Hybrid Algorithm for Data Compression Using Genetic and Huffman Algorithm

Abstract

Text compression plays an important role and it is an essential object to decrease storage size and increase the speed of data transmission through communication channels .

In this research , a hybrid compression system is introduced which depends on the genetic algorithm for finding the best Huffman tree which give the best compression ratio, and then applying another compression method named Oring bits on the results of the primary compression by applying the compression method, and also using a decompression algorithms for both Huffman and Oring bits.

The variety in the text characters leads to the variety in the Huffman trees and finally obtaining the best possible compression . In addition, the increase in the frequencies of the text has an affect on the compression rate .

Structure of proposed system

Introduction

In this section, Figure (1) illustrates the block diagram for the proposed system and its algorithms, and it has reviewed the system mechanism in compression based on some examples, where it illustrates the compression operation through building the trees and codes and applying them .

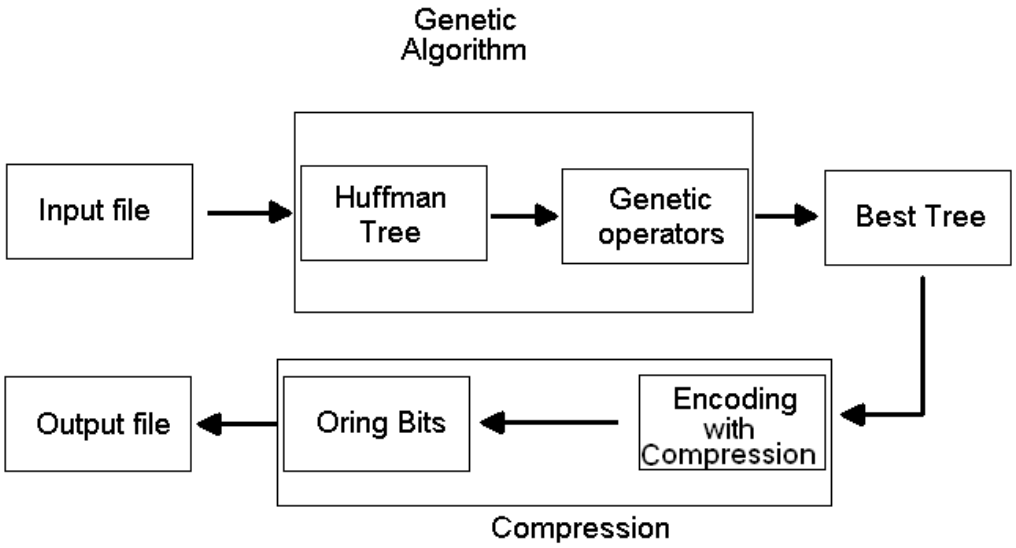


Figure (1) Block Daigram of Proposed System

Creating the Individual & the Initial Population

First, the text is wanted to be compressed and find the best Huffman tree for it is inputted from the notepad as a file. It has used text files which are read from the proposed system depending on the text; the initial population is created, where an individual is produced and from this individual (chromosome) the other individuals of the population are found randomly.

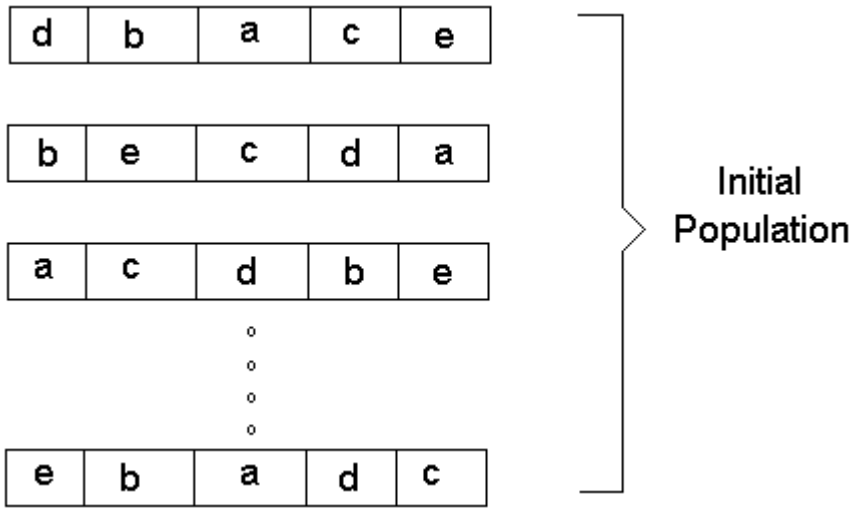
For example:

Text: abccddeeee

Produces the individual (chromosome)

a	b	c	d	e

And from this individual the initial population is produced randomly as follows:



The chromosome (individual) length depends on the variety in the text of the inputted characters. The frequency for each character is computed and then the probability for each character is found.

Building Huffman Tree

After generating the initial population, the Huffman tree is built for each individual in the population depending on their probabilities. The codeword is determined from the tree for each gene, for example in Figure (2):

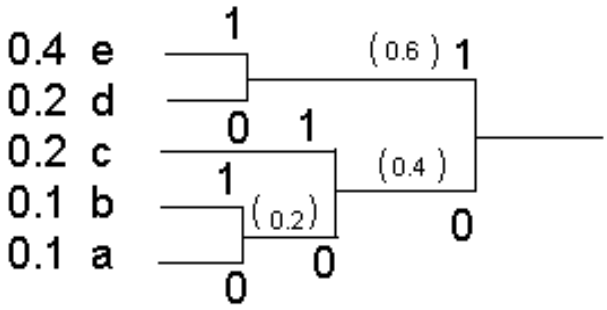


Figure (2) Huffman Codes

Where is:

a= 000 , b=001 , c=01 , d=10 , e=11

Evaluation

Every individual is evaluated by computing the fitness function. The fitness function here is the variance that depends on computing the average size for each individual.

$$A = \sum_{i=1}^{n} (P_i * a_i) \quad , \quad \text{……(1)}$$

$$\text{Variance} = \sum_{i=1}^{n} p_i (a_i - A)^2 \quad , \quad \text{…….(2)}$$

where,

p_i: probability

a_i: number of bits(length code word)

A: average size

For example:

The average size for the previous example shown in Figure (2) is computed as follows:

$$0.4*2 + 0.2*2 + 0.2*2 + 0.1*3 + 0.1*3 = 2.2$$

Then the variance is computed as follows:

$$0.4(2-2.2)^2 + 0.2(2-2.2)^2 + 0.2\ (2-2.2)^2 + 0.1(3-2.2)^2 + 0.1\ (3-2.2)^2 = 0.160$$

Tournament Selection

After the evaluation operation has been made, the two parents that will take place in the crossover operation are selected randomly. Two randomly individuals are chosen to produce the subpopulation; the individual with the least variance is selected to be the first parent. The operation is repeated to get the second parent according to the same parameters above.

The Figure (3) illustrates an example of a binary tournament selection. The population consists of a set of chromosomes whose genes are from the English alphabetic *A*.

