A Novel Initialization for Quantum Evolutionary Algorithms Based on Spatial Correlation in Images for Fractal Image Compression

M. H. Tayarani N., A. Prugel Bennett, M. Beheshti and J. Sabet

Abstract Quantum Evolutionary Algorithm (QEA) is a novel optimization algorithm proposed for class of combinatorial optimization problems. While Fractal Image Compression problem is considered as a combinatorial problem, QEA is not widely used in this problem yet. Using the spatial correlation between the neighbouring blocks, this paper proposes a novel initialization method for QEA. In the proposed method the information gathered from the previous searches for the neighbour blocks is used in the initialization step of search process of range blocks. Then QEA starts searching the search space to find the best matching domain block. The proposed algorithm is tested on several images for several dimensions and the experimental results shows 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.

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

M. H. Tayarani, N.
University of Southampton, e-mail: mhtn1g09@ece.soton.ac.uk

Adam Prugel Bennett
University of Southampton, e-mail: apb@ecs.soton.ac.uk

is discussed. The low speed of fractal image compression blocks its way to practical application. 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 in 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 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 [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 certifiable, the whole of 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 spatial correlation between the neighbor range and domain blocks this paper proposes a novel initialization method for QEA. In the proposed method, based on the information gathered from the search process for the neighbor range blocks, the q-individuals are initialized to represent the better parts of the search space with higher probability. 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 shows better performance for the proposed algorithm than GA and original form 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 $\left|\alpha_{ij}^t\right|^2 + \left|\beta_{ij}^t\right|^2 = 1$, $j = 1,2,,m,m$ is the number of qubits, i.e., the string length of the qubit individual, $i = 1,2,,n,n$ is the number 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:

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

In QEA, only one qubit individual such as 4 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.

2.1 QEA Structure

In the initialization step of QEA, $[\alpha_{ij}^t \quad \beta_{ij}^t]^T$ 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.

2.2 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 qubit individual q_i^t is updated by using the rotation gate $U(\theta)$ 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:

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

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.

Table 1 Lookup Table of $\Delta\theta$, the rotation angle.

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

3 Proposed Method

Statistical studies on fractal image compression problems show for the range blocks, the potential quasi-affine matched domain centralized around its vicinity [1]. It means that the neighbor range blocks have similar fractal codes. Using this idea, [1] proposes a two stage fractal image coding method. The first stage searches around the best matched domain blocks of neighbor range blocks. If the result is not satisfying, the search process starts to find a good domain block. Inspiring this idea, this paper proposes a novel initialization method for Quantum Evolutionary Algorithm in solving fractal image compression problem. In solving fractal image compression problem using genetic algorithm, for each range block, GA searches among the domain pool to find the best matched domain block [10]. Basically the search process for each range block is performed independently and the results of the previous searches are not used in future searches for other range blocks. This paper proposes a novel method which uses the information from previous searches to help the evolutionary algorithm finding better solutions. In the proposed method, based on the information gathered from the previous searches for neighbor range blocks, the q-individuals are initialized to represent the better parts of the search space with higher probability. The new algorithm is proposed as follows:

The Proposed Algorithm
begin
    $t = 0$
1. Initialize $Q(0)$ based on $\mathcal{H}^b$, $\mathcal{H}^m$ and $\mathcal{H}^w$
1. 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
        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.

A comprehensive description of QEA can be found in [11]. The QEA procedure is described as:

1. In QEA the possible solutions are initialized with the values of $\frac{1}{\sqrt{2}}$ to represent the whole search space with the same probability. Giving these values to q-gates means a complete random initialization. Random initialization works well when we do not have previous information about the search space. Knowing where to search, helps QEA searching better parts of the search space and finding better solutions with less searching time. Here we propose a novel initialization approach using the information from previous searches. In the proposed method the best, worst and median solutions of each iteration in the search process for each range block are

stored in $\mathcal{H}^b$, $\mathcal{H}^m$ and $\mathcal{H}^w$ respectively. Where $\mathcal{H}$ is the history of X, containing the best, median and worst possible solutions in previous searches. The initialization step in the proposed algorithm is performed as follows:

$$\mathcal{B}_{l,i} = \frac{1}{T}\sum_{j=1}^{T}\mathcal{H}^b_{l,ij}, \quad \mathcal{M}_{l,i} = \frac{1}{T}\sum_{j=1}^{T}\mathcal{H}^m_{l,ij}, \quad \mathcal{W}_{l,i} = \frac{1}{T}\sum_{j=1}^{T}\mathcal{H}^w_{l,ij} \tag{7}$$

Where $\mathcal{H}^b$, $\mathcal{H}^m$ and $\mathcal{H}^w$ are the history of the best, median and worst possible solutions respectively, $\mathcal{H}^b_{l,ij}$ is the $i-th$ bit of $j-th$ possible solution in $\mathcal{H}^b$, and the index l shows the $l-th$ neighbor range block, $\mathcal{B}_{l,i}$ is the average of $i-th$ bit among all the possible solutions in $\mathcal{H}^b$ of $l-th$ neighbor range block. Here $l = 1,2,...,8$ shows the $l-th$ neighbor of the range block. Each range block has 8 neighbor range blocks. Here $\mathcal{B}_{l,i}$, $\mathcal{M}_{l,i}$ and $\mathcal{W}_{l,i}$ contain the percentage of ones in the best, median and worst possible solutions during the previous searches respectively. The value of $\mathcal{B}_i$ shows the importance of the $i-th$ bit in possible solutions, if $\mathcal{B}_i$ is near 1, it means that in most of better possible solutions, this bit has the value of 1, and it is better to give a value to this q-bit which represents 1 with more probability. Analogously, if $\mathcal{W}_i$ is near 1, it means that in most of worse possible solutions, this bit is 1, therefore it is better to give a value of to this q-bit which represents 0 with more probability. Accordingly, this paper proposes the following method for reinitialization step:

$$\theta^t_{ij} = \frac{\pi}{4} + [2\mathcal{B}_i + \mathcal{M}_i - 3\mathcal{W}_i] \times \frac{\pi}{16} \tag{8}$$

