

USING TRIANGULAR FUNCTION TO IMPROVE SIZE OF POPULATION IN QUANTUM EVOLUTION ALGORITHM FOR FRACTAL IMAGE COMPRESSION

Amin Qorbani¹, Ali Nodehi² and Saeed Nodehi³

¹Department of Computer Engineering, Kordkoy Branch, Islamic Azad University
a.qorbani@hotmail.com

²Department of Computer Engineering, Gorgan Branch, Islamic Azad University
alnodehi@hotmail.com

³Department of Computer Engineering, Gorgan Branch, Mirdamad University
s_nodehi@hotmail.com

ABSTRACT

Fractal Image Compression is a well-known problem which is in the class of NP-Hard problems. Quantum Evolutionary Algorithm is a novel optimization algorithm which uses a probabilistic representation for solutions and is highly suitable for combinatorial problems like Knapsack problem. Genetic algorithms are widely used for fractal image compression problems, but QEA is not used for this kind of problems yet. This paper improves QEA whit change population size and used it in fractal image compression. Utilizing the self-similarity property of a natural image, the partitioned iterated function system (PIFS) will be found to encode an image through Quantum Evolutionary Algorithm (QEA) method. Experimental results show that our method has a better performance than GA and conventional fractal image compression algorithms.

KEYWORDS

Optimization Method, Quantum Evolutionary Algorithms, Genetic Algorithms, Fractal Image Compression

1. INTRODUCTION

Fractals exist in nature widely. Fractal images contain few amount of information, but posses high-level of visual complexity [1]. Fractal image compression is potentially a great coding scheme since it features a high compression ratio and good retrieved image quality. In recent years, many researchers have studied and improved the fractal image encoding and have gotten a lot of achievements. In 1988, the fractal image compression was firstly proposed and utilized a number of affine mappings to denote the original image. Those mappings are iterated convergent and their limit is very close to the original image. In 1992, Jacquin introduced an automatic encoding algorithm for the first time that called baseline fractal image compression or BFC [2–6]. This method breaks the original image into sub-blocks and needs to find the best matched sub-blocks according to self-similarities in the image. BFC makes the fractal image encoding become a very hopeful technique to improve the storage schemes applied in the consumer electronics. Although BFC is very charming, a great deal of time cost during encoding limits it to widely practical applications. In order to solve this problem some improved approaches have been presented. In [6-7] authors proposed classification methods base on the feature of domain

blocks. [8] Proposed a kind of neighborhood matching method based on spatial correlation which makes use of the information of matched range blocks and effectively reduced the encoding time.

Quantum Evolutionary Algorithms are novel algorithms using probabilistic representation for possible solutions. This algorithm used in several problem but is not use in fractal image compression. Genetic algorithm is widely used in fractal image compression. In [9] 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 application. In [10] a genetic algorithm approach is used to improve the speed of searching in fractal image compression. A new method for genetic fractal image compression based on an elitist model in proposed in [11]. In the proposed approach the search space for finding the best self-similarity is greatly decreased. Reference [12] 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 [13] the speed of fractal image compression is improved without significant loss of image quality.

Reference [14] 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 in 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 [15], which controls the parameters of GA adaptively. A spatial correlation genetic algorithm is proposed in [16], 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 certifiable, the whole of image is searched to find the best self-similarity. A schema genetic algorithm for fractal image compression is proposed in [17] to find the best self-similarity in fractal image compression.

Size of the population is an effective parameter of the evolutionary algorithms and has a great role on the performance of EAs. Several researches investigate the effect of population size and try to improve the performance of EAs with controlling the size of the population. A functional sized population GA with a periodic function of saw-tooth function is proposed in [21]. Reference [22] finds the best population size for genetic algorithms. Inspired by the natural features of the variable size of the population [19,23] presents an improved genetic algorithm with variable population-size. In [24] an adaptive population size for the population is proposed for a novel evolutionary algorithm.

In this paper, to improve performance of fractal image compression, a triangular function which changes size of population in each iteration is used. This paper is organized as follow: Section 2 introduces the basics theoretical of fractal image compression. In Section 3, we describe QEA and then introduce triangular function for population size in QEA for fractal image compression in section 4. Section 5 shows the experimental results and finally Section 6 concludes the paper.

