколько времени затрачивается на обработку одного бита КР. Длина КР на шаге *j* определяется как

$$L_{KP}(j) = \log\left(\frac{j!}{\prod_{i=1}^{n} L_i(j)!} \right)$$

Используя (14) можем записать

$$L_{KP}(j) \approx -\sum_{i=1}^{n} L_i(j) \cdot \log \frac{L_i(j)}{j} = j \cdot H_j$$

Здесь

$$H_j = -\sum_{i=1}^{n} \frac{L_i(j)}{j} \log \frac{L_i(j)}{j}$$

энтропия источника информации на *j*-й позиции последовательности. Примем, что энтропия источника не меняется от шага к шагу, т.е. постоянна и равна *H*. Тогда относительная длина КР

$$L_{KP}(j) = j \cdot H \qquad (18)$$

Подставляя (18) в (17) и затем в (16) получим, что

$$T = k \cdot H \cdot \sum_{j=1}^{N} j$$

Сумма арифметической прогрессии

$$\sum_{j=1}^{N} j = \frac{1}{2} \cdot N \cdot (N+1)$$

Таким образом, время кодирования всей перестановки

$$T = \frac{1}{2} \cdot k \cdot H \cdot N \cdot (N+1) \qquad (19)$$

пропорционально квадрату ее длины *N*.

При декодировании характер зависимости остается тот же. Изменяется лишь коэффициент пропорциональности *k*. К настоящему времени разработаны программы нумерационных кодера и декодера на языке ассемблера для персонального компьютера типа IBM PC, позволяющие менять размер КР от 4 байт до 64 Кбайт. Для разработанных программ коэффициент *k* при кодировании равен $1.289 \cdot 10^{-8}$ сек/бит, а при декодировании $2.093 \cdot 10^{-8}$ сек/бит (AMD-K6 200 МГц). Необходимо отметить, что для программ, написанных на разных языках программирования и для разных операционных систем, будет меняться только коэффициент *k*, но не общая зависимость времени кодирования *T* от длины последовательности *N*.

Очевидно, что одну и ту же последовательность длиной *N* можно кодировать разными способами: либо обрабатывать ее целиком, либо разбить на ϑ равных участков длиной N / ϑ и каждый из них обрабатывать отдельно. В последнем случае представляет интерес оптимальное число ϑ, $\vartheta = \overline{1, N}$, таких участков, при котором время, затраченное на кодирование всей перестановки длиной *N*, будет минимальным.

Попробуем определить оптимальное значение ϑ или покажем, что его не существует. Действительно, время обработки всей последовательности длиной *N* является суммой времен обработки каждого из ϑ интервалов, т.е.

$$T(\vartheta) = \sum_{i=1}^{\vartheta} t_i \qquad (20)$$

Время обработки каждого отдельного интервала согласно (19)

$$t_i = \frac{1}{2} \cdot k \cdot H \cdot \frac{N}{\vartheta} \cdot \left(\frac{N}{\vartheta} + 1 \right) \qquad (21)$$

Подставляя (21) в (20) находим

$$T(\vartheta) = \frac{1}{2} \cdot k \cdot H \cdot N \cdot \left(\frac{N}{\vartheta} + 1 \right) \qquad (22)$$

Как видно из полученного выражения оптимального значения ϑ не существует. Время кодирования перестановки максимально при ϑ=1 и минимально при ϑ=*N*. В последнем случае зависимость между временем кодирования и длиной последовательности становится линейной, однако исчезает эффект сжатия информации.

Заключение

Оптимальные свойства алгоритма начинают проявляться при достаточно длинной последовательности, а, следовательно, большой величине КР. Но при этом сильно возрастает время обработки. Например, при размере КР равном 64 Кбайт на компьютере с процессором INTEL PENTIUM 133 МГц время, затрачиваемое на кодирование, составляет 14 минут, а на декодирование 23 минуты (при длине последовательности 112 тыс. байт, текст на русском языке). Количество перестановок с повторениями в этом случае измеряется двоичным числом, занимающим 64 Кбайт. В связи с этим широкое распространение этого алгоритма в системах сжатия информации весьма проблематично.

Другой проблемой данного алгоритма является то, что для распаковки последовательности наряду с КР необходим также КС. При небольших размерах сжимаемого блока данных это приводит к неэффективности использования данного алгоритма по сравнению с адаптивными версиями Хаффмановского и арифметического кодеров.

Список литературы

1. С. И. Ватутин, О. В. Ивахив, И. Д. Калашников, Б. В. Рощин. Алгоритм нумерации перестановок с повторениями. / Моск. авиац. ин-т. – Деп. в НИИЭИР. - 12.04.1978, №3-5621. - 13 с.

2. С. И. Ватутин, О. В. Ивахив, Б. В. Рощин, И. Д. Калашников. Асимптотическая оптимальность нумерационного кодирования источника. / Моск. авиац. ин-т. – Деп. в НИИЭИР. – 19.06.1978, №3-5622. - 13 с.

3. С. И. Ватутин, О. В. Ивахив, Б. В. Рощин. Нумерационное кодирование источника равномерным блочным кодом. / Моск. авиац. ин-т. – Деп. в НИИЭИР. – 27.05.1978, №3-6293. - 6 с.

4. Б. Я. Рябко. Эффективный метод кодирования источников информации, использующий алгоритм быстрого умножения. // Проблемы передачи информации. - 1995, т.№31, №1. - с.3-12.

Лобанов Сергей Викторович, аспирант кафедры 402 радиосистем управления и передачи информации Московского государственного авиационного института (технического университета); Телефон: 904-20-36, e-mail: sergey@degunino.net