A New Initialization Method and a New Update Operator for Quantum Evolutionary Algorithms in Solving Fractal Image Compression

M. H. Tayarani N., M. Beheshti, J. Sabet, M. Mobasher, H. Joneid
University of Southampton
Khayyam Institute of Higher Educations. U.K
mhtn1g09@ece.soton.ac.uk

ABSTRACT: *Fractal Image Compression (FIC) problem is a combinatorial problem which has recently become one of the most promising encoding technologies in the generation of image compression. While Quantum Evolutionary Algorithm (QEA) is a novel optimization algorithm proposed for class of combinatorial optimization problems, it is not widely used in Fractal Image Compression problem yet. Using statistical information of range and domain blocks, and a novel magnetic update operator, this paper proposes a new algorithm in solving FIC. The statistical information of domain and range blocks is used in the initialization step of QEA. In the proposed update operator the q-individuals are some magnetic particles applying attractive force to each other. The force two particles apply to each other depends on their fitness and their distance. The proposed algorithm is tested on several images and the experimental results show better performance for the proposed algorithm than QEA and GA. In comparison with the full search algorithm, the proposed algorithm reaches comparable results with much less computational complexity.*

Keywords: Optimization Algorithms, Quantum Evolutionary Algorithms, Fractal Image Compression

Received: 18 March 2011, Revised 25 April 2011, Revised 29 April 2011

1. Introduction

Fractal Image Compression, proposed by Barnsley has recently become one of the most promising encoding technologies in the generation of image compression [1]. The high compression ratio and the quality of the retrieved images attract many of researchers, but the high computational complexity of the algorithm is its main drawback. One way of decreasing the time complexity is to move from full search method to some optimization algorithms like Genetic Algorithms. From this point of view, several works try to improve the performance of fractal image compression algorithms using Genetic algorithm. In [2] a new method for finding the IFS code of fractal image is developed and the influence of mutation and the crossover is discussed. The low speed of fractal image compression blocks its way to practical applications. In [3] a genetic algorithm approach is used to improve the speed of searching process in fractal image compression. A new method for genetic fractal image compression based on an elitist model is proposed in [4]. In the proposed approach the search space for finding the best self similarity is greatly decreased. Reference [5] makes an improvement on the fractal image coding algorithm by applying genetic algorithm. Many researches increase the speed of fractal image compression but the quality of the image will decrease. In [6] the speed of fractal image compression is improved without significant loss of image quality. Reference [7] proposes a genetic algorithm approach which increases the speed of the fractal image compression without decreasing of the quality of the image. In the proposed approach a standard Barnsley algorithm, the Y. Fisher based on classification and the genetic compression algorithm with quad-tree partitioning are compared. In GA based algorithm a population of transformations is evolved for each range block. In order to prevent the premature convergence of GA in fractal image compression a new approach is proposed in [8], which controls the parameters of GA adaptively. A spatial correlation genetic algorithm is proposed in [9], which speeds up the fractal image compression algorithm. In the proposed algorithm there are two stages, first the spatial correlations in image for both the domain pool and the range pool is performed to exploit local optima. In the second stage if the local optima were not satisfying, the whole image is searched to find the best self similarity. A schema genetic algorithm for fractal image compression is proposed in [10] to find the best self similarity in fractal image compression.

Using statistical information of the image, this paper proposes a novel initialization method for QEA. The proposed algorithm has two steps. At the first step the statistical information of the image is extracted. The information consists of the variance of every domain and range block in the image. Then the best region for each range block is found. In the second step, for each range block, QEA searches among domain pool and finds the best domain block and transformation. Here the information gathered in the first step is used to find the best initialization for quantum bits. This paper also proposes a new update operator for QEA, inspiring magnetic field theory which offers more interaction among q-individuals and binary solutions. In the proposed algorithm, the binary solutions with higher fitness apply stronger force to the q-individuals. Performing this new method, the q-individuals have more chance to find better solutions in less time. The proposed algorithm is tested on several images and experimental results show better performance for the proposed algorithm than GA and original version of QEA.