2. THEORETICAL BASICS OF FRACTAL IMAGE COMPRESSION

The fractal image compression is based on the local self-similarity property and PIFS. The related definitions and theorems are stated as follows [5-6]:

Definition 2.1. Let X be a metric space with metric d_x . A map $w: X \rightarrow X$ is Lipschitz with Lipschitz factor s , if there exists a real value s such that

$$d_X(w(x),w(y)) \leq s d_X(x,y), \quad 0 < s < 1, \quad \forall x,y \in X \quad (1)$$

We also say that w is contractive with contractivity s .

Definition 2.2. Let X be a metric space and d_x be its metric. For a point $x \in X$ and a nonempty set $A \subseteq X$, let us first define the distance of x to A by

$$d_y(x,A) = \inf_{a \in A} d_x(x,a) \quad (2)$$

Then the Hausdorff distance between A and B is defined for any nonempty sets $A, B \subseteq X$ as

$$d(A,B) = \text{Max}(d_h(A,B), d_h(B,A)) \quad (3)$$

Where

$$d_h(A,B) = \sup_{a \in A} d_y(a,B) \quad (4)$$

Let X be the set of $N \times N$ gray level images. The metric is defined as the usual Euclidean distance by regarding the elements in X as vectors of dimension $N \times N$. Let I be a given image belonging to X . The goal is to find a set of transformations $\{w_1, w_2, \dots, w_n\}$, each of which is a restricted function and satisfies (1), such that the given image I is an approximate attractor. The set $\{w_1, w_2, \dots, w_n\}$ is called PIFS. The following theorem is an important fact for PIFS.

Theorem 2.1. Consider a PIFS w_1, w_2, \dots, w_n with $w_i: X \rightarrow X$ for all i . Let $W = \cup w_i$. Then there exists a unique point $A \in X$ such that for any point $B \in X$

$$A = W(A) = \lim_{n \rightarrow \infty} w^n(B) \quad (5)$$

The point A in (5) is called the fixed point or the attractor of the mapping W . Next, the famous Collage theorem will be introduced.

Theorem 2.2. Let $\{w_1, w_2, \dots, w_n\}$ be a PIFS with contractivity factor s . Let B be any nonempty compact set in X . Then we have

$$d(A,B) \leq (1-s)^{-1} d(B,W(B)) \quad (6)$$

where A is the attractor of the mapping W and d is the Hausdorff metric (3).

Let $d(I,W(I)) \leq \epsilon$ where ϵ is a very small positive real number. By the Collage theorem, one can obtain that

$$d(A,I) \leq \frac{\epsilon}{1-\epsilon} \quad (7)$$

From Eq. (7), one can see that after a large number of iterations, an attractor A is generated which is sufficiently close to the given image I .

For practical implementation, let I be a given 256×256 gray level image. The domain pool D is defined as the set of all possible blocks of size 16×16 of the image f , which makes up $(256-16+1) \times (256-16+1) = 58081$ blocks. The range pool R is defined to be the set of all nonoverlapping blocks of size 8×8 , which makes up $(256/8) \times (256/8) = 1024$ blocks.

For each block v from the range pool, the fractal transformation is constructed by searching all elements in the domain pool D the most similar block. Let u denote a sub-sampled domain block which is of the same size as v . The similarity of u and v is measured using Mean Square Error (MSE) defined by

$$MSE = \frac{1}{8^2} \sum_{j=0}^7 \sum_{i=0}^7 [u(i,j) - v(i,j)]^2 \quad (8)$$

The fractal transformation allows the dihedral transformation of the domain blocks, i.e., the eight orientations of the blocks generated by rotating the blocks counter clockwise at angles 0, 90, 180, and 270 degrees and flipping with respect to the line $y = x$, respectively. Thus, for a

given block from the range pool, there are $58081 \times 8 = 464,648$ MSE computations to obtain the most similar block from the domain pool. Thus, in total, one needs $1024 \times 464,648 = 475,799,552$ MSE computations to encode the whole image using this full search compression method.

