
УДК:681.324

А. Ш. Сулейманов, канд. техн. наук
Азербайджанский технический университет
(Азербайджан, AZ1073, Баку, просп. Г. Джавида, 25,
тел.: (09412) 4383260, E-mail: akif@inbox.ru)

Алгоритмы компрессии произвольной двоично-кодированной информации, основанные на применении логических шкал

(Статью представил д-р техн. наук Ю. М. Коростиль)

Рассмотрены компьютерные алгоритмы универсальной схемы сжатия (компрессии) и однозначного развертывания (декомпрессии) произвольной двоично-кодированной информации. Предложены методы сжатия, основанные на статистических данных с применением логических шкал позиционного кодирования, вносящих единообразие в процессы сжатия и развертывания, что позволяет разрабатывать относительно универсальные эффективные средства сжатия данных.

Розглянуто комп'ютерні алгоритми універсальної схеми стискування (компресії) та однозначного розгортання (декомпресії) довільної двоїчно-кодованої інформації. Запропоновано методи стискування, що базуються на статистичних даних з використанням логічних шкал позиційного кодування, які вносять однаковість у процеси стискування та розгортання, що дозволяє розроблювати відносно універсальні ефективні засоби стискування даних.

К л ю ч е в ы е с л о в а: сжатие данных, компрессия (декомпрессия), кодирование, избыточность.

Согласно большинству прогнозов в ближайшем будущем в технически развитых странах основная масса информации в полном объеме будет храниться в безбумажном виде — в памяти компьютерных систем и сетей [1]. Подобные изменения способны выдвинуть на передний план то или иное научное направление или техническое решение. Одним из таких направлений является сжатие дискретных данных, которое занимает одну из ведущих позиций в безбумажной информатике. В настоящее время это весьма актуально, так как информационные ресурсы становятся основным национальным богатством, а эффективность их промышленной эксплуатации все в большей степени определяет экономическое развитие в целом.

В [1—5] показано, что дальнейшие исследования в области компрессии данных должны концентрироваться вокруг следующих направлений:

повышение эффективности сжатия;
увеличение скорости работы алгоритма;
осуществление сжатия на основе новой системы контекстов.

Постановка задачи. Существуют два основных способа сжатия: статистический и словарный. Методы Шеннона—Фэнно, Хаффмена и арифметическое кодирование относятся к статистическим методам. Предложенные алгоритмы (RPOS и RGRUP) также относятся к статистическим методам. В статистическом сжатии каждому символу присваивается код, основанный на вероятности его появления в тексте, которая определяется по предварительному статистическому анализу (СТА).

Лучшие словарные методы используются в алгоритмах Зива—Лемпела (LZ-семейство) LZ77 и LZ78. Это алгоритмы сжатия без потерь, опубликованные в работах [6, 7].

Эти два алгоритма являются наиболее известными в семействе LZ*, куда также включены LZW, LZSS, LZMA и другие алгоритмы (почти все практические словарные кодировщики принадлежат семье алгоритмов, разработанных Я. Зивом и А. Лемпелом.) Эти алгоритмы можно быстро реализовать, но они не обязательно оптимальны, поскольку не выполняют никакого анализа входных данных. В словарном методе группы последовательных символов или «фраз» заменяются кодом. Замененная фраза может быть найдена в некотором «словаре». Как видно, словарные методы основаны на применении только определенных закономерностей языка первичного представления информации.

Словарные алгоритмы более практичны. Их преимущество перед статистическими методами теоретически объясняется тем, что они позволяют кодировать последовательности символов разной длины. Статистические алгоритмы тоже можно использовать для таких последовательностей, но в этом случае их реализация становится весьма ресурсоемкой.

Ключевым для размера получаемых кодов N является размер словаря K во фразах, потому что каждый код при кодировании по методу LZ78 содержит номер фразы в словаре. Из последнего следует, что эти коды имеют постоянную длину, равную округленному в большую сторону двоичному логарифму размера словаря $+8$ (это количество бит в байт-коде расширенного ASCII) $N = \log K + 8$.

В табл. 1 для сравнения приведены данные, полученные с помощью предложенного метода и алгоритма LZ*.