The rest of the paper is organized as follows. Section 2 introduces QEA, in section 3 the new algorithm is proposed and in section 4 is experimented on several images and finally section 5 concludes the paper.

2. Quantum Evolutionary Algorithm

QEA is inspired from the principles of quantum computation, and its superposition of states is based on qubits, the smallest unit of information stored in a two-state quantum computer. A qubit could be either in state “0” or “1”, or in any superposition of the two as described below:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (1)$$

Where α and β are complex numbers, which denote the corresponding state appearance probability, following below constraint:

$$|\alpha|^2 + |\beta|^2 = 1 \quad (2)$$

This probabilistic representation implies that if there is a system of *m* qubits, the system can represent 2^m states simultaneously. At each observation, a qubits quantum state collapses to a single state as determined by its corresponding probabilities.

Consider *i-th* individual in *t-th* generation defined as an *m*-qubit as below:

$$\begin{bmatrix} \alpha_{i1}^t & \alpha_{i2}^t & \dots & \alpha_{ij}^t \dots \alpha_{im}^t \\ \beta_{i1}^t & \beta_{i2}^t & \dots & \beta_{ij}^t \dots \beta_{im}^t \end{bmatrix} \quad (3)$$

Where $|\alpha_{ij}^t|^2 + |\beta_{ij}^t|^2 = 1$, j = 1, 2, ,*m*, *m* is the number of qubits, i.e., the string length of the qubit individual, i = 1,2, ,*n*, *n* is thenumber of possible solution in population and *t* is generation number of the evolution. If there is, for instance, a three-qubits ($m = 3$) individual such as 4.

$$q_i^t = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} & \frac{1}{2} \\ \frac{1}{\sqrt{2}} & \frac{\sqrt{2}}{\sqrt{3}} & \frac{\sqrt{3}}{2} \end{bmatrix} \quad (4)$$

Or alternatively, the possible states of the individual can be represented as:

$$q_{ij}^t = \frac{1}{2\sqrt{6}}|000\rangle + \frac{1}{2\sqrt{2}}|001\rangle + \frac{1}{2\sqrt{3}}|010\rangle + \frac{1}{2}|011\rangle + \frac{1}{2\sqrt{6}}|100\rangle + \frac{1}{2\sqrt{2}}|101\rangle + \frac{1}{2\sqrt{3}}|110\rangle + \frac{1}{2}|111\rangle \quad (5)$$

In QEA, only one qubit individual such as 4 is enough to represent eight states, whereas in classical representation eight

In the initialization step of QEA, $[\alpha_{ij}^{t}\beta_{ij}^{t}]^{T}$ of all q_i^0 are initialized with $\frac{1}{\sqrt{2}}$. This implies that each qubit individual q_i^0 represents the linear superposition of all possible states with equal probability. The next step makes a set of binary instants, x_i^t by observing $Q(t) = \{q_1^t, q_2^t, ..., q_n^t\}$ states, where $X(t) = \{x_1^t, x_2^t, ..., x_n^t\}$ at generation t is a random instant of qubit population. Each binary instant, x_i^t of length m, is formed by selecting each bit using the probability of qubit, either $|\alpha_{ij}^t|$ or $|\beta_{ij}^t|$ of q_i^t. Each instant x_i^t is evaluated to give some measure of its fitness. The initial best solution $b = max_{i=1}^{n}\{f(x_i^t)\}$ is then selected and stored from among the binary instants of $X(t)$. Then, in 'update' $Q(t)$, quantum gates U update this set of qubit individuals $Q(t)$ as discussed below. This process is repeated in a while loop until convergence is achieved. The appropriate quantum gate is usually designed in accordance with problems under consideration.

x_i	b_i	$f(x) \geq f(b)$	$\boldsymbol{\Delta\theta}$
0	0	false	0
0	0	true	0
0	1	false	0.01p
0	1	true	0
1	0	false	-0.01π
1	0	true	0
1	1	false	0
1	1	true	0

