AN EFFICIENT COMPRESSION ALGORITHM USING DICTIONARY-TYPE DATA TRANSFORMATION

M. P. Bakulina

Institute of Computational Mathematics and Mathematical Geophysics SB RAS,
630090, Novosibirsk, Russia

DOI: 10.24412/2073-0667-2025-4-5-10
EDN: KUQHBT

The problem of efficient lossless compression for dictionary-type data is considered. For such data, the coding algorithm is based on the use of a dictionary formed from the text received for compression. It is also known that data processing, such as BWT, can improve the text compression ratio. In this paper, an efficient dictionary-type data compression algorithm based on the modification of BWT is proposed. Experimental results are presented. The results confirm the increase in the data compression ratio by the proposed algorithm compared to the classic archiver bzip2.

Key words: dictionary, BWT-transformation, compression ratio, encoding time, archiver.

References

1. Burrows, M. A block sorting lossless data compression algorithm. M. Burrows, D. Wheeler, Technical Report 124, Digital Equipment Corporation, 1994. P. 18.

2. Ryabko B. Ya. Data compression by means of a “book stack” // Problems of Information Transmission. 1980, V. 16: (4). P. 265–269.

3. Hmelev D. V. Preobrazovanie Barrouza-Willera, massiv suffiksov i szhatie slovarei // Vse o szhatii dannyh, izobrazhenii i video. [Electron. Res.]: `http://compression.ru/download/articles/bwt/khmelev_2003_bwt.pdf`.

4. Kärkkäinen J., Sanders P. Simple linear work suffix array construction // In 30th International Colloquium on Automata, Languages and Programming, number 2719 in LNCS, 2003. P. 943–955.

5. Bell T. C., Cleary J. H., Witten I. H. Text compression. / T. C. Bell, J. H. Cleary, I. H. Witten, Prentice Hall. Englewood Cliffs, 1990. P. 318.

The research was carried out within the framework of a state assignment of the Institute of Computational Mathematics and Mathematical Geophysics SB RAS (ICM&MG SB RAS) 0251-2022-0005.

ЭФФЕКТИВНЫЙ АЛГОРИТМ СЖАТИЯ С ПОМОЩЬЮ ПРЕОБРАЗОВАНИЯ ДАННЫХ СЛОВАРНОГО ТИПА

М. П. Бакулина

Институт вычислительной математики и математической геофизики СО РАН,
630090, Новосибирск, Россия

УДК 519.722
DOI: 10.24412/2073-0667-2025-4-5-10
EDN: KUQHBT

Рассматривается задача эффективного сжатия без потерь для данных словарного типа. Для таких данных алгоритм кодирования основан на использовании словаря, формируемого по тексту, поступающему для сжатия. Известно также, что предварительная обработка данных, например, BWT-преобразование, может улучшить коэффициент сжатия текста. В данной работе предлагается эффективный алгоритм сжатия данных словарного типа, основанный на модификации BWT-преобразования. Приведены экспериментальные результаты, подтверждающие увеличение степени сжатия данных предложенным алгоритмом по сравнению с классическим архиватором.

Ключевые слова: словарь, BWT-преобразование, степень сжатия, время кодирования, архиватор.

Введение. Сжатие данных является одной из важных задач теории информации. Это связано с тем, что задача компактного представления данных с сохранением возможности их однозначного восстановления (декодирования) возникает достаточно часто на практике. Так, предварительное сжатие позволяет существенно улучшить характеристики системы связи. Сжатие применяется при хранении информации, ее передаче со спутников и в других приложениях, причем актуальность проблемы сжатия возрастает в связи с увеличением объемов передаваемой и хранимой информации.

В настоящее время широко известны и находят практическое применение многие алгоритмы компактного представления данных, которые дают сжатие лучше, чем стандартные архиваторы. Один из таких подходов при создании алгоритмов сжатия основан на преобразовании Барроуза-Уилера или BWT-преобразовании [1]. BWT используется для предварительной обработки данных перед сжатием. Преобразование меняет порядок символов во входной строке таким образом, что повторяющиеся подстроки образуют на выходе идущие подряд последовательности одинаковых символов.

