International Journal of Research in Engineering, Science and Management
Volume-2, Issue-5, May-2019
www.ijresm.com | ISSN (Online): 2581-5792

Compression of Data using Genetic Algorithm and Reversible Cellular Automata

Pratik Kulkarni[1], Tejal Gaikwad[2], Shruti Nehete[3], Ameya Naike[4], Vanita Babanne[5]
[1,2,3,4]*Student, Department of Computer Engineering, RMD Sinhgad School of Engineering, Pune, India*
[5]*Professor, Department of Computer Engineering, RMD Sinhgad School of Engineering, Pune, India*

***Abstract*: We describe a lossless data compression technique and its implementation. Data compression is reduction of number of bits which are needed to store or transmit the data. In this paper we are concerned with the optimized reversible cellular automata rules. This is achieved with the help of an Evolutionary Computing. The Genetic Algorithm is used in order to find the best possible rule of Reversible Cellular Automata that reversibly transforms high-entropy binary data into lower entropy binary data. The results we obtained show that this technique could be a significant part of all compression methods.**

***Keywords*: Genetic Algorithm, Reversible Cellular Automata, data compression**

1. Introduction

Data Compression is the reduction in the number of bits which represent the data. The most widely used compression techniques currently are model fit algorithms, i.e they fit their model of compression based on given data. In this paper we introduce a method to transform data to fit the encoding scheme, instead of modelling the encoding scheme to fit the data. The current algorithms which are used, estimate the probability distribution of the data first and then assign shorter codes to the most frequent symbols. The data compression model uses this encoded data as it has smaller size than the original data [1]. Huffman coding is proven optimal for a symbol to symbol coding with a known probability distribution.

Briefly, we transform the data by changing its probability distribution to lower its entropy states. The transformation which takes place has to be reversible to achieve lossless data compression. The proposed compression algorithm is an adaptation of a simple procedure in Burrows Wheeler Transformation [2]. Burrows Wheeler Transformation uses the concept of Run Length Encoding to cluster same letters together and lowering the probabilistic entropy of the data. It has a similar way to transform the text data.

The main challenge is to achieve optimal coding. There are many efficient and optimal solutions which describe the coding problem. But, optimal encoding is proved to be non-computable [3]. Kolmogorov Complexity for calculation of entropy is used. It uses the number of bits of the shortest program that can print the string. The problem of Encoding and Modelling is classified into Artificial Intelligence. By making use of Artificial Intelligence, the most optimal method can be found to transform the data.

For the demonstration of this idea, we use binary data. To reduce the runs of data the concept of Run Length Encoding is used and then the data is stored in a single data value and count. To transform the data reversibly, Reversible Cellular Automata [4] is an excellent framework which provides the necessary rules. The idea is to reduce the probabilistic entropy of binary data by lessening total amount of '1's in the binary data. We apply a class of AI algorithm called Genetic Algorithm [5] to generate a set of random RCA rules to transform the binary data. A new type of Genetic Algorithm which is completely based on the concept of Converting Infeasible individuals into Feasible ones (CIFGA) is used in this particular paper. As Genetic Algorithm is best when there are unconstrained optimization problems.

The technique which is used in the Image Compression based on Genetic Algorithm optimization which is proposed by Mohammad Omari and Salah Yaichi [6] utilizes the efficiency of Genetic Algorithms to enhance the time search for finding better rational numbers with shorter and compact reduced form. The concept of Move-To-Front coding can also be used in order to reduce the entropy of the data. Combination of MTF and RLE is used [7] for lossless audio encoding.

The rest of the paper is organized as follows. In section 2 we review the basics of our methodology. We describe the working of Reversible Cellular Automata in Margolus Neighbourhood, Run Length Encoding using Hilbert Curve, and the Genetic Algorithm. The theory of Shannon Entropy is discussed in section 3, as well as its relation to Run Length Encoding. The results obtained and the comparison of performance is discussed in section 4 which is followed by conclusion and the future scope in section 5.

2. Proposed work

The main aim is to develop a data compression algorithm which will compress data using reversible transformations that convert data into low algorithmic entropy.