Table 1. Lookup Table of $\boldsymbol{\Delta\theta}$, the rotation gate. x_i is the ith bit of the observed binary solution and b_i is the ith bit of the best found binary solution

In QEA, the quantum bit representation can be simply interpreted as a biased mutation operator. Therefore, the current best individual can be used to steer the direction of this mutation operator, which will speed up the convergence. The evolutionary process of quantum individual is completed through the step of "update $Q(t)$". A crossover operator, quantum rotation gate, is described below. Specifically, a qubit individual q_i^t is updated by using the rotation gate $U(q)$ in this algorithm. The *j-th* qubit value of *i-th* quantum individual in generation t, $[\alpha_{ij}^t \beta_{ij}^t]^T$ is updated as:

$$\begin{bmatrix}\alpha_{ij}^{t}\\ \beta_{ij}^{t}\end{bmatrix} = \begin{bmatrix}\cos(\Delta\theta) - \sin(\Delta\theta)\\ \sin(\Delta\theta) - \cos(\Delta\theta)\end{bmatrix}\begin{bmatrix}\alpha_{ij}^{t-1}\\ \beta_{ij}^{t-1}\end{bmatrix} \quad (6)$$

Where $\Delta\theta$ is rotation angle and controls the speed of convergence and determined from Table 1. Reference [11] shows that these values for $\Delta\theta$ have better performance.

3. The Proposed Method

Fractal Image Compression algorithms search all the domain pool for each range block and find the best domain block with best transform which matches the range block. Conventional fractal image compression algorithms use a full search algorithm in domain pool which is a time consuming procedure. Evolutionary algorithms are suitable for this problem because they can find an appropriate domain block with a transform without performing a full search in domain pool. Since Quantum Evolutionary Algorithms are proposed for combinatorial problems like knapsack problem [14] and fractal image compression is in the class of NP-Hard problems, QEA is highly suitable for FIC.

Initialization is an important part of Evolutionary Algorithms. By initializing possible solutions with appropriate values, we can lead the algorithm to search better parts of the search space and help it to reach better solutions with less computation time.

Since QEA is a new method, a small number of works have focused on initialization step in this algorithm and no research has worked on the initialization step in fractal image compression. This paper proposes a novel initialization method for QEA in fractal image compression which uses statistical information of the domain pool to find the best region for each range block.

Searching among all the domain pool for each range block is time consuming and there is a need to some new methods to help the Evolutionary Algorithms finding better solution with higher speed. There are several works in the fractal image compression field that use the variance of domain blocks and range blocks to speed up the search process. Here the statistical information of domain pool is used in the initialization step.

The domain blocks are coded by their horizontal and vertical address in the image. Therefore a solution is a binary string having 3 parts, *px*, *py*, *pT*, representing the horizontal and vertical location of domain blocks in the image and the transformations respectively. The length of the possible solution for a $M \times N$ image is:

$$m = [\log_2(M)] + [\log_2(N)] + 3 \qquad (7)$$

Where m is the size of the possible solutions. Here 8 ordinary transformation are considered: rotate 0^o, 90^o, 180^o, 270^o, flip vertically, horizontally, flip relative to 45^o, and relative to 135^o. The procedure of the proposed method is as follows:

Proposed Algorithm

1. Find the variance of all the domain and range blocks.
2. For each range block do
   begin
3. Find the n nearest domain blocks to the range block. Code the n domain blocks.
4. Initialize the n q-individuals in Q^0, based on the coded domain blocks.
   t = 0
5. while not termination condition do
   begin
   $t = t + 1$
6. make X^t by observing the states of Q^{t-1}
7. evaluate the particles in X^t and store their performance in magnetic fields B^t
8. normalize B^t according to 13
9. evaluate the mass M^t for all particles according to 14
10. for all q-individuals q_i^t in Q^t do
    begin
11. $F_i = 0$
12. find N_i
13. for all x_u^t in N_i do
14. $F_{ij} = F_{ij} + \dfrac{(x_{uj}^t - (\beta_{ij}^t)^2) \times B_u^t}{D(x_i^t, x_u^t)}$
    end
15. for all q-individuals q_i^t in Q^t do
    begin
16. $v_{ij}^{t+1} = \dfrac{F_{ij}}{\eta \times M_i}$
17. $q_{ij}^{t+1} = q_{ij}^{t+1} + v_{ij}^{t+1}$

end

end

end

QEA has a population of quantum individuals $Q(t) = \{q_1^t, q_2^t, ..., q_n^t\}$, where t is generation step and n is the size of population.