Так как в полученной после BWT-преобразования последовательности вероятности появления одинаковых символов выше, чем редких, то возникает задача эффективного кодирования длин серий. В известных алгоритмах сжатия на основе BWT для этого выполняется шаг по замене каждого символа расстоянием до его предыдущей встречи (например, в алгоритме move to front, MTF [2]), а далее к полученному набору чисел

Исследования выполнены в рамках государственного задания ИВМиМГ СО РАН 0251-2022-0005.

применяется метод энтропийного сжатия, например, кодирование Хаффмана или арифметическое кодирование. На практике алгоритм сжатия вида BWT > MTF > Хаффман применен в архиваторе bzip2.

Кроме того, практика показывает, что большинство архиваторов, использующих BWT, достигают хорошего сжатия на текстовых данных, например, текстовых файлов на естественных языках. В [3] доказана обратимость BWT-преобразования и возможность восстановить массив суффиксов.

Кроме алгоритмов, использующих побуквенное кодирование, примером которого является код Хаффмана, известны также методы словарного сжатия данных [4]. В таких методах для имеющегося сообщения, например, текста на естественном языке, составляется словарь, представляющий собой список повторяющихся последовательностей символов (слов) с указанием частоты их появления. Тогда при кодировании сообщения каждому слову приписывается код, длина которого тем меньше, чем выше частота его встречаемости. Отметим, что архиватор bzip2 является одним из лучших архиваторов для данных словарного типа.

В данной работе предлагается эффективный алгоритм сжатия данных словарного типа, основанный на модификации BWT-преобразования. Приводятся экспериментальные результаты, подтверждающие эффективность предложенного метода.

1. Преобразование Барроуза-Уилера. Процедуру BWT-преобразования можно условно разделить на 4 этапа: 1) выделяется блок из входного потока исходных данных; 2) формируется матрица всех перестановок, полученных в результате циклического сдвига блока; 3) все перестановки сортируются в соответствии с лексикографическим порядком символов каждой перестановки; 4) на выход подаются последний столбец матрицы и номер строки, соответствующий первоначальному блоку.

Опишем теперь формальную процедуру BWT-преобразования.

Пусть текст T состоит из $N+1$ букв, занумерованных с нуля: $T[0\ldots N]$. Буквы $T[i]$ принадлежат некоторому упорядоченному алфавиту A. Зададим на строках из букв алфавита A лексикографический порядок $\leq$. Обозначим через $S_k(T)$ циклический сдвиг текста T на k символов влево:

$$S_k(T)[j] = T[(j+k)(mod\ n+1)].$$

Существует перестановка σ чисел $\{0,\ldots,N\}$, которая удовлетворяет условию

$$S_{\sigma(i)}(T) \leq S_{\sigma(i+1)}(T), \qquad i = 0,\ldots,N-1. \tag{1}$$

Перестановку σ называют суффиксным массивом. Она играет важную роль в индексации информации (например, с их помощью можно найти все повторяющиеся подстроки текста). Тогда преобразование Барроуза-Уилера текста T — это текст $B[0\ldots N] = BW(T)$, буквы которого заданы соотношением

$$B_i = \left(S_{\sigma(i)}T\right)[N] = \left(S_{\sigma(i)-1}T\right)[0] = T[\sigma(i) - 1\ (mod\ N+1)].$$

В [3] доказано, что для восстановления исходного текста T из преобразования B достаточно знать число I, задаваемое условием $\sigma(I) = 0$ или $S_{\sigma(I)}T = T$. Чтобы найти перестановку σ, можно воспользоваться любым из известных методов построения суффиксных массивов для текста T (см., например, [5]), а затем добавить суффикс, соответствующий символу $T(N)$.

2. Преобразование данных словарного типа. Рассмотрим теперь BWT-преобразование для данных словарного типа и на основе него опишем новое преобразование, которое будет основой нового алгоритма сжатия данных.