Как видно из сопоставительного анализа LZ* методов компрессии данных, они являются в определенной степени проблемно-ориентированными (специализированными). В основу алгоритмов реализации этих методов положены семантические, лингвистические, структурные, вероят-

ностно-статистические и иные характеристики языка первичного представления информации (системы работают полностью на лексическом уровне). Поэтому они не отвечают требованиям систем обработки, хранения и передачи данных и управления. Кроме того, отсутствует четкая теоретическая база системного решения данной задачи.

Предложенные методы сжатия информации, основаны на концепции применения абстрактного множества (алфавита) двоичных квантов определенной длины (3, 4, 5, ..., 8 битов) и логических шкал позиционного кодирования. Применение алфавита двоичных квантов с различными длинами без учета структуры его построения и семантики, в отличие от алгоритма Лемпеля—Зива, дает возможность найти общую теоретическую базу компрессии данных для их относительной универсализации, функциональной полноты и прагматической ценности.

Предложенный алгоритм может быть применен повторно после первичного сжатия методом LZ*. Возможна также декомпозиция файлов на основе его статистических данных на более эффективные с точки зрения сжатия подфайлы с применением логической шкалы позиционного кодирования (ЛШПК). Кроме того, эти алгоритмы можно использовать в системах анализа текстов для структурного представления [8, 9].

Алгоритмы сжатия и однозначного развертывания дискретных данных. Из сопоставительного анализа наиболее распространенных методов и алгоритмов компрессии следует, что для системного решения задачи

Таблица 1

Метод сжатия (характеристика)	LZ*	RPOS и RGRUP
Тип алгоритма	Словарный	Статистический
Предварительный СТА	Отсутствует	Для предварительной оценки выполняется СТА
Входные файлы	Язык первичного представления информации	Абстрактные множества (алфавиты) двоичных квантов
Время реализации	Сравнительно быстрое	Медленное (для увеличения требуются аппаратные расходы и параллелизм обработки)
Область применения	Относительно специализированные	Имеется возможность универсализации
Длина сжатых кодов	Постоянная	Переменная
Словесное (буквенное) сжатие	Возможно	Возможно (использование в системах анализа текстов для структурного представления)
Множественность сжатия	Возможность ограничена	Возможно

сжатия данных (для их относительной универсализации) более эффективны методы, основанные на применении ЛШПК [10, 11]. Это объясняется их следующими характерными особенностями:

1. Применение статистических характеристик (частоты встречаемости букв) алфавитов сжатия позволяет получить значительный эффект от сжатия.

2. Возможность применения в качестве алфавита сжатия не только символов (букв алфавита) языка первичного представления информации, но и битовых групп различной длины (квантов).

3. Возможность разбить файл, содержащий любую информацию, на подфайлы для эффективного применения методов сжатия; декомпозиция файла повышает вероятность получения эффекта от сжатия и увеличивает степень секретности его хранения или передачи по коммуникационным каналам.

4. Возможность унифицировать основные процедуры обработки данных для разработки модульных алгоритмов и аппаратно-программных средств сжатия и развертывания с возможной программируемой коммутацией последних.

Рассмотрим два метода сжатия данных.

Первый — метод последовательного исключения букв (RPOS-преобразование) — основан на применении локальных статистических характеристик (частота встречаемости каждого кванта-буквы) сегмента данных, который рассматривается как стохастическая последовательность битов, а сжатие (преобразование) проводится последовательным исключением из обрабатываемого сегмента соответствующих букв алфавита.

Второй — метод исключения группы букв (RGRUP-преобразование) — также основан на применении локальных статистических характеристик сегмента данных.

Однако сжатие происходит путем выделения и фиксации посредством ЛШПК группы букв, обладающих сравнительно большой избыточностью.

Оба метода основаны на технологии применения ЛШПК, которая позволяет проводить последовательную декомпозицию файла на эффективные подфайлы и создавать параллельные алгоритмы сжатия (развертывания).