For a given range block v , the fractal transformation also allows the adjustment of the contrast p and the brightness q on the subsample domain block u . The similarity is measured by the quantity $d = \|p u_k + q - v\|$, where u_k , $0 \leq k \leq 7$, are the eight orientations of u . By simple optimization method, p and q can be computed directly as

$$p = \frac{[N\langle u, v \rangle - \langle u, 1 \rangle \langle v, 1 \rangle]}{[N\langle u, u \rangle - \langle u, 1 \rangle^2]} \quad (9)$$

And

$$q = N^{-1} \cdot [\langle v, 1 \rangle - p \langle u, 1 \rangle] \quad (10)$$

Where $N=64$.

The position of the most similar domain block, the contrast p , the brightness q , and the orientation k constitute the fractal code of the given range block v . In practice, for 256×256 image, 16 bits are required to represent the position of the domain block. Finally, as v runs over all 1024 range blocks in the range pool R , the encoding process is completed.

To decode, one first makes up the 1024 affine transformations from the compression codes and chooses any initial image. Next, one performs the 1024 affine transformations on the image to obtain a new image, and then proceeds recursively. According to Theorems 2.1 and 2.2, the sequence of images will converge. The stopping criterion of the recursion is designed according to user's application and the final image is the retrieved image of fractal coding.

3. QUANTUM EVOLUTIONARY ALGORITHMS

QEA is inspired from the principles of quantum computation, and its superposition of states is based on q-bits, the smallest unit of information stored in a two-state quantum computer [18,20]. A q-bit could be either in state "0" or "1", or in any superposition of the two as described below:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (11)$$

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

$$|\alpha|^2 + |\beta|^2 = 1 \quad (12)$$

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

3.1 Representation

QEA uses a novel representation based on the above concept of q-bits. Consider i th individual in t th generation defined as an m -q-bit as below:

$$q_i^t = \begin{bmatrix} \alpha_{i1}^t & \alpha_{i2}^t & \dots & \alpha_{im}^t \\ \beta_{i1}^t & \beta_{i2}^t & \dots & \beta_{im}^t \end{bmatrix} \quad (13)$$

Where $|\alpha_{ij}^t|^2 + |\beta_{ij}^t|^2 = 1, j=1,2,\dots,m$, m is the number of q-bits, i.e., the string length of the q-bit individual, $i=1,2,\dots,n$, n is the number of possible solution in population and t is generation number of the evolution. Since a q-bit is a probabilistic representation, any superposition of states is simultaneously represented. If there is, for instance, a three-q-bits($m = 3$) individual such as (14):

$$q_i^t = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} & \frac{1}{2} \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} & \frac{1}{2} \end{bmatrix} \quad (14)$$

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

$$q_i^t = \frac{1}{2\sqrt{6}}|000\rangle + \frac{1}{2\sqrt{2}}|001\rangle + \frac{1}{2\sqrt{3}}|010\rangle + \frac{1}{2}|011\rangle + \frac{1}{2\sqrt{6}}|100\rangle + \frac{1}{2\sqrt{2}}|101\rangle + \frac{1}{2\sqrt{3}}|110\rangle + \frac{1}{2}|111\rangle \quad (15)$$

Note that the square of above numbers are true probabilities, i.e. the above result means that the probabilities to represent the state $|000\rangle, |001\rangle, |100\rangle, |010\rangle$ are $1/24, 1/8, 1/24$ and $1/12$ respectively. Consequently, the three-q-bits system of (14) has all eight states information at the same time.

Evolutionary computing with the q-bit representation has a better characteristic of diversity than classical approaches since it can represent superposition of states. Only one q-bit individual such as (14) is enough to represent eight states, whereas in classical representation eight individuals are needed. Additionally, along with the convergence of the quantum individuals, the diversity will gradually fade away and the algorithm converges.

3.2 QEA Structure

In the initialization step of QEA, $[\alpha_{ij}^t \ \beta_{ij}^t]^T$ of all q0i are initialized with $1/\sqrt{2}$. This implies that each q-bit 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 $Q(t) = \{q_1^t, q_2^t, \dots, q_n^t\}$ states, where $X(t) = \{x_1^t, x_2^t, \dots, x_n^t\}$ at generation t is a random instant of q-bit population. Each binary instant, x_i^t of length m , is formed by selecting each bit using the probability of q-bit, either $|\alpha_{ij}^t|^2$ or $|\beta_{ij}^t|^2$ 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 q-bit 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.

