

# МОДИФИЦИРОВАННЫЙ АЛГОРИТМ СЖАТИЯ ДАННЫХ

Гуринович А. Б., Митьковец Л. В., Сидоров Д.  
Кафедра информационных технологий и управления,

Белорусский государственный университет информатики и радиоэлектроники  
Минск, Республика Беларусь

E-mail: gurinovich@bsuir.by, lidiatommo@gmail.com, sam65th@mail.ru

*Сжатие данных применяется для более рационального использования устройств хранения и передачи данных путем устранения избыточности. Оно будет оставаться в сфере внимания из-за все более возрастающих объемов хранимых и передаваемых в ЭВМ данных, кроме того, его можно использовать для преодоления некоторых физических ограничений, таких как, например, сравнительно малая пропускная способность линии.*

## ВВЕДЕНИЕ

Исследуемый модифицированный алгоритм компрессии является алгоритмом сжатия данных без потерь. Предлагаемый алгоритм J-битного кодирования (ЖВЕ) [4] работает путем манипулирования битами данных для уменьшения размера и оптимизации входных данных для другого алгоритма. Идея этого алгоритма состоит в том, чтобы разделить входные данные на две группы: первая будет содержать исходный ненулевой байт, а вторая — битовое значение, объясняющее положение ненулевого «1» и нулевого «0» байтов. Затем обе группы могут быть сжаты отдельно с помощью другого алгоритма сжатия данных для достижения максимальной степени сжатия.

### I. ПОШАГОВЫЙ ПРОЦЕСС J-БИТНОГО КОДИРОВАНИЯ.

Пошаговый процесс сжатия можно описать следующим образом:

- Считывание входных данных по байтам.
- Определение прочитанных байт как «0» или «1» байт.
- Запись «0» байт в данные I и запись бита «1» во временные байтовые данные, или бита «0» во временные байтовые данные для нулевого входного байта.
- Повторение первых трех шагов до тех пор, пока временные байтовые данные не будут заполнены 8 битами данных.
- Если временные байтовые данные заполнены 8 битами, то запись байтового значения производится в данные II.
- Очистка временных байтовых данных
- Повторение первых шести шагов до тех пор, пока не будет достигнут конец файла.
- Запись комбинированных выходных данных:
  1. Запись исходной входной длины.
  2. Запись данных I.
  3. Запись данных II.

На рис. 1 показан визуальный пошаговый процесс сжатия для J-битного кодирования. Вставленная исходная длина входного сигнала в начало выходного сигнала будет использоваться в качестве информации для размера данных I и данных II.

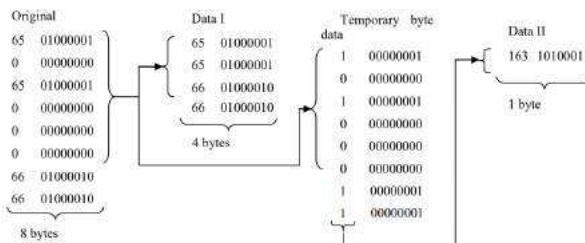


Рис. 1 – Алгоритм ЖВЕ

Пошаговый процесс декомпрессии можно описать следующим образом:

- Считывание исходной длины входного сигнала.
- Если было сжато отдельно, то распаковка данных I и данных II (необязательно).
- Побитное считывание данных II.
- Определение, является ли бит чтения «0» или «1».
- Запись на выход: если бит чтения равен «1» то чтение и запись данных I на выход, если бит чтения равен «0» то запись нулевого байта на выход.
- Повторение шагов 2-5 до тех пор, пока не будет достигнута исходная длина входного сигнала.

### II. СРАВНЕНИЕ РЕЗУЛЬТАТОВ В КОМБИНАЦИИ С ДРУГИМИ АЛГОРИТМАМИ СЖАТИЯ

В сравнении используются методы компрессии: кодирования длин серий (RLE) [3], преобразования Берроуза-Уилера (BWT) [2], уменьшения избыточности (MTF) [3] и арифметического кодирования (ARI) [3].

Различные комбинации алгоритмов сжатия данных тестируются с пятью типами файлов. Каждый тип состоит из 80 образцов. Каждый образец имеет разный размер, чтобы показать ре-

альное состояние файловой системы. Все образцы не сжаты; они включают в себя необработанные растровые изображения и звук без сжатия с потерями. Образцы для эксперимента приведены на рис. 2.

