

A DISTRIBUTED QUANTUM EVOLUTIONARY ALGORITHM WITH A NEW CYCLING OPERATOR AND ITS APPLICATION IN FRACTAL IMAGE COMPRESSION

Ali Nodehi¹, Hosein Mohamadi² and Mohamad Tayarani³

¹Department of Computer Engineering, Gorgan Branch, Islamic Azad University
alinodehi@gorganiau.ac.ir

²Department of Computer Engineering, Azadshahr Branch, Islamic Azad University
h_mohamadi@hotmail.com

³Department of Computer Engineering, Southampton University, United Kingdom
tayarani@ieee.org

ABSTRACT

QUANTUM EVOLUTIONARY ALGORITHM (QEA) IS A NOVEL OPTIMIZATION ALGORITHM, PROPOSED FOR COMBINATORIAL PROBLEMS LIKE KNAPSACK AND TRAP PROBLEMS. WHILE FRACTAL IMAGE COMPRESSION IS IN THE CLASS OF NP-HARD PROBLEMS AND QEA IS HIGHLY SUITABLE FOR THE CLASS OF COMBINATORIAL PROBLEMS, QEA IS NOT WIDELY USED IN FRACTAL IMAGE COMPRESSION YET. IN ORDER TO IMPROVE THE PERFORMANCE OF FRACTAL IMAGE COMPRESSION ALGORITHMS, THIS PAPER PROPOSES A DISTRIBUTED QEA WITH A NOVEL OPERATOR CALLED CYCLING QUANTUM EVOLUTIONARY ALGORITHM. IN STANDARD QEA THE DIVERSITY IN THE POPULATION DECREASES ACROSS THE GENERATIONS. DECREASING THE DIVERSITY OF THE POPULATION DECREASES THE EXPLORATION PERFORMANCE OF THE ALGORITHM AND CAUSES THE ALGORITHM TRAPPING IN THE LOCAL OPTIMA. IN THE PROPOSED ALGORITHM, THERE ARE SOME SUBPOPULATIONS SEARCHING THE SEARCH SPACE. AFTER THE SUBPOPULATIONS ARE TRAPPED IN A LOCAL OPTIMUM, THE BEST OBSERVED POSSIBLE SOLUTIONS IN THE SUBPOPULATIONS ARE EXCHANGED IN A CYCLIC MANNER. THE PROPOSED ALGORITHM IS USED IN FRACTAL IMAGE COMPRESSION AND EXPERIMENTAL RESULTS ON SEVERAL IMAGES SHOW BETTER PERFORMANCE FOR THE PROPOSED ALGORITHM THAN GENETIC ALGORITHMS AND QEA. IN COMPARISON WITH CONVENTIONAL FRACTAL IMAGE COMPRESSION, THE PROPOSED ALGORITHM FINDS A SUITABLE SOLUTION WITH MUCH LESS COMPUTATIONAL COMPLEXITY

KEYWORDS

QUANTUM EVOLUTIONARY ALGORITHM, OPTIMIZATION ALGORITHM, FRACTAL IMAGE COMPRESSION, GENETIC ALGORITHM

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.

Fractal Image Compression has attracted much interest due to its high compression ratio and quality of compressed image. Fractal Image Compression problem is in the class of NP-Hard problems. Quantum Evolutionary Algorithm is a novel optimization algorithm proposed for class of combinatorial problems [9]. The probabilistic representation of possible solutions in QEA helps the algorithm escaping from local optima. While QEA is highly suitable for class of NP-Hard problems, it is not widely used in Fractal Image Compression. Several works have focused on improving the performance of QEA in solving knapsack and some other problems. In order to set the parameters of QEA for practical applications, [10] proposes some guidelines for parameter settings. In another work, by leading immune concepts in Artificial Immune Systems, [11] proposes a novel algorithm that is called the immune quantum-inspired evolutionary algorithm. In a more recent work, [12] proposes a novel multi-universe parallel Immune QEA that uses a learning mechanism. The proposed method maintains the population diversity along with better convergence speed. A multi-agent QEA is introduced in [13] and the multi-agent operators, competition and cooperation are performed to. The rotation gate is also considered as leverage to improve the performance of QEA. In [14] a new method of calculating rotation angle of quantum gate is proposed that causes a rapid convergence for QEA. In order to preserve the diversity in population and empower the search ability of QEA, [15] proposes a novel diversity preservation operator for QEA. Reference [16] proposes a sinusoid sized population QEA that makes a tradeoff between exploration and exploitation.