The description of the proposed algorithm is as follows:

1. In this step the variance of all the domain and range blocks is calculated.

2. This step performs a search on domain pool for each range block.

3. Here the variance of the blocks is used to find the similarity between range and domain blocks. If the domain pool is D and D_k is the k-th domain block in domain pool, then the most similar domain block to the range block R_i is:

$$S_1 = \left\{ D_j \mid D_j \in D, \forall k \neq j, D_k \in D \Rightarrow \left| v(R_i) - v(D_j) \right| \leq \left| v(R_i) - v(D_k) \right| \right\} \quad (8)$$

The second most similar domain block to the range block R_i is:

$$S_1 = \left\{ D_j \mid D_j \in D, \forall k \neq j, D_k \in D \Rightarrow \left| v(R_i) - v(D_j) \right| \leq \left| v(R_i) - v(D_k) \right| \right\} \quad (9)$$

And the n-th most similar domain block to the range block R_i is:

$$S_n = \left\{ D_j \mid D_j \in D - \bigcup_{i=1}^{n-1} S_i, \forall k \neq j, D_k \in D - \bigcup_{i=1}^{n-1} S_i \Rightarrow \left| v(R_i) - v(D_j) \right| \leq \left| v(R_i) - v(D_k) \right| \right\} \quad (10)$$

Where n is the size of the population in QEA and S_n contains the n-th most similar domain blocks to the range block R_i. After finding the n most similar domain blocks, these domain blocks, $S_1, S_2, ..., S_n$ are coded as binary solutions and stored to $Y_1, Y_2, ..., Y_n$ to be used for the initialization step.

4. This step is the initialization step of QEA. In this step the n q-individuals in the population are initialized based on the n most similar domain blocks found in the previous step. In the proposed algorithm the address part of q-individuals $Q(0)$, px, py are found as follows:

$$q_{ij}^0 \begin{cases} \left[\sqrt{1 - \frac{i-1}{2(n-1)}} \quad \sqrt{\frac{i-1}{2(n-1)}} \right]^T & if\ Y_{ij} = 0 \\ \left[\sqrt{\frac{i-1}{2(n-1)}} \quad \sqrt{1 - \frac{i-1}{2(n-1)}} \right]^T & if\ Y_{ij} = 1 \end{cases} \quad (11)$$

Where Y_{ij} is the j-th bit of the code of ith most similar domain block to the range block, $i = 1, 2, ..., n$, n is the size of the population in QEA, m is the size of qindividuals, $j = 1, 2, ..., m - 3$. The bits in p_t part of q_{ij}^0 are set to: $q_{ij}^0 = \left[\frac{\sqrt{2}}{2} \quad \frac{\sqrt{2}}{2} \right]^T$

For i = 1, 2, ,n, and $j = m - 2, m - 1, m$.

5. This step makes a set of binary instants $X^t = \{x_i^t \mid i = 1,2,..., n\}$ at generation t by observing $Q^{t-1} = \{q_i^{t-1} \mid i = 1,2,..., n\}$ states, where X^t at generation t is a random instant of qubit population and n is the size of population. Each binary instant, x_i^t of length m, is formed by selecting each bit using the probability of qubit, either $|\alpha_{ij}^{t-1}|^2$ or $|\beta_{ij}^{t-1}|^2$ of q_i^{t-1} .Observing the binary bit x_{ij}^t from qubit $[\alpha_{ij}^t \ \beta_{ij}^t]^T$ performs as:

$$x_{ij}^t = \begin{cases} 0 & if\ R(0,1) < |\alpha_{ij}^t|^2 \\ 1 & otherwise \end{cases} \quad (12)$$

Where R(.,.) is a uniform random number generator.