Математическая модель алгоритма. Введем некоторые основные понятия и определения. Сегмент стохастической последовательности битов назовем двоичным кортежем и представим в виде

$$M_0 = (x_1, x_2, \dots, x_q).$$

Здесь $x_q \in M$, где $M = \{0, 1\}$; q — количество битов (символов) стохастической последовательности.

Обозначим непрерывную конечную последовательность битов длиной z через m_z и назовем эту величину шагом квантования:

$$m_z = (y_1, y_2, \dots, y_z).$$

Значения m_z и q необходимо выбрать так, чтобы q / m_z было целым числом:

$$K_z = q / m_z,$$

где z — максимальная длина шага квантования в битах. После квантования M_0 шагом квантования m_z получим K_z -компонентный кортеж

$$M'_0(m_z) = (M_1^{(m_z)}, M_2^{(m_z)}, \dots, M_{K_z}^{(m_z)}).$$

Предположим, что какие-то компоненты кортежа $M'_0(m_z)$ отличаются определенным свойством. Множество этих компонентов обозначим через $A^\tau(m_z)$. Тогда ЛШПК(m_z) можно выделить те компоненты кортежа $M'_0(m_z)$, которые входят в $A^\tau(m_z)$:

$$\text{ЛШПК}(m_z) = (\alpha_1^{(\tau)}, \alpha_2^{(\tau)}, \dots, \alpha_{K_z}^{(\tau)}),$$

$$np_j M'_0 = (M_j^{(m_z)}), j = 1, 2, \dots, K_z,$$

$$\alpha_j^{(\tau)} = \begin{cases} 1, & \text{если } M_j^{(m_z)} \in A^{(\tau)}(m_z), \\ 0, & \text{если } M_j^{(m_z)} \notin A^{(\tau)}(m_z). \end{cases}$$

Здесь τ — номер этапа разбиения компонент кортежа на два непересекающихся характеристических множества. Естественно, что при квантовании M_0 шагом квантования m_z количество различных m_z -битных букв (назовем множество этих букв алфавитом преобразования, в данном случае — сжатия) будет 2^{m_z} . Тогда один и тот же кортеж M_0 можно квантовать $(z-1)$ различными квантами (поскольку после квантования двоичного кортежа шагом $z=1$ получаем исходный кортеж M_0 , эта операция не имеет смысла) и получить $(z-1)$ различных множеств алфавита сжатия.

Обозначим конкретную букву определенного алфавита через

$$\beta_j^*(m_z) \in \{\beta_{j(m_z)}(m_z)\}, j(m_z) = 0, 1, 2, \dots, (2^{m_z} - 1)u_{-j}^* \in j(m_z).$$

Естественно, что конкретный двоичный кортеж при квантовании шагом квантования m_z может содержать не все разнообразные буквы $\beta_{j(m_z)}(m_z)$ в количестве 2^{m_z} , и локальная частота их встречаемости будет для каждого алфавита разной. Частоту буквы $\beta_j^*(m_z)$ обозначим через $\eta_j^*(m_z)$. Тогда после квантования исходного кортежа M_0 шагом квантова-

ния m_z образуем две упорядоченные последовательности (для простоты записи не будем отмечать шаг квантования m_z):

$$\eta_{j_1} \geq \eta_{j_2} \geq \eta_{j_3} \geq \dots \geq \eta_{j_Q}, \quad (1)$$

$$\beta_{j_1}, \beta_{j_2}, \beta_{j_3}, \dots, \beta_{j_Q}, \quad (2)$$

где $j_1, j_2, j_3, \dots, j_Q \in \{0, 1, 2, \dots, (2^{m_z} - 1)\}$.

Сжатие двоичного кортежа назовем R -преобразованием. Будем пользоваться обозначением RX , где X — отображает мнемонику конкретного применяемого метода компрессии данных. Оба рассматриваемых метода основаны на вероятностно-статистических характеристиках букв алфавита сжатия. Общим для обоих методов является формирование последовательностей (1) и (2), т. е. определение частоты встречаемости каждой буквы ($\eta_j^*(m_z)$) и упорядочение, а также определение буквы сжатия $\beta_j^*(m_z)$.