A. Reversible cellular automata

A cellular automaton consists of a regular grid of cells, each in one of a finite number of states, such as on and off. For each

cell, a set of cells called its neighbourhood is defined relative to the specified cell. A new generation is created, according to some fixed rule that determines the new state of each cell in terms of the current state of the cell and the states of the cells in its neighbourhood. Binary data can be encoded as a grid of on and off cells, with '1' being a cell that is alive and '0' being a dead cell.

We make use of Margolus Neighbourhood in partitioned cellular automaton to transform this grid reversibly. The following figure demonstrates how rules in the Margolus Partitioning scheme affect grids of cells

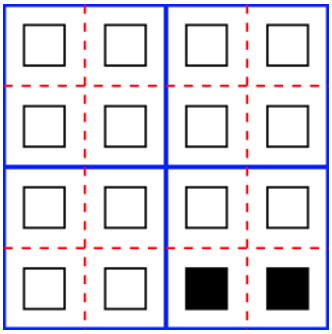

Fig. 1. The Margolus neighborhood for a two-dimensional block cellular automaton. The partition of the cells alternates between the set of 2x2 blocks indicated by the solid blue lines, and the set of blocks indicated by the dashed red lines. The white cells are alive, and dead cells are indicated by black.

In Margolus neighborhood, each block consists of 4 cells, thus the transition function must have 4 inputs and 4 outputs. For binary cellular automata, this gives 24 = 16 possible input and output states. A block transition function must map every possible input tosome output. To be reversible, this mapping must be bijective, thus effectively being a transposition of 16 elements. State of the block can be encoded into one 4-bit binary number.

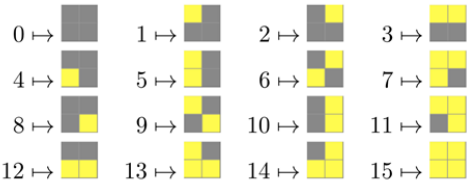

Fig. 2. Each unique partitioning block can be assigned a number from 0-15

Now any function can be written down as a list of 16 numbers in the range [0, 15]. For the rule to be reversible, then each number should occur exactly one time in the list.

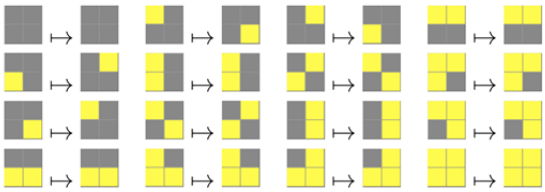

Fig. 3. The rule [0,8,4,3,2,5,9,7,1,6,10,11,12,13,14,15] would apply the given transformations to each block in the margolus partitioning scheme

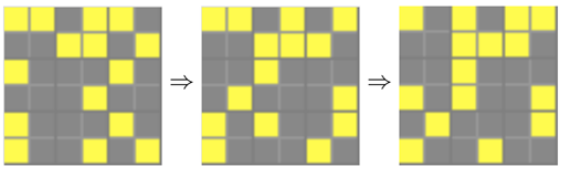

Fig. 4. The same rule [0,8,4,3,2,5,9,7,1,6,10,11,12,13,14,15] is applied twice to a grid. First, partitioned blocks with the dark grey border undergo transformation. Second, the blocks with the light grey border undergo transformation. It is to be noted that the grid is a torus, meaning all the corner cells make up a single block in the second partition.

Since all RCA rules are permutations of the original rule [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15] we can store

B. Binary run length encoding

Run-length encoding (RLE) is a very simple form of lossless data compression which encodes the runs of each element in the data to be encoded. These runs are stored as a single data value and count, rather than as the original run. However for binary data, since we have to deal with only two elements, we only need to store the data of the first element, and then the runs for each consecutive element.

E.g The binary string ‘000001100111’ can be encoded as [‘0’, 5, 2, 2, 3]. To decode, take 5 ‘0’s, concatenate 2 ‘1’s, 2 ‘0’s, and finally 3 ‘1’s. Each consecutive run refers to alternating elements in the string.

C. File I/O

The input file is first read in binary mode. This is then split into 10x10 matrices, and each matrix is given to the Genetic Algorithm to optimize and compress

D. Iterative genetic algorithm

Genetic Algorithms are class of evolutionary AI algorithms that solve both constrained and unconstrained optimization problems.

Using a genetic algorithm, we first generate a set of random solutions and rank them based on their score on the fitness function. The fitness function applies the solution to the binary data, transforming it to see how well it has decreased its Shannon Entropy, and returns the score.