Several works try to improve the algorithm of fractal image compression using Genetic algorithm. The low speed of fractal image compression blocks its way to practical application. In [17] 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 proposed in [18]. In the proposed approach the search space for finding the best self similarity is greatly decreased. Reference [19] 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 [20] the speed of fractal image compression is improved without significant loss of image quality. In order to prevent the premature convergence of GA in fractal image compression a new approach is proposed in [21], which controls the parameters of GA adaptively. A schema genetic algorithm for fractal image compression is proposed in [22] to find the best self similarity in fractal image compression.

In evolutionary algorithms the diversity in the population is decreased as generations advance. When the algorithm reaches its mature state, the diversity in the population is decreased to its lowest quantity. In order to deal with this problem, this paper proposes a distributed QEA with a

cycling exchange of best observed possible solutions. The proposed algorithm is applied on fractal image compression problem and shows better performance for the proposed algorithm than conventional QEA and GA. This paper is organized as follows, in Section 2, the theoretical basics of fractal image compression is presented. In section 3, QEA and its representation is presented, Section 4 proposes the novel distributed QEA with the cycling operator, in Section 5 the proposed algorithm is applied on fractal image compression. Several experimental results are performed to in Section 6 to test the proposed algorithm, and finally Section 7 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 non-overlapping 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 counterclockwise 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

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):

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

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.

The pseudo-code of QEA algorithm is defined as [4]:

Procedure QEA

```

begin
    t=0
    1. initialize Q(0).
    2. make X(0) by observing the states of Q(0).
    3. evaluate X(0).
    4. Store X(0) into B(0). Store the best solution among X(0) into b.
    5. while not termination condition do
        begin
            t=t+1
        6. make X(t) by observing the states of Q(t-1)
        7. evaluate X(t)
        8. update Q(t) using Q-gates
        9. store the best solutions among B(t-1) and X(t) into B(t)
        10. store the best solution among B(t) into b
        11. if global migration condition
            then migrate b to B(t) globally
        12. else if local migration condition
            then migrate  $b_k^t$  in B(t) to B(t) locally
        end
    end
end
    
```

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

1. In the initialization step all q-bits α_{ij}^0 and β_{ij}^0 , $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, m$ are initialized with $2^{-1} \cdot \sqrt{2}$. It means the probability of observing "0" and "1" for all q-bits is equal.
2. In this step the binary solutions $X(0) = \{x_1^0, x_2^0, \dots, x_n^0\}$ at generation $t=0$ are creating by observing Q(0). Observing x_{ij}^t from qubit $[\alpha_{ij}^t \ \beta_{ij}^t]^T$ is performed as below:

$$x_{ij}^t = \begin{cases} 0 & \text{if } R(0,1) < |\alpha_{ij}^t|^2 \\ 1 & \text{otherwise} \end{cases} \tag{16}$$

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

3. All solutions in X(t) are evaluated with fitness function.
4. Store X(0) into B(0). Select best solution among X(0) and store it to b.

5. The while loop is running until termination condition is satisfied. Termination condition can be considered as maximum generation condition or convergence condition.
6. Observing X(t) from Q(t-1).
7. Evaluate X(t) by fitness function
8. Update Q(t)
- 9, 10. Store the best solutions among B(t-1) and X(t) to B(t). If the fittest solution among B(t) is fitter than b then store the best solution into b.
- 11, 12. If global migration condition is satisfied copy b to all the solutions in B(t). If local migration condition is satisfied replace some of solutions in B(t) with best one of them.