6. Each binary instant x_i^t is evaluated to give some measure of its objective. In this step, the fitness of all binary solutions of X^0 are evaluated and stored in B^t .

7. Next the normalization is performed on B^t . The normalization is performed as:

$$B_i = \frac{B_i - Min}{Max - Min} \tag{13}$$

Where: $Min = minimum_{i=1}^{n}(B_i^t)$, $Max = maximum_{i=1}^{n}(B_i^t)$

The magnetic field of each particle is normalized in the range of [0-1]. This is because the fitness values of possible solutions are problem dependent. The range of the fitness of the possible solutions can be in any range, since the amount of the magnetic field controls the movement of the particles, we normalize the amount of magnetic field.

8. In this step the mass of all particles is calculated and stored in M^t:

(14)

9. In this step in the "for" loop, the resultant force of all forces on each particle is calculated.

10. At first the resultant force F_i to particle x_i^t is initialized to zero.

11. In the proposed algorithm, each particle interacts only with its neighbors i.e. each particle applies its force only to its neighbors. In this step the neighbors of x_i^t are determined randomly. Each possible solution has 4 neighbors.

12. In this step, the applied force to particle x_i^t by its neighbor's x_u^t, $\forall x_u^t \in N_i$ is calculated.

13. The force which is applied by x_u^t to x_i^t relates to the distance between two particles and the magnetic field of x_u^t. It is calculated as:

$$F_{ij} = \frac{\left(x_{uj}^t - (\beta_{ij}^t)^2\right) \text{x } B_u^t}{(x_i^t, x_u^t)} \tag{15}$$

Here F_i shows the force applied to q-individual q_i^t. The part "$x_{uj}^t - (\beta_{ij}^t)^2$" shows the direction which the q-individual moves and $(\beta_{ij}^t)^2$ is the probability of q_i^t representing state "1". Where D(.,.) is the distance between each pair of neighboring particles and is calculated as:

$$\mathrm{D}(x_j^t, x_u^t) = \frac{1}{m}\sum_{j=1}^{m} |x_{ij}^t - x_{uj}^t| \tag{16}$$

Where x_i^t and x_u^t are *i-th* and *u-th* binary solutions of the population in iteration *t* respectively and x_{ij}^t is the *j-th* bit of ith binary solution in iteration *t*. This step is the main step in the proposed algorithm.

14, 15, 16. In these steps the location of q-individuals are updated. The proposed update operator has two advantages. First according to 15 the observed binary solutions with higher fitness have bigger magnetic field ***B*** and apply stronger force to the qindividual, therefore the better binary solutions exert stronger attraction force. Here unlike Q-Gate the movement of q-individuals is not constant throughout the search process and varies for various q-individuals and even various dimensions. Second in the proposed update operator even the inferior binary solutions have effect on the qindividuals but with smaller amplitude. Accordingly the interaction among possible solutions is much more than Q-Gate and the inferior binary solutions participate in the search process.

3.1 Parameter tuning

As it is seen in step 16 of the algorithm, the proposed algorithm has a parameter of *h*. This section tries to find the best parameter for the proposed update operator. Since fractal image compression is a time consuming algorithm and finding the best parameter needs several experiments for several values of the parameter, this section finds the best parameter for some benchmark functions which are much faster. The size of population for all the experiments is set to 25, and the parameter is set to η = (1, 2, 3, 4, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50). Figure 1 shows the parameter setting for the proposed algorithm on Knapsack problem and Generalized Schwefel's Function 2.26. The results are averaged over 30 runs. According to Figure 1, the best parameter for Knapsack problem repair type 1, the best parameter is $\eta = 5$, the best parameter for Knapsack penalty type 2 is $\eta = 20$ and the best parameter for Generalized Schwefel is 10. This paper finds the best parameter for the proposed update operator for several benchmark functions and the results are summarized in Table 2. As it is clear in Table 2, for all the numerical function problems the best parameter for the proposed update operator is 10.