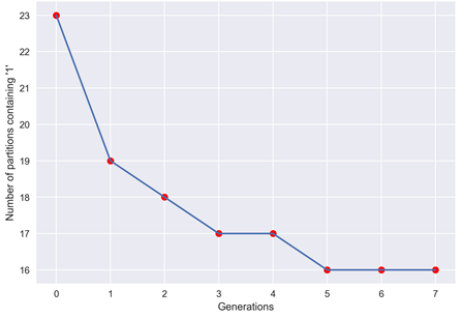

Fig. 5. Generations vs. no. of partitions contained

The top 2% of all solutions are then made to crossover with the top 2.5% of the same set of solutions. A small percentage of

the set is mutated to preserve randomness in solutions. This is done repeatedly until a satisfactory amount of fitness is achieved; over successive generations, the set "evolves" toward an optimal solution. Fig. 6. Within 7 generations, the number of partitions that contain '1' gradually decreases to 16 in a 20x20 matrix.

The graph shows the selection of best fitting chromosome from set of generated chromosome. Genetic Algorithm are iterated until the "best fitting value" of the chromosome stabilize and does not change for the next generation, which means algorithm has converged to the solution.

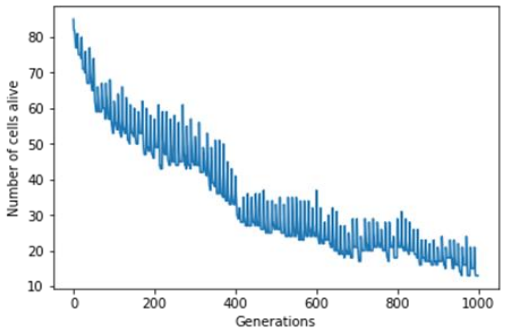

Fig. 6. Generations vs. no. of cell alive

This crossover function inserts randomly selected parts of the parent chromosomes into the child chromosome and also ensures that the new generation of solutions have no repetition. Each set of parents produce two new children. This is done iteratively to consistently produce better optimized solutions.

E. System architecture

Data compression modules keeps developing new technologies to compress and decompress various type of data. In this work we introduce a method for compressing the any kind of data using RCA rules along with run length encoding to compress the data using its algorithmic entropy using Genetic algorithm.

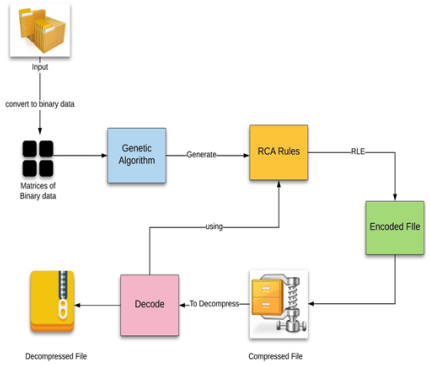

Fig. 7. System architecture

We will take file with any extension as input to system. After that we will convert the target file into binary. All the binary data will be transformed in data matrix. Then will apply RCA transformation on the data matrix to generate output with data with low entropy. Then using Run length encoding to encode the transformed data and generate output file. Finally, will take the output file as input and reverse the process using the RCA rules to generate the output file.

3. Theory

The Shannon entropy equation provides a way to estimate the average minimum number of bits needed to encode a string of symbols, based on the frequency of the symbols.

where P_{i} is the probability of a given symbol.

To decrease entropy, we reduce the amount of 1's in the binary data. The following scatter plot demonstrates how the size of Run Length Encoded data changes when the entropy of the binary data decreases:

Reducing the amount of 1's in the binary data reduces it's Shannon Entropy [11]. For each state i, there is a set of probabilities j and an entropy H for each state which is given by C.E. Shannon. It doesn't matter if we reduce 1's or 0's, the effect will be the same.

We have seen how lowering the Shannon entropy of binary data leads to better compression ratios. Finding the best cellular automata rule that decreases the amount of 1's is an optimization problem. This is particularly hard as the number of solutions, good or bad, that exist are 16! in number.

4. Datasets

A total of 6 file are used in the data set. The data collected included file of any extension and type available in information technology. They ranged from 20 kb to 100 Mb, of which few file included enwik8, jpeg. Following Table gives the Size of the data set used.