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:

$$\begin{bmatrix} \alpha_{ij}^{t+1} \\ \beta_{ij}^{t+1} \end{bmatrix} = \begin{bmatrix} \cos(\Delta\theta) & -\sin(\Delta\theta) \\ \sin(\Delta\theta) & \cos(\Delta\theta) \end{bmatrix} \begin{bmatrix} \alpha_{ij}^t \\ \beta_{ij}^t \end{bmatrix} \quad (17)$$

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. DISTRIBUTED QEA WITH CYCLING OPERATOR

In QEA when the algorithm converges, all the q-individuals reach similar values and their quantum bits converge to either 1 or 0. This means that there is little if any exploration during latter stages of the algorithm’s convergence and that all of the individuals search about a single optimum before converging to it. In other words, the algorithm cannot explore the other parts of

the search space well, and all of the individuals are exploiting a single optimum. This section proposes a novel Distributed QEA with a cycling operator to improve the diversity of the population and so the performance of the QEA. In the proposed algorithm the population is divided into some dependent subpopulations searching the search space. Since each subpopulation searches the search space independently, each of them reaches a different local optimum. Here convergence estimation is used to determine if the subpopulation is trapped in a local optimum. The converged subpopulations are considered as the trapped subpopulations and the cycling operator will be performed on them. In the cycling operator, the best observed of the subpopulations will be exchanged. When the best observed possible solutions are exchanged, the q-individuals of each subpopulation will move toward another local optimum. During this movement, the q-individuals have a chance to meet new local optima and it increases the chance of possible solutions in finding global optimum.

The algorithm of the proposed algorithm is briefly shown in figure 1. The propose algorithm is similar to except in steps 4 and 6. In this step the diversity preserving operator is performed on all the q-individuals.

In step 4 the best possible solution in each subpopulation is found and the best possible solution throughout the search is stored in B1. All the q-individuals in each subpopulation search around B1. The distributed nature of the proposed algorithm prevents the whole population from trapping in local optima. Although there is a chance for each of the subpopulation trapping in local optima, the statistical nature of QEA guarantees the subpopulations searching distinct local optima.

After trapping in local optima, the cycling operator is applied on the subpopulations. Here convergence estimation is used to determine if the subpopulation is trapped in a local optimum. The q-individual q_i^t converges if it satisfies the below constraint [23]:

$$\frac{1}{m} \sum_{k=1}^m |1 - 2|\alpha_{i,k}^t|^2| > \gamma \quad (18)$$

Where, m is the size of q-individuals. This constraint is satisfied only when the average distance between the q-bits of the q-individual q_i^t and the converged states ($[\alpha \ \beta]^T = [0 \ 1]^T$ or $[\alpha \ \beta]^T = [1 \ 0]^T$) is less than γ .

In step 6, when the subpopulations are converged, the best observed possible solutions are exchanged among the subpopulation in a cyclic manner:

- 1- $B_S = B_1$
- 2- For $l=1$ to $S-1$
- 3- $B_l = B_{l+1}$

Where B_1 is the best observed possible solution of the 1-th subpopulation and S is the number of subpopulations.

5. DISTRIBUTED QEA FOR FRACTAL IMAGE COMPRESSION

The proposed algorithm for each range block, searches among all the domain pool to find the best domain block and the best transformation. The coding method for the q-individuals in the proposed method is as figure 2.



Figure 2. 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.

In the proposed approach each q-individual, has three parts: P_x shows the horizontal position of domain block, P_y 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.

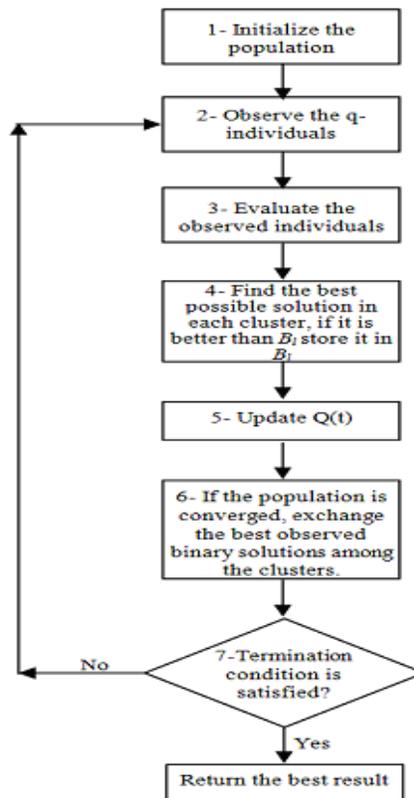


Figure 1. The algorithm of the propose algorithm.