Ном.	Тип	Кол-во	Хар-ки
1	Растровое изображение	80	Raw 8 bit
2	Растровое изображение	80	Raw 24 bit
3	Текстовый документ	80	
4	Исполняемый файл	80	
5	Звук WAVE	80	WAV
6	Видео AVI	80	VBR

Рис. 2 – Используемые в сравнении образцы

Последовательность комбинаций методов компрессии:

1. BWT+MTF+RLE;
2. BWT+RLE+ARI;
3. RLE+BWT+MTF+RLE+ARI;
4. RLE+BWT+MTF+JBE+ARI.

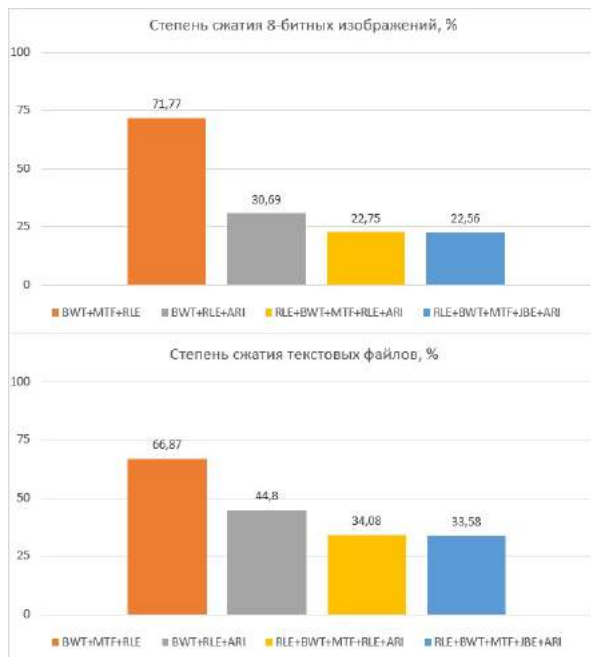


Рис. 3 – Сравнение результатов сжатия

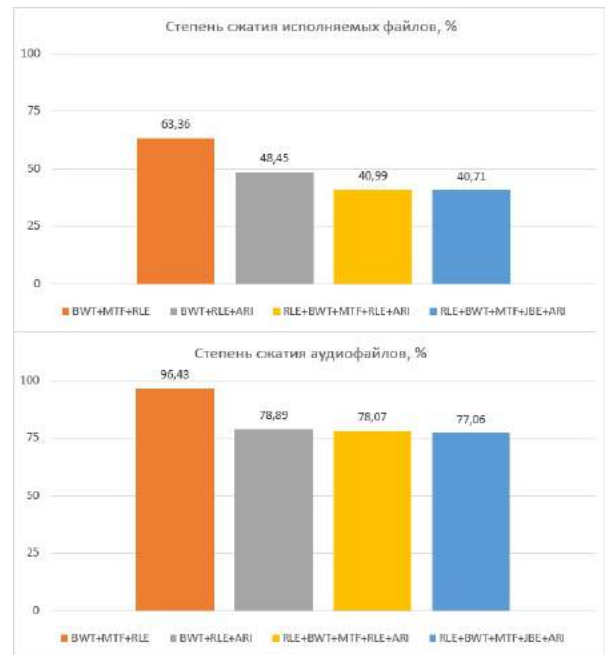


Рис. 4 – Сравнение результатов сжатия (продолжение)

На рис. 3 и рис. 4 показано, что используемые образцы различных типов данных сжимаются с большей степенью компрессии в сочетании с предложенным алгоритмом.

### III. ЗАКЛЮЧЕНИЕ

В исследовании проведен эксперимент с использованием 6 типов файлов с 80 различными размерами для каждого типа. В результате протестировано и проведено сравнение 4 комбинаций алгоритмов. Предложенный алгоритм дает лучшую степень сжатия при вставке между преобразованиями «move-to-front transform» (MTF) и арифметическим кодированием (ARI). Рассматриваемый алгоритм имеет перспективу внедрения в структуру других алгоритмов сжатия данных.

### IV. СПИСОК ЛИТЕРАТУРЫ

1. Capo-chichi, E. P., Guyennet, H. and Friedt, J. K-RLE a New Data Compression Algorithm for Wireless Sensor Network. In Proceedings of the 2009 Third International Conference on Sensor Technologies and Applications.
2. Nelson, M. 1996. Data compression with Burrows-Wheeler Transform. Dr. Dobb's Journal.
3. Springer, Handbook of Data Compression Fifth Edition.
4. Agus Dwi Suarjaya. Information Technology Department Udayana University. Bali, Indonesia. A New Algorithm for Data Compression Optimization.