3.3 Quantum Gates Assignment

The common mutation is a random disturbance of each individual, promoting exploration while also slowing convergence. Here, 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 q-bit individual q_i^t is updated by using the rotation gate $U(\theta)$ in this algorithm. The j th q-bit value of i th quantum

individual in generation t $[\alpha_{ij}^t \ \beta_{ij}^t]^T$ is updated as:

$$\begin{bmatrix} \alpha_{ij}^{t+1} \\ \beta_{ij}^{t+1} \end{bmatrix} = \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) \\ \sin(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix} \begin{bmatrix} \alpha_{ij}^t \\ \beta_{ij}^t \end{bmatrix} \quad (16)$$

Where $\Delta\theta$ is rotation angle and controls the speed of convergence and determined from Table I. Reference [18] shows that these values for $\Delta\theta$ have better performance.

TABLE I. Lookup table OF $\Delta\theta$.

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

4. TRIANGULAR FUNCTION FOR POPULATION SIZE IN QEA FOR FRACTAL IMAGE COMPRESSION (TFPSQEAFIC)

One of the main approaches to maintain the diversity of the population and improve the performance of the evolutionary algorithms is using a variable size for the population. In [19] a variable size population is proposed for QEA that improves the performance of QEA. Here, we use a triangular function for the size of the population with partially re-initialization of the q-individuals which case to improve the performance of QEA for fractal image compression. The proposed TFPSQEA method for fractal image compression is block-based, i.e., each q-individual is composed of the parameters of an affine transformation for one range block. The setup of our method is summarized as follows:

1. An important part of TFPSQEAFIC is q-individuals formation. Since the fractal encoding utilizes the PIFS to encode every range block, one takes the absolute position (P_x, P_v) of a domain block and the dihedral transformation P_t to constitute a q-individuals. A q-individuals is 19 bits in length as shown in Fig. 1, in which 8, 8, and 3 bits are allocated for (P_x, P_v) and P_t , respectively. P_x shows the horizontal position of domain block, P_v shows the vertical position of the domain block and P_t shows the transformation. The transformations are the 8 ordinary transformations: rotate 0° , 90° , 180° , 270° , flip vertically, horizontally, flip relative to 45° , and relative to 135° . Each part of each solution is a real number and is converted to an integer number before evaluation process. The length of q-individual for a $M \times N$ image is:

$$L = \lfloor \log_2^M \rfloor + \lfloor \log_2^N \rfloor + 3 \quad (17)$$

P_x	P_y	P_t
-------	-------	-------

Figure 1. The coding of the addresses for domain blocks. P_x is the horizontal position of domain block, P_y is the vertical position of domain block and P_t is the transformation.

2. For power of q-individuals (fitness function), the distance of both range block and sub-sampled domain block is measured by MSE. The fitness value is defined as the reciprocal of MSE.
 3. set t=0.
 4. initialize quantum population Q(0) with the size of $n(0) = \bar{n}$.
 In the initialization step, the quantum-individuals q_i^0 are located in a structured population. Then $[\alpha_i^0 \ \beta_i^0]$ of all q_i^0 are initialized with $1/\sqrt{2}$, where $i=1,2,\dots,n$ is the location of the q-individuals in the population, $k=1,2,\dots,m$, and m is the number of q-bits in the individuals. This implies that each q-bit individual q_i^0 represents the linear superposition of all possible states with equal probability..
 5. make X(0) by observing the states of Q(0).
 This step makes a set of binary instants $X(0) = \{x_i^0 | i = 1,2,\dots,n\}$ at generation t=0 by observing $Q(0) = \{q_i^0 | i = 1,2,\dots,n\}$ states, where X(t) at generation t is a random instant of q-bit population and n is the size of population. Each binary instant, x_i^0 of length m, is formed by selecting each bit using the probability of q-bit, either $|\alpha_{i,k}^0|^2$ or $|\beta_{i,k}^0|^2$ of q_i^0 . Observing the binary bit $x_{i,k}^t$ from q-bit $[\alpha_{i,k}^t \ \beta_{i,k}^t]^T$ performs as:

$$x_{i,k}^t = \begin{cases} 0 & \text{if } R(0,1) < |\alpha_{i,k}^t|^2 \\ 1 & \text{otherwise} \end{cases} \quad (18)$$
 Where $R(\cdot)$ is a uniform random number generator.
 6. Evaluate X(0).
 Each binary instant x_i^0 is evaluated to give some measure of its objective. In this step, the fitness of all binary solutions of X(0) are evaluated.
 7. for all binary solutions x_i^0 in X(t) do.
 Begin
 - 7.1. find neighborhood set N_i in X(0).
 - 7.2. find binary solution x with best fitness in N_i .
 - 7.3. save x in B_i .
 End.
- In these steps the neighborhood set N_i of all binary solutions x_i^0 in X(0) are found and the best solution among N_i is stored in B_i . In the structured proposed algorithm each individual is the neighbor of itself that is x_i belongs to neighborhood set N_i . B_i is the best possible solution, which the q-individual q_i^t has reached.

8. Repeat not termination condition.

The loop is terminated when the termination condition is satisfied. Termination condition here is when maximum number of iterations is reached.

9. Set $t = t + 1$.

10. $n(t) = f(t)$.

In the proposed algorithm, the size of the population is a function of the iteration number. This function is triangular function that shown in below:

$$n(t) = \text{Round} \left[\bar{n} - A + 2A \max \left(\min \left(2T^{-1} \cdot \text{mod}(t, T), 2 - 2T^{-1} \cdot \text{mod}(t, T) \right), 0 \right) \right] \quad (19)$$

Where $n(t)$ is the size of the population in generation t , \bar{n} is the average size of the population, A is the amplitude of the periodic function of population size, T is the period of the function of population, $\text{Round}(\cdot)$ is the round function (rounds its input to nearest integer) and $\text{mod}(\cdot, \cdot)$ is modulus after division function. Fig.2 shows this function. In this step, $n(t)$, the size of the population in iteration t , is calculated by this function.

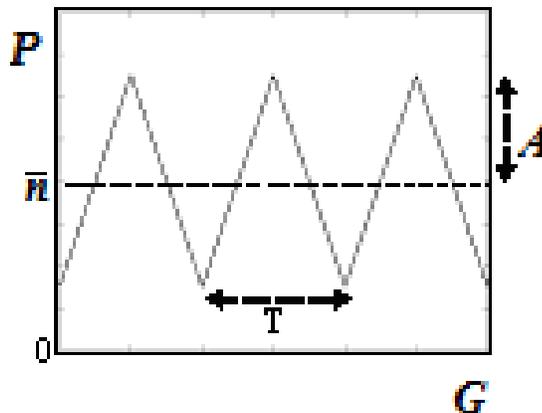


Figure 2. The triangular function which is used for the population size. T is the period of the functions, A is the amplitude and P is the size of the population in generation t .

11. if $n(t) > n(t-1)$ create random q -individuals.

If $n(t)$, the size of the population in iteration t , is greater than $n(t-1)$, it means that the size of the population is increased. So creating random q -individuals, until the size of ring structured population be equal to $n(t)$.

12. if $n(t) < n(t-1)$ eliminate the q -individuals with worst observed fitness.
If $n(t)$, the size of the population in iteration t is smaller than $n(t-1)$, eliminate the q -individuals which have the worst observed solution, until the size of ring structured population reaches $n(t)$.
13. make $X(t)$ by observing the states of $Q(t-1)$.
In this step, we are Observing the binary solutions $X(t)$ from $Q(t)$.
14. evaluate $X(t)$.
15. update $Q(t)$ based on B_i and $X(t)$ using Q -gates
Here, The quantum individuals are updated using Q -gate.
16. Repeat for all binary solutions x_i^t in $X(t)$ do
 Begin
 16.1. find neighbourhood, set N_i in $X(t)$.
 16.2. select binary solution x with best fitness in N_i .
 16.3. if x is fitter than B_i save x in B_i
 End.