Table 1
Filetype and size

Sr. No	File Type	Size
1	Jpeg	5.00 MB
2	mp3	8.53 MB
3	pdf	253 Kb
4	pptx	61 Kb
5	enwik8	100 Mb

5. Results

Table 2
File types and compression ratio

File name	File Type	Size	Compressed Size	Compression Ratio
enwik8.txt	Text File	100 mb	37.7 mb	62.3%
IEEE_Paper.pdf	PDF	306 kb	237 kb	22.6%
generations.png	PNG	94 kb	81 kb	13.83%

Our findings suggest that the existing algorithms work on a single type of data, and none of the techniques contain AI. We introduce Genetic Algorithms in AI using reversible cellular automata to compress the data. Further study shows that this technique takes computational time to compress the data and the data will be decompressed in very short time. To add on to this drawback, we can implement it in parallel using CUDA to decrease the compression time of the data.

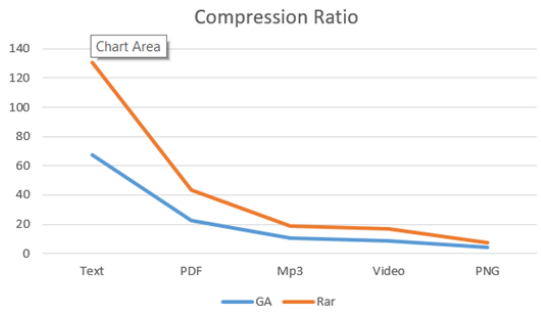

Fig. 8. Compression ratio

6. Discussion

We have described a compression technique that works by applying genetic algorithm using reversible cellular automata. Our algorithm is completely general-purpose. Good compression is achieved on data which is ordered and has low information density like text files, but it does not perform as well on dense file types like PDF and image files. Our technique achieves better compression in relation to previous algorithms because it decompresses the file at lower algorithmic time.

The technique which we have used achieves moderate compression time comparable with other algorithms, depending on the file type used, yet is closer to the Lempel and Ziv. Unlike Lempel and Ziv, our algorithm decompresses the file faster. Previous algorithm were developed to suit the module to the data type and were lossy in nature, and they compressed only particular data. The proposed algorithm compresses any type of file by converting the data into binary format. We are using reversible transformation using RCA therefore the data is not lost during compression. The conversion into binary allows us to compress to any random data without making a different module for the random data.

7. Conclusion

The study shows the effectiveness of genetic algorithms used in the data compression. The use of Artificial intelligence in compression is effective and reasonable to achieve lossless data compression and decompression. This system applies Genetic algorithms to produce Reversible Cellular Automata(RCA) rules. Applying these rules, the data is changed over into low shannon entropy to produce small Run Length Encoding (RLE) to compress the data. The system is rational and effectual lossless data compression. This study also shows it is worthwhile to transform and preprocess data before modelling its probability distribution.

References

[1] Huffman, D.A, “A method for the construction of minimum-redundancy codes,’ Proc. IRE, 1952, 40, (9), pp. 1098-1101
[2] M. Burrows and D. J. Wheeler, “A Block-sorting Lossless Data Compression Algorithm.”
[3] G. J. Chaitin, A. Arslanov, C. Calude, “Program-Size Complexity Computes the Halting Problem,” (1995, Sept.) IBM Research Division, New York, USA, University of Auckland, New Zealand.
[4] Harold V. McIntosh (1991, April) “Proposed Reversible Cellular Automata.”
[5] Yu-Gen Gao and De-Yu song, “A new improved Genetic Algorithm and its property analysis,’ 2006.
[6] Mohammed Omari and Salah Yaichi, “Image Compression Based on Genetic Algorithm Optimization.”
[7] Hend. A. Elsayed, “Burrows-Wheeler Transform and Combination of Move-to-Front Coding and Run Length Encoding for Lossless Audio.”
[8] A universal algorithm for sequential data compression by Jacob Liv and Abraham Lempel, 1977, May.
[9] Lorenzo Cappellari and Giancarlo Calvagno, “Lifting based design of reversible cellular automata for scalable coding of binary images,’ February 2010.
[10] J. L. Bentley, D. D. Sleator, R. E. Tarjan, and V. K. Wei, “A Locally Adaptive Data Compression Scheme.”
[11] C. E. Shannon, “A Mathematical Theory of Communication,” October 1948.