Эта процедура называется «статистический анализ файла и упорядочение частоты встречаемости букв» и имеет следующие особенности:

1. Переменная длина шага квантования обуславливает особые требования к длине исходного файла. Длина файла в битах должна быть нацело делимой на все принятые значения шага квантования m_z .

2. Вероятность получения максимальной эффективности от сжатия более значительна при возможно большей длине шага квантования m_z . Однако с увеличением значения m_z увеличивается и 2^{m_z} , а это обстоятельство, в свою очередь, порождает другие трудности (при больших значениях m_z увеличивается общее количество множеств алфавита сжатия, в результате чего увеличивается требуемый объем памяти для хранения букв).

3. При увеличении значения m_z увеличиваются параметры алгоритмов обработки.

Таким образом, для принятия компромиссного решения необходимо рассматривать все перечисленные особенности в комплексе. Для программной реализации алгоритмов выбранных методов максимальная длина шага квантования принята $m_z = 8$, т. е. $z = 8$. При этом $m_z = \{3, 4, 5, 6, 7, 8\}$, так как при $z = 2$ существенная эффективность преобразования маловероятна.

RPOS-преобразование. Исходными данными для работы указанного алгоритма являются последовательности (1) и (2), полученные в результате работы программы статистического анализа файла и упорядочения частоты встречаемости букв. Основные типовые процедуры следующие:

- выборка очередного кванта из сегмента обработки;
- сравнение выбранного кванта с соответствующим эталонным квантом;
- формирование ЛШКП;

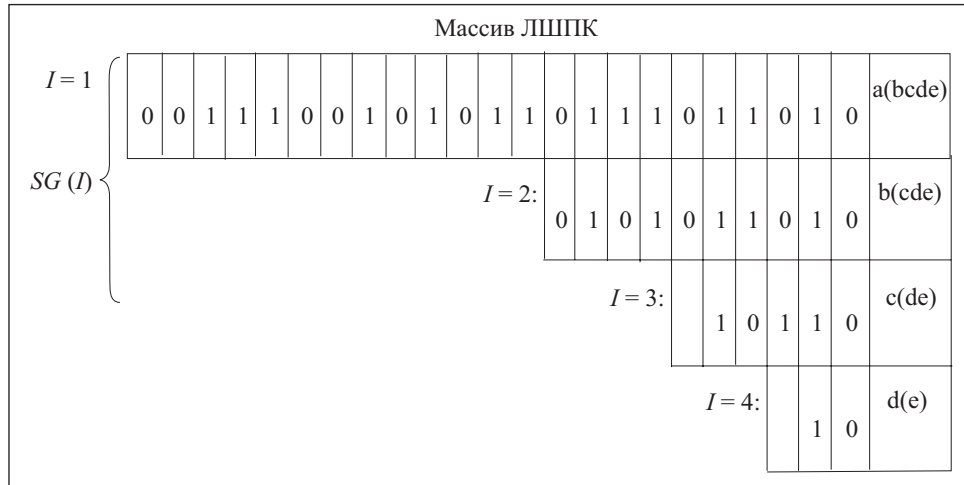


Рис. 1. Процесс получения ЛШПК

формирование выходных данных RPOS-преобразования (тега, шкал и др.).

Время работы программы в основном линейно зависит от длины обрабатываемого сегмента данных (и следовательно, от размера файла обработки). Поэтому для достижения высокой производительности необходимо применять методы параллельной обработки исходных квантов. Для этого должны быть разработаны типовые модули параллельной обработки квантов.

Пример. Пусть задана такая последовательность букв (исходный сегмент): «асбааадбасбабааааае». Процесс сжатия данной информации методом RPOS-преобразования состоит в следующем. Процедура формирования ЛШПК выполняется для каждого неповторяющегося кванта (буквы a, b, c, d, e) исходного сегмента (рис. 1) — массив ЛШПК $SG(I)$. Длина каждой шкалы и переменная величина зависит от частоты встречаемости буквы, для которой формируется предыдущая шкала. Следовательно, для организации параллельного формирования ЛШПК необходимо учитывать, что предблоком для этих блоков будет расчетный блок определения длин отдельных ЛШПК и соответствующих исключаемых букв. Общее количество блоков (I) формирования ЛШПК должно быть $(2^{m_z} - 1)$.