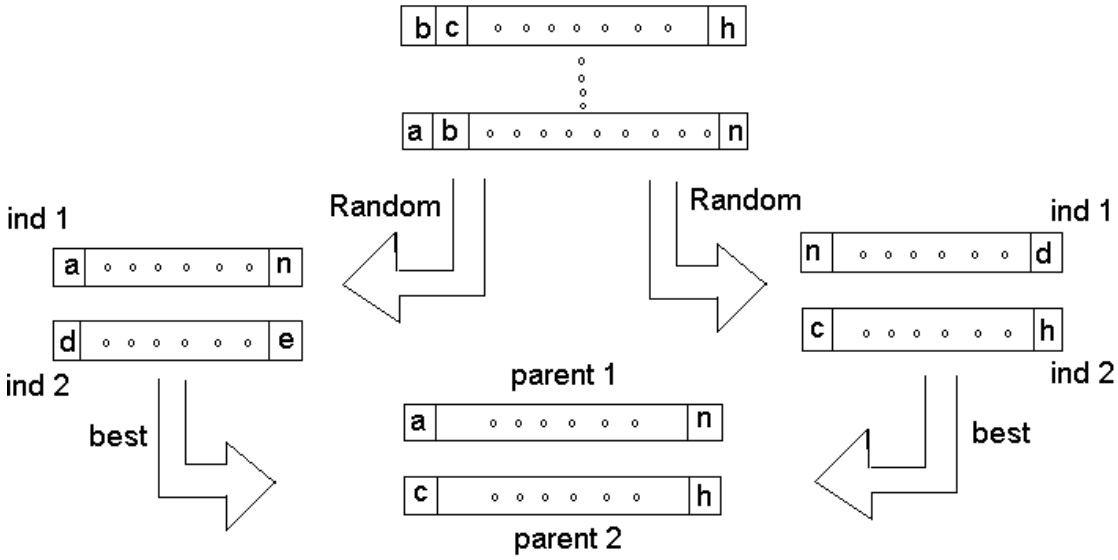


Figure (3) Binary Tournament Selection Example

Crossover Operation

In the proposed system, the cycle crossover (CX) has been chosen as one of the permutation crossover operators that mate the matching with the problem. This type of crossover gives a variety of

individuals; in addition it avoids the conflict in the genes constructing the chromosome (individual), which is the most important property that must be available in the proposed system.

The following example in Figure (4) describes that:

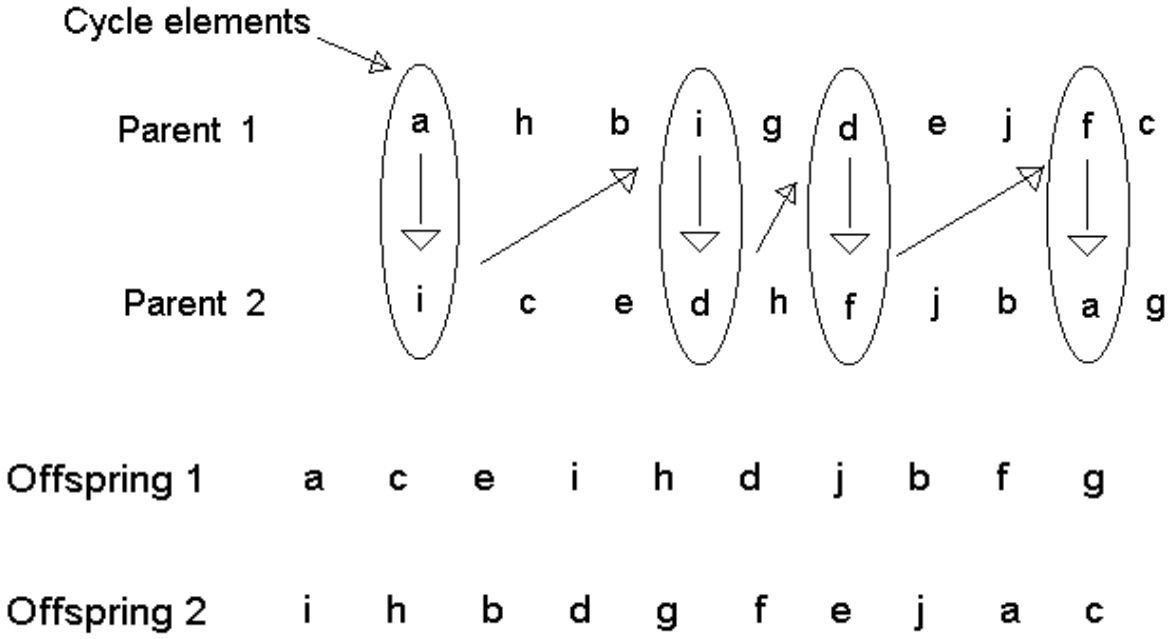
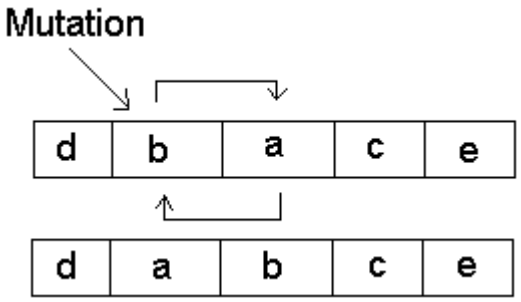