Пусть текст $T[0\ldots N]$ состоит из n частей, разделенных специальными символами (разделителями) $\Lambda_0\ldots\Lambda_{n-1}$, положение которых упорядочено, то есть

$$T = T_0\Lambda_0\ldots T_{n-1}\Lambda_{n-1}.$$

Будем считать также, что разделители лексикографически строго предшествуют буквам алфавита A:

$$\Lambda_k < a \quad \text{для } \forall a \in A\backslash\{\Lambda_0,\ldots,\Lambda_{n-1}\} \quad \text{и} \quad \forall k = 0,\ldots,n-1.$$

Тогда при применении BWT-преобразования все разделители будут стоять в первом столбце матрицы следующего вида:

$$\begin{gathered}
\Lambda_{\omega(0)}\ldots B_0\\
\vdots\\
\Lambda_{\omega(n-1)}\ldots B_{n-1}\\
B_{\eta(n)}\ldots B_n\\
B_{\eta(n+1)}\ldots B_{n+1}\\
\vdots\\
B_{\eta(N)}\ldots B_N,
\end{gathered}$$

где ω — перестановка для разделителей, а η — перестановка для букв последнего столбца.

Рассмотрим теперь преобразование исходного текста B, то есть текст $\tilde{B}$, который получен из текста B с помощью замены всех разделителей Λ_k на один разделитель Λ, предшествующий всем символам алфавита A исходного текста T. Формально это преобразование запишется так:

$$\tilde{B}_i = \begin{cases} B_i, \text{ если } B_i \neq \Lambda_k, \quad k = 0,\ldots,n-1\\ \Lambda, \text{ если } B_i = \Lambda_k \text{ для некоторого } 0 \le k \le n-1. \end{cases}$$

В [3] показано, что если по $\tilde{B}_i$ с соответствующими перестановками вместо индекса i сформировать строки $\tilde{T}_J$, то наборы строк преобразованного текста $(\tilde{T}_J, j = 0,\ldots,n-1)$ и строк исходного текста $(T_k,\ k = 0,\ldots,n-1)$ совпадают с точностью до перестановки. Кроме того, по $\tilde{B}$ однозначно восстанавливается и исходный текст T.

3. Алгоритм сжатия данных B_NEW. На основе приведенного выше преобразования опишем теперь новый алгоритм сжатия данных словарного типа B_NEW.

Пусть мы имеем текст, представляющий собой n последовательностей символов $T_0,\ldots,T_{n-1}$, разделенных специальными символами-разделителями:

$$T = T_0\Lambda_0\ldots T_{n-1}\Lambda_{n-1},$$

при этом порядок разделителей соответствует лексикографическому порядку текста, то есть $\Lambda_i < \Lambda_j \Leftrightarrow T_i < T_j$.

Алгоритм B_NEW состоит из двух этапов. На первом этапе построим перестановку σ, удовлетворяющую условию (1). При этом для простоты порядок на суффиксах будем задавать так: $\Lambda_i < \Lambda_j \Leftrightarrow i < j$. Далее с помощью перестановки σ строим текст $\tilde{B}$ по правилу:

Таблица 1
Результаты сравнения степеней сжатия текстов с помощью B_NEW и bzip2

№	Название	V текста (байт)	k (бит/символ)	
			B_NEW	bzip2
1	article	67739	**2,79**	2,82
2	book	349270	**3,06**	3,11
3	journal	307930	**2,71**	2,74
4	document	962186	**2,68**	2,79
5	works	5410342	**2,51**	2,66

Таблица 2
Результаты сравнения времени кодирования с помощью B_NEW и bzip2

№	Название	V текста (байт)	t (с)	
			B_NEW	*bzip2*
1	article	67739	**0,86**	0,73
2	book	349270	**5,43**	5,01
3	journal	307930	**5,12**	4,78
4	document	962186	**19,05**	18,32
5	works	5410342	**63,15**	61,48

$$\tilde{B}_i = \begin{cases} S_{\sigma(i)-1}T[0], \text{ если } S_{\sigma(i)-1}T[0] \neq \Lambda_k,\ k = 0,\ldots,n-1 \\ \Lambda, \text{ если } S_{\sigma(i)-1}T[0] = \Lambda_k \text{ для некоторого } 0 \leq k \leq n-1. \end{cases}$$