При выполнении процедуры формирования выходных данных ($\langle a, b, c, d, e, \text{массив ЛШПК } SG(I) \rangle$) программы RPOS-преобразования целесообразно применять стековые структуры организации памяти.

RGRUP-преобразование. Исходными данными для работы указанного алгоритма также являются последовательности (1) и (2). Сначала из

упорядоченной последовательности выбираем группу букв (наиболее часто встречающихся) $GRUP = 2^k$ (где $k \geq m_z - 2$), удовлетворяющую условиям эффективности сжатия. Затем формируем следующие компоненты сжатого сегмента:

ЛШПК (основное отличие от предыдущего метода состоит в том, что в данном случае формируется только одна шкала, обозначенная $L(GRUP)$) —

$$L(GRUP) = (r_1, r_2, \dots, r_{K_z});$$

$$r_i = \begin{cases} 1, & \text{если } \beta_i(m_z) \in GRUP, \\ 0, & \text{если } \beta_i(m_z) \notin GRUP; \end{cases}$$

кортеж (сжатый сегмент) —

$$s_0^{(1)} = (y_1, y_2, \dots, y_{K_z});$$

$$y_i = \begin{cases} \beta_i(m_z), & \text{если } r_i = 0, \\ z_i \in Z_0, & \text{если } r_i = 1; \end{cases}$$

$$Z_0 = \{(b_1 b_2 b_3 \dots b_k)_n\}, \quad n = 0, 1, 2, \dots, (2^{k_z} - 1), \quad b_k = \{0, 1\}.$$

Здесь K_z — общее число квантов (букв); k — разрядность сжатых кодов.

Выходные данные программы RGRUP формируются в виде кортежа

$$s_1^{(1)} = (\text{тег}, L(GRUP), s_0^{(1)}).$$

Следует заметить, что после получения статистических данных (о частоте встречаемости букв) файла прежде всего должен быть определен эффективный для конкретного файла алгоритм преобразования. Эта процедура является промежуточной между процедурами статистической обработки и непосредственно преобразования. При сравнении данного алгоритма с предыдущим видно, что они состоят приблизительно из аналогичных модулей типовых процедур.

Таблица 2

Буква	Частота встречаемости	Код
a	12	000
b	4	001
c	3	010
d	1	011
e	1	100

Примечание: коды букв условные

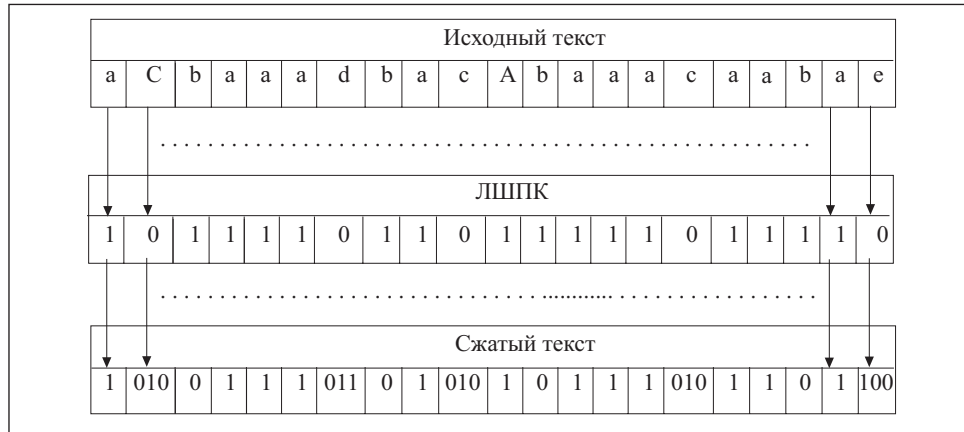


Рис. 2. Последовательности формирования ЛШПК и сжатого текста

Пример. Рассмотрим ту же последовательность букв «a c b a a a d b a c a b a a a c a a b a e». Процесс сжатия выполняется в следующем порядке. Сначала вычисляются и упорядочиваются частоты встречаемости каждой буквы (табл. 2.).