Figure (4) Crossover Operation

Mutation:

From computing the probabilities, it is determined whether there is mutation or not. The mutation probability (Pm) is 0.0009,if the gene's probability is less than or equal to the Pm then mutation occurs at that gene. On condition mutation occurs, the gene's location that happened at it the mutation is exchanged with the succeeding gene. If the mutation occurr at the last gene, in this condition this gene's location is exchanged with the first gene. Figure (5) illustrates the mutation operation:



Figure(5) Mutation Operation

4.7 Evaluate Offspring and Replacement

The new offspring produced by the crossover operation after building the Huffman tree for each new individual are evaluated and the fitness function is computed.

Then the new offspring is compared with the worst individuals in the population (biggest variance), the offspring are exchanged with the worst individuals in case the offspring is better than or equal to the selected individuals in order to get the best variety in the population.

In other cases, no exchange happens. After finishing the evaluation operation, there will be a new selection and crossover operation again and continue in this way.

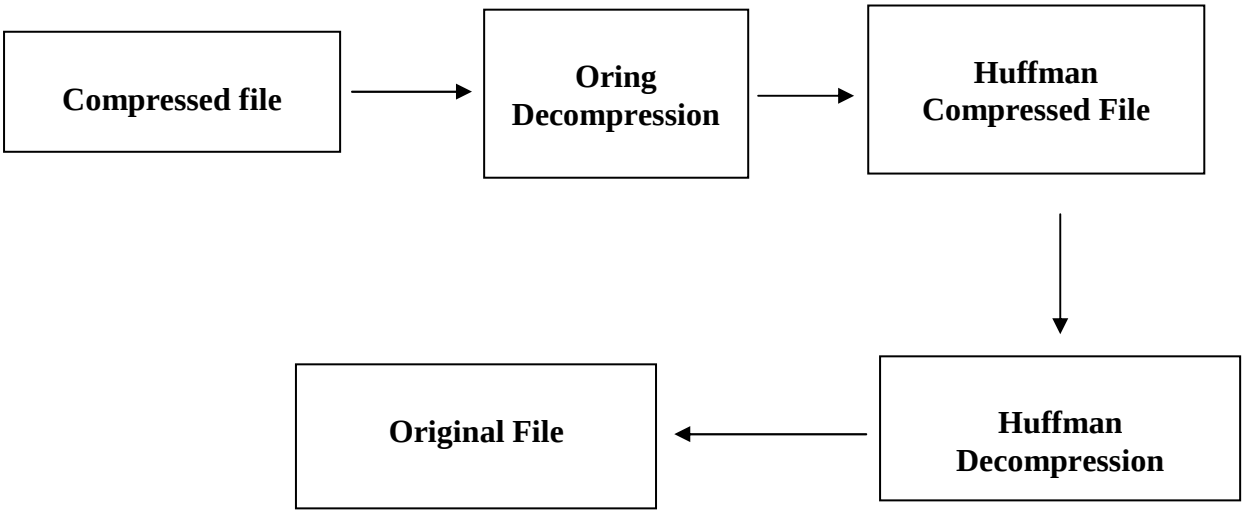
Stop Criteria

The operation including the crossover operation and generating a new population is continued for some generations (may be thousands of times).