Problem	η	Problem	η	Problem	η	Problem	η
Kpck Rep 1	5	Kpck Rep 2	35	Kpck Pen 1	5	Kpck Pen 2	20
Trap	2	Schwefel	10	Rastrigin	10	Ackley	10
Griawank	10	Penalized1	10	Penalized2	10	Kennedy	10
Michalewicz	10	Goldberg	10	Sphere	10	Rosenbrock	10
Schwefel 2.21	10	Dejong	10	Schwefel 2.22	10		

Table 2. Best parameter for the proposed Update operator. The results are averaged over 30 runs

4. Experimental Results

This section experiments the proposed algorithm and compares the proposed algorithm with the performance of GA and original version of QEA in fractal image compression. The proposed algorithm is examined on images Lena, Pepper and Baboon with the size of 256×256 and gray scale. The size of range blocks is considered as 88 and the size of domain blocks is considered as 16×16. In order to compare the quality of results, the PSNR test is performed:

$$PSNR = 10 \times \log\left(\frac{255^2}{\frac{1}{M \times N}\sum_{i=1}^{N}\sum_{j=1}^{M}(f(i,j) - g(i,j))^2}\right) \tag{17}$$

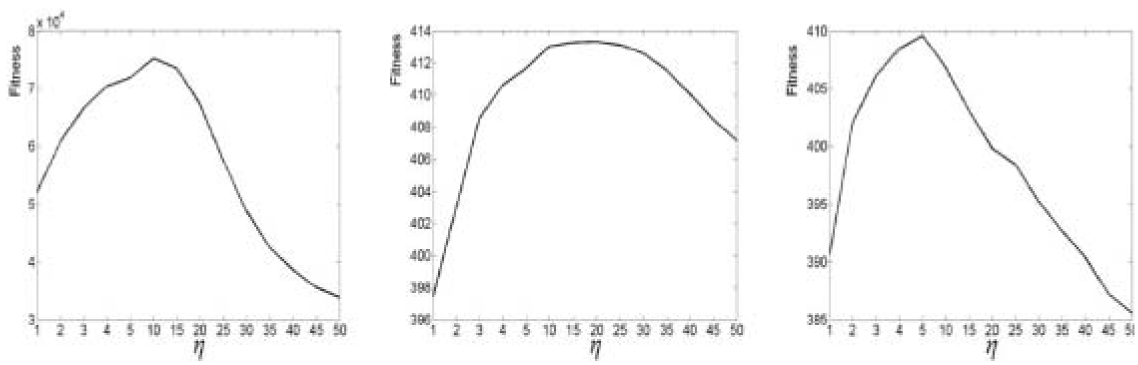

Figure 1. The effect of the parameter η on the performance of the proposed algorithm on Generalized Schwefel's Function 2.26, Knapsack problem Penalty Type 2 and Trap problem.

Where $M \times N$ is the size of image. The crossover rate in GA is 0.8 and the probability of mutation is 0.003 for each allele. Table 3 shows the experimental results using proposed algorithm and GA. The number of iterations for GA, QEA and the proposed algorithm for all the experiments is 200. The parameter η is considered as $\eta = 10$. According to Table 3 the proposed algorithm improves the performance of fractal image compression for all the experimental results.