6. EXPERIMENTAL RESULTS

This section experiments the proposed algorithm 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:

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

Where $m \times n$ is the size of image.

The crossover rate in GA is 0.8 and the probability of mutation is 0.003 for each allele. the parameters of QEA is set to Table I. The size of subpopulations is set to 5 for all the experiments. The convergence criteria, γ is considered as 0.98. Table II shows the experimental results on the proposed algorithm and GA. The number of iterations for GA, QEA and the proposed algorithm 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.

TABLE II. Experimental results on Lena, Pepper and Baboon

Picture	Method	Population 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	27.95
		15	3,072,000	27.43
	PROPOSED ALGORITHM	30	6,144,000	28.61
		25	5,120,000	28.29
		20	4,096,000	28.11
		15	3,072,000	27.49
	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 ALGORITHM	30	6,144,000	29.57
		25	5,120,000	29.19
		20	4,096,000	28.83
		15	3,072,000	28.54
	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 ALGORITHM	30	6,144,000	19.59
		25	5,120,000	19.33
		20	4,096,000	19.10
		15	3,072,000	18.61
	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

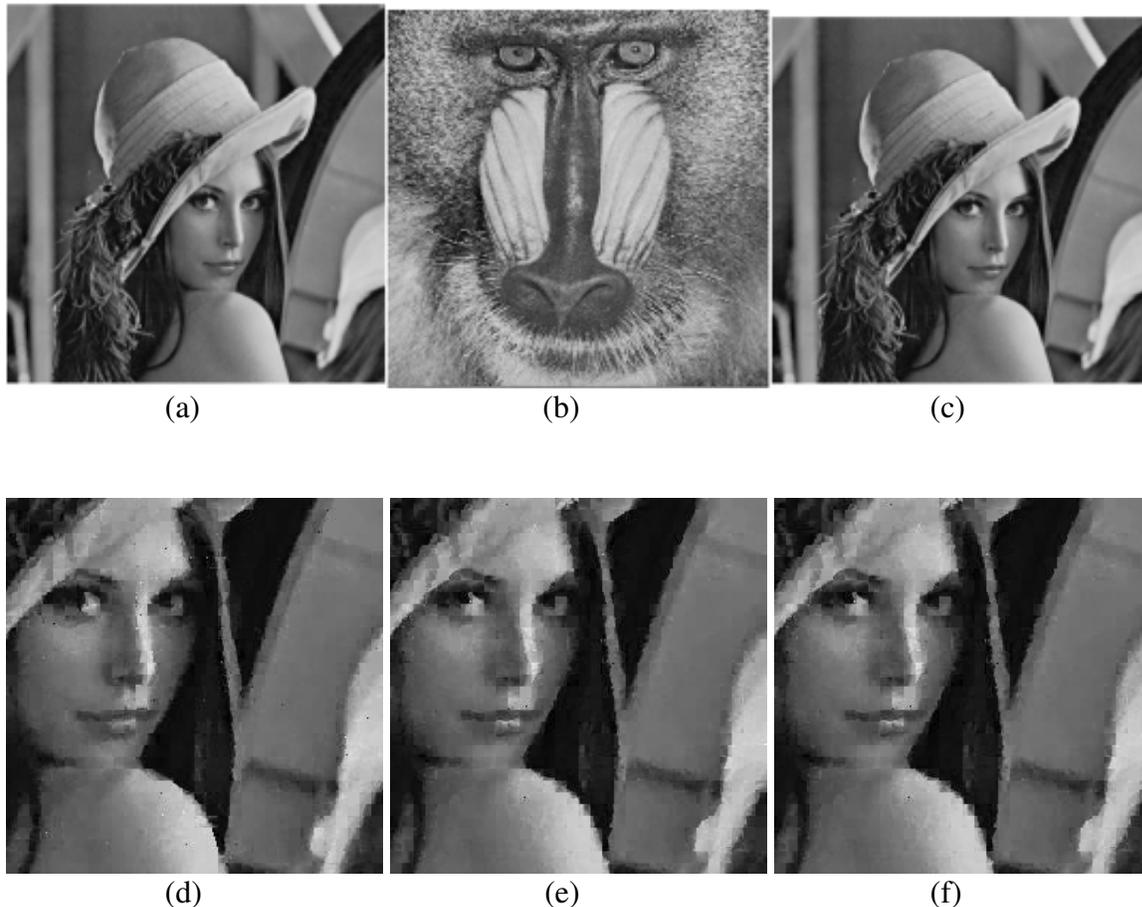


Figure 3. (a) Original image Lena of size 256×256 , (b) initial image for decoder, (c) full search method, (d) proposed method, (e) GA method, (f) QEA method