The loop is for all binary solutions x_i^t ($i=1,2,\dots,S$) in the population and then Finding the neighbors of the binary solution located on the location i .

In next time we should find the best possible solution in the neighborhood N_i , and store it to x . If x is fitter than B_i then store x to B_i .

For each range block to fine optimal domain block, this algorithm should be down.

The triangular function for the population has two cycles. One cycle is increasing the size of population. In the increasing cycle, the new quantum individuals are created and inserted in the population. Creating new random quantum individuals increases the diversity of the population and improves the exploration performance of the algorithm. The other cycle is the decreasing cycle. In this cycle, the worst quantum individuals of the population are eliminated. This treatment improves the exploitation of the algorithm by exploiting the best solutions and ignoring the inferior ones. This means that the proposed method has two cycles: exploration cycle and exploitation cycle

5. EXPERIMENTAL RESULTS

This section experiments the proposed method and compares the proposed algorithm with the performance of GA 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 8×8 and the size of domain blocks is considered as 16×16 . In order to compare the quality of results, the PSNR test is performed:

$$PSNR = 10 \times \log \left(\frac{255^2}{\frac{1}{M \times N} \sum_{i=1}^N \sum_{j=1}^M (f(i,j) - g(i,j))^2} \right) \quad (19)$$

Where $M \times N$ is the size of image.

Parameters in triangular function set by $A=0.2$ and $T=100$ that are the amplitude of the periodic function of population size and the period of the function of population respectively. The crossover rate in GA is 0.8 and the probability of mutation is 0.003 for each allele. Table II shows the experimental results on the proposed algorithm and GA. The number of iterations for GA and the proposed method for all the experiments is 200. According to Table II the proposed algorithm improves the performance of fractal image compression for all the experimental results. As it is clear in Table II, TFPSQEA has better performance than GA in all the experiences. It shows that TFPSQEA is a suitable algorithm for fractal image compression. By using triangular function, the performance of QEA is improved. Although the PSNR for the proposed algorithm is less than the full search algorithm for all the experiences, but the advantage of the proposed algorithm is lower computational complexity. For example for Baboon picture, full search algorithm performs 59,474,944 MSE computations, while the proposed algorithm with the population size of 30 performs 6,144,000 MSE computations, i.e. ten times less. The proposed method reaches an admissible result much less computational complexity. Fig.3 shows the experimental results on Lena for the proposed algorithm and GA.

TABLE II. COMPARISON BETWEEN THE PROPOSED ALGORITHM AND GA.

Picture	Method	Population size	MSE computations	PSNR	
Lena	Full Search	-	59,474,944	28.85	
	SFQEA	15	3,072,000	27.30	
		20	4,096,000	27.59	
		25	5,120,000	28.10	
		30	6,144,000	28.15	
	GA	15	3,072,000	27.27	
		20	4,096,000	27.55	
		25	5,120,000	28.04	
		30	6,144,000	28.11	
	Pepper	Full Search	-	59,474,944	29.85
		SFQEA	15	3,072,000	28.13
			20	4,096,000	28.67
25			5,120,000	28.94	
30			6,144,000	29.17	
GA		15	3,072,000	28.11	
		20	4,096,000	28.64	
		25	5,120,000	28.92	
		30	6,144,000	29.14	
Baboon		Full Search	-	59,474,944	20.04
		SFQEA	15	3,072,000	18.42
			20	4,096,000	18.68
	25		5,120,000	19.04	
	30		6,144,000	19.19	
	GA	15	3,072,000	18.41	
		20	4,096,000	18.65	
		25	5,120,000	19.02	
		30	6,144,000	19.17	