After this, we will consider that one of the individuals with the best fitness (the least variance) prevalent in the population. The rate of prevailing may reach 50% or more and this rate is one of the stop criteria; the other stop criteria are the number of cycles (generations) that can reach 1000 generation.

As a result, the best individual is the most prevalent in the last population.

Decompression Process



Result Analysis

The number of generation is 200 and the number of individuals is 15.

abccddeeeeabccddeeeeabccddeeeeabccddeeee…….

Table (1) Allcode of Example 1.

Individual	Allcode	Variance
abcde	14	1.360
decba	13	0.960
ebdca	12	0.160

where the allcode represents the number of bits that the Huffman tree consists of.

The variance table (2) that is obtained after 300 generation is :-

Table (2)Allcode of Example 2.

Individual	Allcode	Variance
teydorclmnqwba	54	0.213
tyeqorclabnwmd	55	0.367
abcdeqwrtynmlo	56	0.521
endmylabwcrotq	57	0.675
mcrloaytdwqnbe	57	0.828
elnrtwbmaoqcdy	58	0.982

As it is noticed from the Table (2), the same allcode value gives two different variance values. In this case, the structure of the tree has an effect on the codeword length and finally has an effect on the variance.

Table (3) illustrates a set of examples taking into consideration different file sizes, while the chromosome length, population size, and generation number are fixed.

Table (3) Effect of Different File Size

	File name	File size	Chrom. Length	Population size	Generation No.	Compression Ratio
1	File 1	150B	5	15	200	63.333%
2	File 2	300B	5	15	200	68.000%
3	File 3	600B	5	15	200	70.333%

Table (4) illustrates a set of examples taking into consideration different chromosome lengths, while the other parameters are fixed.

Table (4) Effect of Different Chromosome Length

	File name	File size	Chrom. Length	Population size	Generation No.	Compression Ratio
1	File 3	600B	5	25	300	70.333%
2	File 6	600B	10	25	300	56.500%
3	File 11	600B	20	25	300	40.333%

Table (5) illustrates the effect of the probability of genes on the compression ratio. The increment in the probability of genes gives the gene the shortest codeword and as a result increases the compression ratio.

Table (5) Effect of Probability of Gene

	File name	File size	Chrom. Length	Population size	Generation No.	Compression Ratio
1	File 1	150B	5	15	200	63.333%
2	File12	100B	5	15	200	71.000%
3	File10	90B	5	15	200	71.111%

After applying the Oring bit algorithm on some of the above files, taking into consideration that the ratio of zeros in the file must be above 65% in order to get on good results.

The files that satisfied the condition and gave good results are illustrated in Table (6).

Table (6) Results of Oring Files

	File name	File size	Chrom. Length	Compression ratio without Oring	Compression ratio with Oring
1	File 6	600B	10	56.500 %	62.167 %
2	File 10	90B	5	70.103 %	77.320 %
3	File 9	884B	3	79.186 %	84.615 %

Also the proposed system has been applied on segments from DNA file taking different file sizes and results are viewed as follows:

Table (7) Results of DNA Files

File name	File size	Compression ratio
DNA 1	72 B	59.722%
DNA 2	153 B	67.320%
DNA 3	300 B	71.333%
DNA 4	542 B	72.878%

It has been noticed that there was not a large variety in the individuals because of the small number of genes (only four genes). It has also been noticed that it was difficult to get good results from applying the Oring on the DNA file because there was no occurrence of a large sequence of single genes.