Предположим, что $m_z = 3$ (кванты — длина в битах каждой буквы) и $k = 1$ (длина сжатых букв). Тогда количество выбранных букв, подлежащих сжатию, равно 2^k . Это буквы «a» и «b». На основании этого формируется последовательность преобразования.

Выходные данные формируются в виде

$$S_1^{(1)} = (\underbrace{a, b, 1, 0}_{\text{тег}}, \text{ЛШПК}, \text{сжатый текст}),$$

а последовательности ЛШПК и сжатого текста — в виде, представленном на рис. 2.

В сжатом тексте вместо трехразрядного кода букв ($a = 000$, $b = 001$) используем одnorазрядные коды $a_{сж} = 1$ и $b_{сж} = 0$.

При сравнении двух схем сжатия видно, что кроме условий сжатия, таких как вид текста, адаптация и многосторонность работы с разными жанрами, существует много других факторов, например количество памяти и времени, необходимое для осуществления сжатия, которые следует учитывать для каждой схемы. В табл. 3 для сравнения приведены результаты сжатия текста с помощью наиболее известных методов сжатия и с использованием предложенных алгоритмов, полученные на основе численного эксперимента.

Средства сжатия и развертывания по критерию эффективности сжатия могут быть разделены на два класса. К первому классу можно отнести средст-

Таблица 3

Вид информации	Тип файла	Исходный размер файла (байт)	Результат сжатия по программе				
			Arj.exe	Lharc.exe	Pkzip.exe	Cf.exe RPOS	Ca.exe RGRUP
Машинные коды	f1.exe	103189	<u>61375</u> 1,7	<u>65149</u> 1,6	<u>65353</u> 1,6	<u>102873</u> 1,0	<u>93079</u> 1,1
Английский текст	f2. Txt	8764	<u>2525</u> 3,5	<u>2637</u> 3,3	<u>2675</u> 3,3	<u>6986</u> 1,3	<u>4287</u> 2
База данных	f3.dBf	362	<u>200</u> 1,8	<u>135</u> 2,7	<u>236</u> 1,5	<u>165</u> 2,2	<u>138</u> 2,6
Русский текст	f4. Rtxt	26279	<u>4343</u> 6	<u>4650</u> 5,6	<u>5863</u> 4,4	<u>11876</u> 2,2	<u>10461</u> 2,5
Числовая информация	f5.dt	269	<u>236</u> 1,1	<u>199</u> 1,4	<u>278</u> 0,96	<u>164</u> 1,6	<u>181</u> 1,5

Примечание: под чертой указан коэффициент сжатия

ва, обеспечивающие максимальную степень сжатия с наибольшей производительностью. Если при этом используются только программные средства, то они занимают большой объем памяти, состоят из большого числа модулей и отличаются сложным алгоритмом обмена программными модулями между различными уровнями памяти. В этом случае целесообразно использовать параллельные алгоритмы обработки данных [10]. Тогда исследованные методы и алгоритмы сжатия и развертывания данных, основанные на применении ЛШПК, являются наиболее подходящими.

Ко второму классу относятся используемые для сжатия программно-аппаратные средства. При этом аппаратные расходы лимитируются параметрами производительности системы в целом и эффективностью сжатия данных. Такая система характеризуется наличием большого числа разнообразных специализированных модулей сжатия и развертывания данных и достаточно сложным алгоритмом их коммутации.

На основе рассмотренных алгоритмов можно разработать программные и программно-аппаратные средства, относящиеся к обоим указанным классам. В результате анализа предложенных методов компрессии выделим следующие типовые укрупненные процедуры:

1. Квантование двоичной стохастической последовательности переменным шагом квантования $m=3, 4, \dots, 8$ получения статистических данных и их упорядоченной последовательности (букв алфавитов).

2. Формирование тега и массива ЛШПК.

3. Формирование преобразования данных, т.е. сжатие.
4. Развертывание (каждый метод компрессии имеет соответствующий метод обратного преобразования).
5. Определение эффективного метода (алгоритма) преобразования конкретного сегмента (анализ локальных статистических данных).