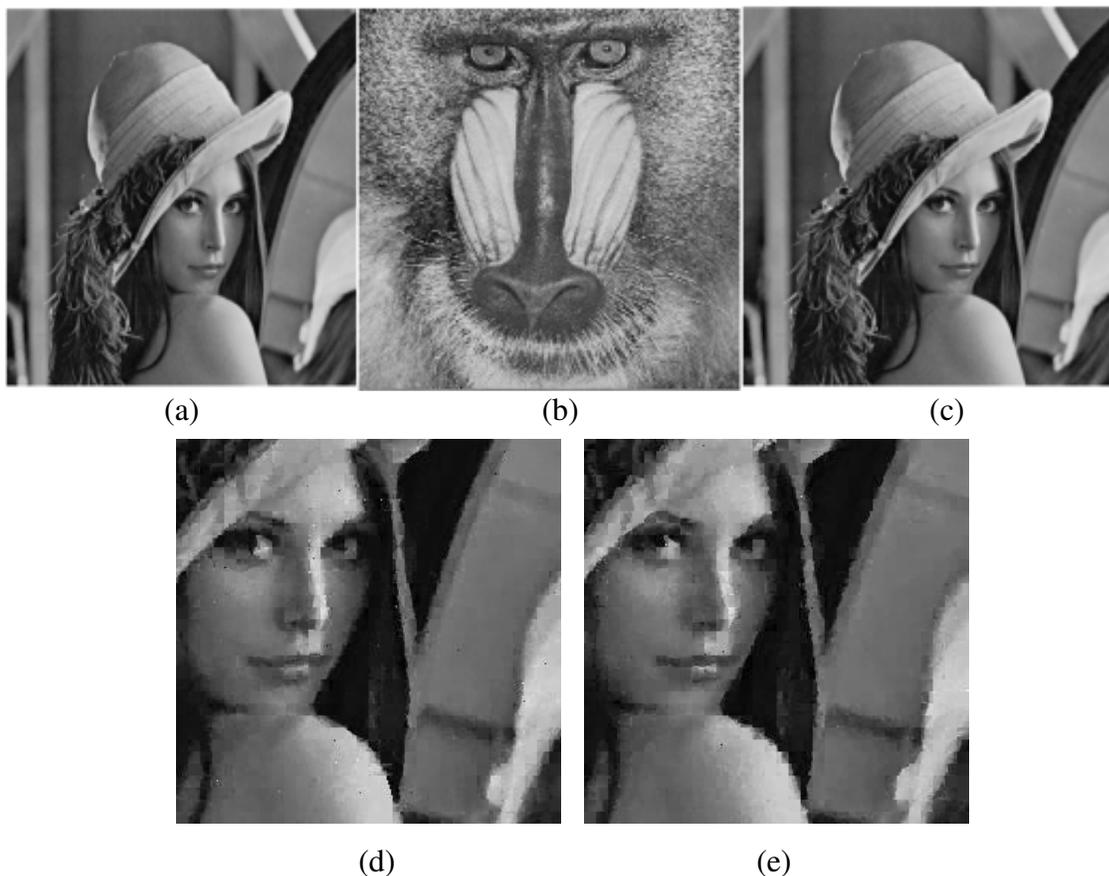


Figure 3. (a) Original image Lena of size 256×256 , (b) initial image for the decoder of the fractal compression, (c) full search method, (d) proposed TFPSQEA method, (e) GA method.

6. CONCLUSIONS

This paper uses a functional sized population in QEA for fractal image compression. Quantum Evolutionary Algorithms are novel algorithm proposed for combinatorial optimization problems like knapsack problem. Since fractal image compression is in the class of NP-Hard problems, QEA is highly suitable for this problem but is not properly applied to fractal image compression yet. One of the main approaches to maintain the diversity of the population and improve the performance of the evolutionary algorithms is using a variable size for the population. This paper uses a triangular function for the size of the population with partially re-initialization of the q -individuals that case to improve the performance of QEA for fractal image compression. Finally experimental results on Lena, Pepper and Baboon picture show an improvement on fractal image compression. Experimental result shows that the performance of our method is upper than conventional method from the encoding speed and it is better than from GA in retrieved image quality.

ACKNOWLEDGEMENTS

The authors would like to return thanks the financial support the Islamic Azad University, Kordkoy Branch, Iran.