Conclusions

1. **The subpopulation size has an effective role in reaching to the optimal solution; if the size of the subpopulation is more than 3, this will lead to premature conversion to a solution that may not be the optimal solution.**
2. **The same allcode value gives two different variance values. In this case, the structure of the tree has an effect on the codeword length and finally has an effect on the variance.**

3. **The number of individuals (population size) has an effect on obtaining the best tree where the small population size leads to a small variety and as a result leads to small crossover operations between the individuals because of getting the same first gene. As a result, the best individual (tree) will not be obtained.**
4. **The number of generations has an effect on reaching to the best Huffman tree where continuing in selecting various individuals gives a larger chance in selecting the best possible individual.**
5. **In order to get on the best compression for the Oring method when merging the Huffman compression method and Oring, the character with the largest probability takes the value '0' instead of the value '1' in building the Huffman tree.**
6. **The chromosome length has an effect on the compression ratio. Whenever the chromosome length was longer, the compression ratio is smaller. This happens because the increase in the size of the Huffman tree causes as a result an increase in the number of bytes transmitted through the transmission channel.**

Future Works

1- **Applying the meta -Genetic Algorithm on Genetic Algorithm (Optimization of GA).**

A meta-genetic algorithm has been used to optimize the genetic algorithm for cell placement. The three parameters optimized are the crossover rate, inversion rate and mutation rate. The meta-genetic algorithm is itself a genetic optimization process which runs the genetic algorithm to solve a placement problem and manipulates the genetic parameters to optimize the fitness of the genetic algorithm. The individuals in the

population of meta-genetic algorithm consist of three integers in the range [0,20], representing the mutation rate, inversion rate, and crossover rate for the genetic algorithm.

2- Using Breeder Genetic Algorithm (BGA):-

BGA represents a class of random optimization techniques gleaned from the science of population genetics, which have proved their ability to solve hard optimization problems with continuous parameters. BGA which can be seen as a recombination between Evaluation strategies (ES)and Genetic Algorithm(GA), uses truncation selection which is very similar to the (u ,λ)strategy in ESs and the search process is mainly driven by recombination making BGAs very similar to GAs. It has been proven that BGAs can solve problems more efficiently than GAs due to the theoretical faster convergence to the optimum and they can, like GAs, be easily written in a parallel form.

3- Applying the compression algorithm, the (Delta algorithm) on the research results for decreasing the cost of communication channels (reducing band width).

After finishing the compression operation and converting the sequence of binary numbers to integer numbers and before saving it as bytes, the delta algorithm is applying in order to reduce the value of the integer numbers and finally reducing the amplitude for the transmitted signal and this gives us a small band-width.

4- The proposed system can be applied on images by dealing with an

image as segments where Huffman algorithm is applied on each

segment.

REFERENCES

[1] A. Joglekar & others, "Genetic Algorithms and Their Use in the Design of Evolvable Hardware", Fr. Conceicao Rodrigues College of Engineering, Bandra, Mumbai - 400 050,IEEE, 1997.

[2] A. Khelfa , "A Genetic Clustering for Image Segmentation " , M.SC Thesis, April 2002.

[3] A. Armstrong and others, " Genetic Algorithms for Image Compression ", 2001 .

[4] A. W. berger and others," A Hybrid Coding Strategy for Optimized Test Data Compression", University of Innsbruck, Austria, Proceedings IEEE International Test Conference, Charlotte, NC, USA, September 30 – October 2, 2003.

[5] B. J. Schachter and others," Some Experiments in Image Segmentation by Clustering of Local Feature Values ", Pattern Recognition, 11, 1, 19, 1978.

[6] B. D. Davison and others, " Effect of Global Parallelsim on A Steady State GA ", Proceedings of The Evolutionary Computing and Parallel Processing Workshop (GECCO'99), Orlando, 1999.

[7] D. Whitley, "A Genetic Algorithm Tutorial " , Computer

Science Department Colorado State University, Fort Collins CO.,

1994.

[8] D. Salomon , " Data Compression the complete reference " spring verlag Newyourk ,USA , 1998 .

[9] F. Hansson ",Lempel-Ziv-Welch and Huffman compression ", 25 January 2004 .