Для повышения эффективности работы всей системы данные процедуры должны выполняться с максимально возможными совмещениями микро- и макроопераций. В паспортных тегах файлов должны находиться необходимые, заранее полученные, статистические данные [12]. Тогда процессы сжатия будут выполняться более производительнее, без привлечения дополнительных программных средств, а типовые модули обработки данных дадут возможность коммутировать более эффективную систему сжатия и развертывания данных.

Выводы. Предложенные методы сжатия информации, основанные на применении абстрактного множества (алфавита) двоичных квантов без учета структуры его построения и семантики позволяют найти системное решение задачи компрессии данных и дают возможность осуществлять их многократное сжатие, т.е. сжатие сжатых данных. Недостатком данных алгоритмов является быстрое действие операций сжатия вследствие предварительного статистического анализа входных файлов. Для устранения этого недостатка исследуются возможности аппаратно-программной реализации и распараллеливание обработки для различных шагов квантования при определении эффективности сжатия.

Одним из преимуществ предложенных алгоритмов является возможность их применения при защите информации (они не позволяют криптоаналитику установить присущий естественному языку статистический порядок) и в системах анализа текстовой информации для структурного представления.

Computer algorithms of the universal scheme of a compression and ambiguous decompression of arbitrary binary-coded information have been considered. Methods of compression based on the statistic data and application of logical scales of positional coding bringing uniformity in the processes of compression and decompression and allowing to develop relatively universal effective means for compression of the data are offered.

1. Глушков В. М. Основы безбумажной информатики. «Наука Э». — М. : Гл. ред. физ.-мат. литературы, 1987. — С. 552.
2. Пейстрик Г. Как удвоить, а то и утроить, емкость жесткого диска// PC Magazine (Russian Edition). — 1992. — № 2. — С. 15—26.
3. Chen Z., Gehrke J., Korn F. Query Optimization in Compressed Database Systems // Proc. 2001 ACM-SIGMOD Int. Conf. Management. Santa Barbara. — CA. — 2001. — P. 271—282.

4. Сулейманов А. Ш., Пашаев И. С. и др. Обнаружение ошибок при передаче числовых массивов в сети ЭВМ с использованием упорядочения // Изв. вузов «Нефть и газ» .— 1997. — № 3—4. — С. 52—54.
5. Курбаков К. И. Кодирование и поиск информации в автоматическом словаре. — М. : Сов. радио, 1968. — С. 246.
6. Ziv J., Lempel A. A Universal Algorithm for Sequential Data Compression // IEEE Transactions on Information Theory. — 1977. — Vol. IT-23, № 3. — P. 337—343.
7. Ziv J., Lempel A. Compression of Individual Sequences via Variable-Rate Coding // Ibid.— 1978. — Vol. IT-24, № 5. — P. 530—536.
8. Сулейманов А. Ш. Структурное представление текстовой информации на основе методов сжатия // Информационные технологии моделирования и управления. — 2008. — Вып. 7 (50). — С. 765—771.
9. Сулейманов А. Ш. Семантическая близость и семантические расстояния между текстами // Международный научно-технический сб. «Адаптивные системы автоматического управления» — Днепропетровск : Системные технологии, 2007. — 164 с.
10. Сулейманов А. Ш., Дамадаев М. М. Аппаратные и программные средства динамического сжатия данных // Науч. тр. Национальной академии авиации. — 1999. — № 1. — С. 169—174
11. Сулейманов А. Ш., Касумов Н. К. Об одном методе компрессии в решении проблем резервного копирования и восстановления данных на фоне сближения возможностей SAN и NAS // Изв. НАН Азербайджана. Сер. физико-технических и математических наук. — 2004. — XXIV, № 2. — С. 35—39.
12. Сулейманов А. Ш. Интеллектуализация систем обработки информации. — Баку : «Элм», 2004. — 240 с.

Поступила 18.02.09;
после доработки 09.04.09

СУЛЕЙМАНОВ Акиф Шамил оглы, канд. техн. наук, доцент, зав. кафедрой Азербайджанского технического университета. В 1976 г. окончил Московский энергетический ин-т. Область научных исследований — интеллектуализация системы обработки информации, сжатие информации.