Picture	Method	Pop Size	MSE Computations	PSNR
Lena	Full Search	-	59,474,944	28.85
	QEA	30	6,144,000	28.49
		25	5,120,000	28.28
		20	4,096,000	28.95
		15	3,072,000	27.43
	Proposed Method	30	6,144,000	28.59
		25	5,120,000	28.42
		20	4,096,000	29.04
		15	3,072,000	27.57
	GA	30	6,144,000	28.11
		25	5,120,000	28.04
		20	4,096,000	27.55
		15	3,072,000	27.27
Pepper	Full Search	-	59,474,944	29.85
	QEA	30	6,144,000	29.55
		25	5,120,000	29.09
		20	4,096,000	28.87
		15	3,072,000	28.12
	Proposed Method	30	6,144,000	29.65
		25	5,120,000	29.29
		20	4,096,000	28.92
		15	3,072,000	28.52
	GA	30	6,144,000	29.14
		25	5,120,000	28.92
		20	4,096,000	28.64
		15	3,072,000	28.11
Baboon	Full Search	-	59,474,944	20.04
	QEA	30	6,144,000	19.28
		25	5,120,000	19.18
		20	4,096,000	18.95
		15	3,072,000	18.62
	Proposed Method	30	6,144,000	19.65
		25	5,120,000	19.33
		20	4,096,000	19.07
		15	3,072,000	18.73
	GA	30	6,144,000	19.17
		25	5,120,000	19.02
		20	4,096,000	18.65
		15	3,072,000	18.41

Table 3. Experimental results on Lena, Pepper, and Baboon. The results are an average over 10 runs

5. Conclusion

In order to improve the performance of QEA in solving Fractal Image Compression problem, this paper proposes a novel magnetic update operator which works better than conventional q-gate update operator. This paper also uses the statistical information of range and domain blocks to lead the possible solutions toward better parts of the search space to help them finding better solutions. Finally experiments on Lena, Pepper, and Baboon pictures show an improvement on evolutionary algorithms solving fractal image compression.

References

[1] Xing-yuan, W., Fan-ping, L., Shu-guo, W. (2009). Fractal image compression based on spatial correlation and hybrid genetic algorithm, Elsevier, *Journal of Vis.Ccommun.Iimage R*. p. 505-510.

[2] Yang Xuan, Liang Dequn. (1996). An improved genetic algorithm of solving IFS code of fractal image, *In:* IEEE 3rd international conference on signal processing.

[3] Chen, X., Zhu, G., Zhu, Y. (1998). Fractal image coding method based on genetic algorithms, *In:* International Symposium on Multispectral Image Processing.

[4] Mitra, S. K., Murthy, C. A., Kundu, M. K. (1998). Technique for fractal image compression using genetic algorithm, IEEE Transon Image Processing, 7 (4) 586-593.

[5] Xun, L., Zhongqiu, Y. (2000). The application of GA in fractal image compression, *In:* 3rd IEEEWorld Congress on Intelligent Control and Automation.

[6] Gafour, A., Faraoun, K., Lehirech. (2003). A Genetic fractal image compression, *In:* ACS/IEEE International Conference on Computer Systems and Applications.

[7] Mohamed, F. K., Aoued. (2005). Speeding Up Fractal Image Compression by Genetic Algorithms, *Journal of Multidimension Systems and Signal processing*, 16(2).

[8] Xi, L., L. Zhang, L. (2007). A Study of Fractal Image Compression Based on an Improved Genetic Algorithm, *International Journal of Nonlinear Science*, 3(2) 116-124m.

[9] Wu, M., Teng, W., Jeng, J., Hsieh, J. (2006). Spatial correlation genetic algorithm for fractal image compression, *Journal of Chaos, Solitons and Fractals*, 28 (2) 497- 510.

[10] Wu, M., Teng, W., Jeng, J., Hsieh, J. (2007). Schema genetic algorithm for fractal image compression, *Journal of Engineering Applications of Artificial Intelligence*, 20 (4) 531-538.

[11] Han, K., Kim, J. (2002). Quantum-inspired evolutionary algorithm for a class of combinatorial optimization, *IEEE Transactions on Evolutionary Computing*, 6 (6).

[12] Han, K., Kim, J. (2004). Quantum-Inspired Evolutionary Algorithms with a New Termination Criterion, He Gate, and Two-Phase Scheme, *IEEE Trans. on Evolutionary Computation*, 8 (2).

[13] Zhong, W., Liu, J., Xue, M., Jiao, L. (2004). A Multi-agent Genetic Algorithm for Global Numerical Optimization, *IEEE Transactions on Sys, Man and Cyber*, 34, 1128-1141.