[10] G.B. Parker and others, "Evolving Hexapod Gaits Using a Cyclic Genetic Algorithm " , Department of Computer Science Indiana University Bloomington, 1998.

[11] Goldberg D. E. , " Genetic Algorithms in Search, Optimization and Machine Learning " ,Addison-Wesley , 1989 .

[12] G. Kempe, "Computer Science Honours Research Report Compression and Computational Gene Finding" , 1November 2002.

[13] G. E. Blelloch "Introduction to Data Compression" Computer Science Department Carnegie Mellon University , October 16, 2001.

[14] H. J. Martinez and others, " Evolution of Cellular Automata for Digital Image Compression", de Ingenieria Universidad Central de Venezuela Caracas, Venezuela, 2006

[15] From Wikipedia, the free encyclopedia "Huffman coding ", GNU Free Documentation license, January 19,1996.

[16] I. Harvey ," The Microbial Genetic Algorithm ",School of Cognitive and Computing Sciences ,University of Sussex , Brighton BN1 9QH, UK, inmanh@cogs.susx.ac.uk.,1996.

[17] J. Socha, "Efficient Meta-Heuristics for Lossless Data Compression", School of Computer Science University of

Waterloo, Waterloo, Ontario, 2003.

[18] J. M. Pullen," Data Compression, Security Principles Integrity, Appropriate Use ",2/3/03 © 2003.

[19] J. Smith & others, " Replacement Strategies in Steady State Genetic Algorithms : Static Environment ", Foundation of Genetic Algorithms V, 219, 1998.

[20] M. Obitko ,"Introduction to Genetic Algorithms with Java Applets",1998.

[21] M. Sasikumar , "Genetic Algorithms ",NCST, Bombay.

[22] P. M. Elizabeth and others, "Genetic Algorithm for VLSI Design ,layout & test Automation ", 1999, prentice Hall , PTR USA.

[23] P. Busetti ,"Genetic algorithms overview",1995.

[24] R. C. Gonzalez and others, "Digital Image Processing", University of Tennessee, 1992.

[25] R. He ,"Indexing Compressed Text", A thesis,Waterloo, Ontario, Canada, 2003

[26] R. Müller , "Image Compression" Part II: Image Processing Computer Graphics and Image Processing , Winter Semester 2003/04.

[27] R. A. Watson and others, "Recombination Without Respect: Schema Combination and Disruption in Genetic Algorithm Crossover " , Volen Center for Complex Systems, Brandeis University, Waltham, 2000 .

[28] Sheila Horan,"Data Compression Statistics and Implications ", New Mexico State University, chapter 27- Data compression, 1997.

[29] S. W. Smith, A sample chapter from: The Scientist and

Engineer's Guide to Digital Signal Processing", 1997 .

[30] T. A. Abbas Al- Asadi "A Hybrid Algorithm For Images Compression" , Ph.D. Thesis, Computer Science ,

[31] T. H. Al-Kafaji University of Technology, 2004 . "Improving Document Retrieval Using Genetic Algorithm" , M.SC. Thesis, September, 2000.

[32] U. Aickelin , "Solving Multiple-Choice Problems with Genetic Algorithms", School of Computer Science University of Nottingham UK uxa@cs.nott.ac.uk.

[33] W. M. Spears "Adapting Crossover in a Genetic Algorithm " , Washington, D.C. USA , 1995.

[34] W. KEONG NG," Lossless and Lossy Data Compression " ,Nanyang Technological University ,Singapore, 1996.

[35] W. J. Mason and others ," Optimal Earth Orbiting Satellite Constellations Via a Pareto Genetic Algorithm" Department of Aeronautical and Astronautical Engineering University of Illinois at Urbana-Champaign Urbana,1998.

[36] From Wikipedia, the free encyclopedia," Genetic algorithm" , GNU Free Documentation license, June 30, 2005.