REFERENCES

- [1] Peitgen H.O., Jurgens H. and Saupe D., "Chaos and Fractals: New Frontiers of Science", Springer Verlag, New York, 1992.
- [2] Barnsley, M.F., "Fractal Everywhere", Academic Press, California, 1988.
- [3] Barnsley, M.F. and Demko, S., "Iterated function systems and the global construction of fractals. Proceedings of the Royal Society, PP 243-275, 1985.
- [4] Barnsley M.F., Elton J.H. and Hardin D.P., "Recurrent iterated function systems Constr.", Approx. 3-31, 1989.
- [5] Jacquin A.E., "coding based on a fractal theory of iterated contractive image transformations", IEEE Trans. Image Process, PP 18-30, 1992.
- [6] Fisher Y., "Fractal Image Compression: Theory and Application", Springer Verlag, Berlin, 1995.
- [7] Wang Z., Zhang D. and Yu Y., "Hybrid image coding based on partial fractal mapping", Signal Process. Image Commun, pp 767-769, 2000.
- [8] Truong T.K., Kung C.M and Jeng J.H., Hsieh M.L., "Fast fractal image compression using spatial correlation, Chaos Solitons Fract", Chaos Solitons Fract, pp 1071-1076, 2004.
- [9] Univ X.J., "An improved genetic algorithm of solving IFS code of fractal image" IEEE 3rd international conference on signal processing 1996.
- [10] Chen X., Zhu G.X. and Zhu Y., "Fractal image coding method based on genetic algorithms" International Symposium on Multispectral Image Processing, 1998.
- [11] Mitra S.K., Murthy C.A. and Kundu M.K., "Technique for fractal image compression using genetic algorithm", IEEE Trans on Image Processing Vol 7 no 4 pp 586-593 1998.
- [12] Xun L. and Zhongqiu Y., "The application of GA in fractal image compression", 3rd IEEE World Congress on Intelligent Control and Automation, 2000.
- [13] Gafour A., Faraoun and Lehireche, "A Genetic fractal image compression". ACS/IEEE International Conference on Computer Systems and Applications, 2003.
- [14] Faraoun K.M. and Boukelif A., "Speeding Up Fractal Image Compression by Genetic Algorithms", Springer Journal of Multidimension Systems and Signal processing. Vol 16, No 2, 2005.
- [15] Lifeng X. and Liangbin Z., "A Study of Fractal Image Compression Based on an Improved Genetic Algorithm", International Journal of Nonlinear Science Vol 3, No 2, pp 116-124m 2007.
- [16] Wu M.S, Teng W.C., Jeng J.H and Hsieh J.G. , "Spatial correlation genetic algorithm for fractal image compression" Elsevier Journal of Chaos, Solitons and Fractals. Volume 28, Issue 2, pp 497-510, 2006.
- [17] Wu M.S., Jeng J.H. and Hsieh J.G., "Schema genetic algorithm for fractal image compression" Elsevier Journal of Engineering Applications of Artificial Intelligence Vol 20, Issue 4, pp 531-538 2007.
- [18] Han k. and Kim J., "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," IEEE Trans. on Evolutionary Computation, vol. 6, no. 6, 2002.
- [19] Koumousis V. K. and Katsaras C. P., "A Saw-Tooth Genetic Algorithm Combining the Effects of Variable Population Size and Reinitialization to Enhance Performance," IEEE Trans. Evol. Comput. vol. 10, pp. 19-28, 2006.
- [20] Kim H.J., "Quantum-Inspired Evolutionary Algorithms with a New Termination Criterion, He Gate, and Two-Phase Scheme," IEEE Trans. on Evolutionary Computation, vol. 8, no. 2, 2004.
- [21] . Tayarani- N and M.- R. Akbarzadeh-T "Improvement of Quantum Evolutionary Algorithm with Functional Sized Population". WSC 2008 Online World Conference on Soft Computing in Industrial Applications 10th - 21st of November 2008.
- [22] Dawei Li Wang "A study on the optimal population size of genetic algorithm" Proceedings of the 4th World Congress on Intelligent Control and Automation, 2002.
- [23] Shi, X.H. Wan, L.M. Lee, H.P. Yang, X.W. Wang, L.M. Liang, Y.C. "An improved genetic algorithm with variable population-size and a PSOGA based hybrid evolutionary algorithm" International Conference on Machine Learning and Cybernetics, 2003.

- [24] Qin Jun Kang Li-Shan “A novel dynamic population based evolutionary algorithm for revised multimodal function optimization problem” Fifth World Congress on Intelligent Control and Automation, 2004.