7. CONCLUSION

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. This paper proposes a Distributed QEA with a cycling Operator for QEA in fractal image compression. In evolutionary algorithms during the search process the diversity in the population is decreased, and the algorithm is trapped in the local optima. This is true for QEA which uses a probabilistic representation for the individuals. Here we propose a new method to help the algorithm escaping from the local optima. In the proposed algorithm the population is divided into some subpopulations and a cycling operator exchanges the best observed possible solutions among the subpopulations. The experimental results on Lena, Pepper, and Baboon picture show an improvement on fractal image compression.

ACKNOWLEDGEMENTS

The authors would like to return thanks the financial support the Islamic Azad University, Gorgan 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] K. Han, J. Kim, "Quantum-inspired evolutionary algorithm for a class of combinatorial optimization," IEEE Trans. on Evolutionary Comput, vol., no. 6, 2002.
- [10] Kuk-Hyun Han Jong-Hwan Kim "On setting the parameters of quantum-inspired evolutionary algorithm for practical application", IEEE Congress on Evolutionary Computation, 2003.
- [11] Ying Li Yanning Zhang Rongchun Zhao Licheng Jiao, "The immune quantum-inspired evolutionary algorithm ", *International Conference on Systems, Man and Cybernetics*, 2004 *IEEE*.
- [12] Xiaoming You, Sheng Liu and Dianxun Shuai "On Parallel Immune Quantum Evolutionary Algorithm Based on Learning Mechanism and Its Convergence", Springer, *Advanced in Natural Computation*. 2006.
- [13] Chaoyong Qin Jianguo Zheng and Jiyu Lai, "A Multiagent Quantum Evolutionary Algorithm for Global Numerical Optimization", Springer, *Life System Modeling and simulation*, Vol 4689, pp 380-389, 2007.
- [14] Rui Zhang Hui Gao, "Improved Quantum Evolutionary Algorithm for Combinatorial Optimization Problem", IEEE International Conference on Machine Learning and Cybernetics, 2007.
- [15] M.-H. Tayarani- N. and M.- R. Akbarzadeh-T "A Cellular Structure and Diversity Preserving operator in Quantum Evolutionary Algorithms". IEEE World Conference on Computational Intelligence 2008.
- [16] M.-H. Tayarani- N. and M.- R. Akbarzadeh-T "A Sinusoid Size Ring Structure Quantum Evolutionary Algorithm". IEEE International Conference on Cybernetics and Intelligent Systems Robotics, Automation and Mechanics 2008.
- [17] Xiao Chen, Guang-Xi Zhu, and Yaoting Zhu, "Fractal image coding method based on genetic algorithms" International Symposium on Multispectral Image Processing 1998.
- [18] Mitra, S.K. Murthy, C.A. Kundu, M.K "Technique for fractal image compression using genetic algorithm", IEEE Trans on Image Processing Vol 7 no 4 pp 586-593 1998.
- [19] Lu Xun Yu Zhongqiu "The application of GA in fractal image compression", 3rd IEEE World Congress on Intelligent Control and Automation, 2000.
- [20] Gafour, A. Faraoun, K. Lehireche, "A Genetic fractal image compression". ACS/IEEE International Conference on Computer Systems and Applications, 2003.
- [21] Lifeng Xi, Liangbin Zhang, "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.

- [22] Ming-Sheng Wu, Jyh-Horng Jeng, and Jer-Guang Hsieh, "Schema genetic algorithm for fractal image compression" Elsevier Journal of Engineering Applications of Artificial Intelligence Vol 20, Issue 4, pp 531-538 2007.
- [23] K. Han, J. Kim, "Quantum-Inspired Evolutionary Algorithms with a New Termination Criterion, H Gate, and Two-Phase Scheme," IEEE Trans. on Evolutionary Computation, vol. 8, no. 2, 2004.