На втором этапе полученный после такого преобразования (назовем его B_new) текст $\tilde{B}$ подвергнем сжатию с помощью известного метода MTF (см. [2]), а к получившемуся набору чисел — алгоритм Хаффмана, то есть осуществим преобразование B_new > MTF > Хаффман. Отметим, что эта цепочка преобразований аналогична преобразованию в архиваторе bzip2. Отличие заключается в том, что вместо BWT-преобразования к данным словарного типа применяется новое преобразование B_new, позволяющее с помощью группировки данных повысить эффективность сжатия.

4. Экспериментальные результаты и их сравнение. Для подтверждения эффективности предложенного алгоритма сжатия B_NEW было проведено сравнение результатов его работы с результатами работы архиватора bzip2. В качестве данных словарного типа были взяты следующие данные: article — статья в журнале; book — монография; journal — математический журнал; document — документация; works — собрание сочинений А. С. Пушкина.

Сравнение проводилось по двум характеристикам — степени сжатия (использовалась наиболее употребляемая единица измерения бит/символ) и времени кодирования и декодирования. В Таблице 1 приведены результаты степеней сжатия выбранных текстов алгоритмом B_NEW и архиватором bzip2.

В Таблице 2 приведены результаты времени кодирования и декодирования этих же текстов. В качестве результата принималось усредненное общее время работы программы по 10 запускам (использовался таймер в 1000 Гц).

Из таблиц 1 и 2 видно, что предложенный алгоритм B_NEW показывает лучшую степень сжатия по сравнению с алгоритмом bzip2 при сравнительно небольшом увеличении

времени кодирования и декодирования. Заметим также, что наиболее существенное увеличение сжатия наблюдается для текстов с большим объемом памяти, например, document и works. Эти тексты обладают большим объемом словаря, и поэтому предложенное в п. 3 преобразование данных словарного типа должно давать для этих текстов наиболее эффективное сжатие, что и подтверждается практическими результатами.

Заключение. Проведенное экспериментальное сравнение предложенного метода кодирования, основанного на модификации BWT-преобразования, со стандартным архиватором bzip2 подтверждает эффективность предложенного алгоритма B_NEW. Данный алгоритм может быть использован для сжатия различных данных словарного типа, например, текстов на естественных языках.

Список литературы

1. Burrows M. A block sorting lossless data compression algorithm // M. Burrows, D. Wheeler. Technical Report 124, Digital Equipment Corporation, 1994. 18 pages.

2. Рябко Б. Я. Data compression by means of a «book stack» // Problems of Information Transmission. 1980, V. 16: (4). P. 265–269.

3. Хмелев Д. В. Преобразование Барроуза-Уилера, массив суффиксов и сжатие словарей // Все о сжатии данных, изображений и видео: сайт. [Эл. Рес.]: `http://compression.ru/download/articles/bwt/khmelev_2003_bwt.pdf`.

4. Bell T. C., Cleary J. H., Witten I. H. Text compression // T. C. Bell, J. H. Cleary, I. H. Witten, Prentice Hall. Englewood Cliffs, 1990. C. 318.

5. Kärkkäinen J., Sanders P. Simple linear work suffix array construction // In 30th International Colloquium on Automata, Languages and Programming, number 2719 in LNCS. 2003. C. 943–955.

Бакулина Марина Павловна — канд. физ.-мат. наук, научный сотрудник Института вычислительной математики и математической геофизики (ИВМиМГ СО РАН). Научные интересы: теория кодирования, сжатие данных, моделирование информационных систем, e-mail: `marina@rav.sscc.ru`.

Bakulina Marina Pavlovna is a Researcher of Institute of Computational Mathematics and Mathematical Geophysics (ICMMG SB RAS). PhD in Physics and Mathematics. Research interests: coding theory, data compression, information systems modeling.

Дата поступления — 17.08.2025