Where $j = 1,2,...,l$, l is the number of neighbor blocks, $i = 1,2,,m$, m is the number of q-bits in the q-individuals, i. e. the dimension of the problem. In order to prevent the q-individuals getting stuck in local optima of previous searches the other q-individuals are initialized randomly:

$$\theta^t_{ij} = \frac{\pi}{4} \tag{9}$$

For $j = l+1, l+2, ..., n$ where n is the size of the population.

The proposed initialization operator gathers information from the previous searches and initializes the q-individuals with the values representing better parts of search space.

2. In this step the binary solutions $X(0) = \{x^0_1, x^0_2, ..., x^0_n\}$ at generation $t = 0$ are created by observing $Q(0)$. Observing x^t_{ij} from qubit $[\alpha^t_{ij} \quad \beta^t_{ij}]^T$ is performed as below:

$$x^t_{ij} = \begin{cases} 0 & if \;\; U(0,1) < |\alpha^t_{ij}|^2 \\ 1 & otherwise \end{cases} \tag{10}$$

Where $U(.,.)$, 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.

3.1 Coding

In the proposed algorithm for each range block, QEA searches among all the domain pool to find the best match domain block. The domain blocks are coded by their horizontal and vertical address in the image. Therefore a solution is a binary string having 3 parts, p_x, p_y, p_T, representing the horizontal and vertical location of domain block in the image and the transformation respectively. The length of the possible solution for a $M \times N$ image is calculated as follows:

$$m = \lceil log_2(M) \rceil + \lceil log_2(N) \rceil + 3 \tag{11}$$

Where m is the size of the possible solutions. Here 8 ordinary transformation are considered: rotate $0°, 90°, 180°, 270°$, flip vertically, horizontally, flip relative to 45°, and relative to 135°.

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 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:

$$PSNR = 10 \times log \left(\frac{255^2}{\frac{1}{M \times N} \sum_{i=1}^{N} \sum_{j=1}^{M} (f(i,j) - g(i,j))^2} \right) \tag{12}$$

Where M×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 2 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. According to Table 2 the proposed algorithm improves the performance of fractal image compression for all the experimental results.

Table 2 Experimental results on Lena, Pepper, and Baboon

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	28.58
		25	5,120,000	28.39
		20	4,096,000	29.02
		15	3,072,000	27.56
	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	29.62
		25	5,120,000	29.28
		20	4,096,000	28.93
		15	3,072,000	28.51
	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	19.63
		25	5,120,000	19.25
		20	4,096,000	19.09
		15	3,072,000	18.77
	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

5 Conclusion

Using spatial correlation between the neighbor blocks in images, this paper proposes a novel initialization method for QEA in fractal image compression. In the proposed method, during the search process for each range block, some information about the best, worst and median possible solutions is gathered. Using this information in the initialization step in the search process of neighbor range blocks, this paper proposes a novel method in solving fractal image compression problem. Several experiments on Lena, Pepper, and Baboon pictures show an improvement on evolutionary algorithms solving fractal image compression.

References

1. W. Xing-yuan, L. Fan-ping, W. Shu-guo *Fractal image compression based on spatial correlation and hybrid genetic algorithm*, Elsevier, Journal of vis. commun. image R. pp. 505-510. 2009.
2. Yang Xuan, Liang Dequn, *An improved genetic algorithm of solving IFS code of fractal image*, IEEE 3rd international conference on signal processing 1996.
3. X. Chen, G. Zhu, and Y. Zhu, *Fractal image coding method based on genetic algorithms*, International Symposium on Multispectral Image Processing 1998.
4. S.K. Mitra, C.A. Murthy, M.K Kundu, *Technique for fractal image compression using genetic algorithm*, IEEE Trans on Image Processing Vol 7 no 4 pp 586-593 1998.
5. L. Xun and Y. Zhongqiu *The application of GA in fractal image compression*, 3rd IEEE World Congress on Intelligent Control and Automation, 2000.
6. Gafour, A. Faraoun, K. Lehireche, *A Genetic fractal image compression*, ACS/IEEE International Conference on Computer Systems and Applications, 2003.
7. F. K. Mohamed, B. Aoued, *Speeding Up Fractal Image Compression by Genetic Algorithms*, Springer Journal of Multidimention Systems and Signal processing. Vol 16, No 2, 2005.
8. L. Xi, L. 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.
9. M. Wu, W. Teng, J. Jeng and J. Hsieh, *Spatial correlation genetic algorithm for fractal image compression*, Elsevier Journal of Chaos, Solitons and Fractals. Volume 28, Issue 2, pp 497-510, 2006.
10. M. Wu, J. Jeng, and J. Hsieh,*Schema genetic algorithm for fractal image compression*, Elsevier Journal of Engineering Applications of Artificial Intelligence Vol 20, Issue 4, pp 531-538 2007.
11. K. Han and J. Kim, *Quantum-inspired evolutionary algorithm for a class of combinatorial optimization*, IEEE Transactions on. Evolutionary Computing, Vol. 6. No 6, 2002.