A Novel Functional Sized Population Quantum Evolutionary Algorithm for Fractal Image Compression

M.- Tayarani- N.

Azad University of Mashhad, Iran

tayarani@ieee.org

Abstract. Quantum Evolutionary Algorithm (QEA) is a novel optimization algorithm which uses a probabilistic representation for solution and is highly suitable for combinatorial problems like Knapsack problem. Fractal image compression is a well-known problem which is in the class of NP-Hard problems. Genetic algorithms are widely used for fractal image compression problems, but QEA is not used for this kind of problems yet. This paper uses a novel Functional Sized population Quantum Evolutionary Algorithm for fractal image compression. Experimental results show that the proposed algorithm 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

Recently we proposed a ring structure sinusoid sized ring structure population for QEA (SRQEA) [1]. In another work several functions for the size of population in QEA is proposed in [2] and tested for several benchmark functions. 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 [2]. Reference [3] finds the best population size for genetic algorithms. Inspired by the natural features of the variable size of the population [4] presents an improved genetic algorithm with variable population-size. In [5] an adaptive population size for the population is proposed for a novel evolutionary algorithm. Reference [6] proposes a scheme to adjust the population size to provide a balance between exploration and exploitation. To preserve the diversity in the population in QEA, [7] proposes a novel diversity preserving operator for QEA.

Several works try to improve the algorithm of fractal image compression using Genetic algorithm. In [10] 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 [11] 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 [12]. In the proposed approach the search space for finding the best self similarity is greatly decreased. Reference [13] 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 [14] the speed of fractal image compression is improved without significant loss of image quality. Reference [15] 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 quadtree 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 [16] which control the parameters of GA adaptively. A spatial correlation genetic algorithm is proposed in [17] which speed 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 [18] to find the best self similarity in fractal image compression.

While these approaches report a good improvement on fractal image compression, QEA which is highly suitable for combinatorial problems are not used for fractal image compression yet. Here we use a novel version of QEA which in fractal image coding has a better performance than GA. This paper is organized as follow: Section 2 introduces the proposed functional sized population for QEA. In Section 3 QEA is used for fractal image compression. Section 4 shows the experimental results and finally Section 5 concludes the paper.

3. Functional Sized population QEA (FSQEA)

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 [1] a variable size population is proposed for QEA that improves the performance of QEA. They use a sinusoid function for the size of the population with partially reinitialization of the q-individuals. Here to improve the performance of QEA for fractal image compression, a functional population size for QEA is proposed. In addition to the sinusoid function, this paper uses some other functions for QEA; the functions are saw-tooth [2], inverse saw-tooth, triangular, sinusoid [1] and square functions. Fig. 1 shows the functions which are examined in this paper. The pseudo code of the proposed Functional Size QEA (FSQEA) is described as below:

Procedure FSQEA

begin

    t=0

```
initialize quantum population Q(0) with the size of n(0) = n̄
make X(0) by observing the states of Q(0).
evaluate X(0).
for all binary solutions x^0_i in X(t) do
  begin
    find neighborhood set N_i in X(0).
    find binary solution x with best fitness in N_i
    save x in B_i
  end
while not termination condition do
  begin
   t=t+1
    n(t) = f(t)
   if n(t)>n(t-1) create random q-individuals
    if n(t)<n(t-1) eliminate the q-individuals with worst observed fitness
    make X(t) by observing the states of Q(t-1)
    evaluate X(t)
    update Q(t) based on B_i and X(t) using Q-gates
    for all binary solutions x^t_i in X(t) do
      begin
        find neighborhood set N_i in X(t).
        select binary solution x with best fitness in N_i
        if x is fitter than B_i save x in B_i
      end
  end
end
```

The pseudo code of FSQEA is described as below:

1. In the initialization step, the quantum-individuals q^0_i are located in a structured

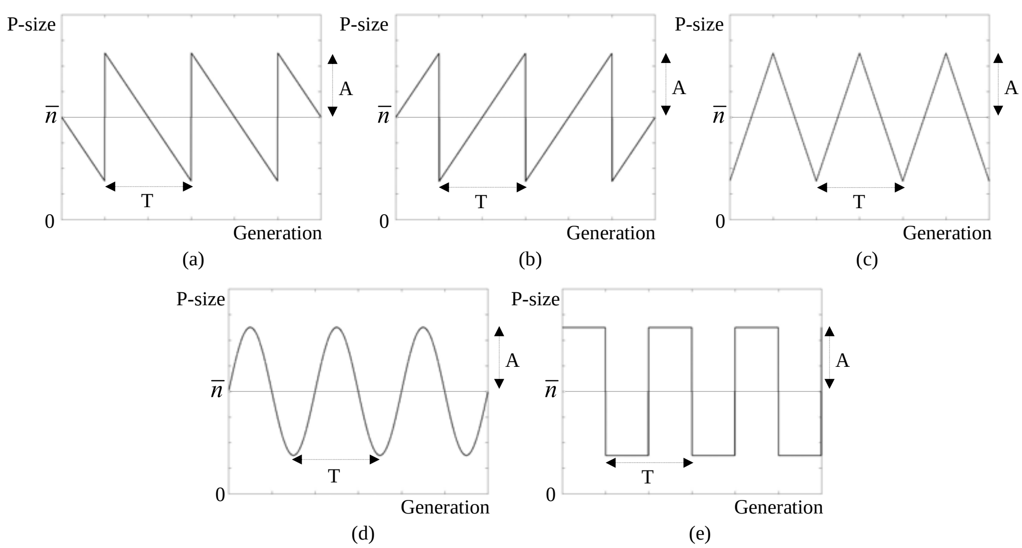

Fig. 1. a) The functions which are used for the population size. a) saw-tooth b) inverse saw-tooth c) triangular d) sinusoid e) square. T is the period of the functions, A is the amplitude and P-size is the size of the population

population. Then $[\alpha_i^0\ \beta_i^0]^T$ of all q^0_i are initialized with $1/\sqrt{2}$, where i=1,2,…,n is the location of the q-individuals in the population, k=1,2,...,m, and m is the number of qubits in the individuals. This implies that each qubit individual q^0_i represents the linear superposition of all possible states with equal probability.

2. This step makes a set of binary instants $X(0)=\{x_i^0|i=1,2,\ldots,n\}$ at generation t=0 by observing $Q(0)=\{q_i^0|i=1,2,\ldots,n\}$ 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^0_i of length m, is formed by selecting each bit using the probability of qubit, either $|\alpha_{i,k}^0|^2$ or $|\beta_{i,k}^0|^2$ of q^0_i. Observing the binary bit $x^t_{i,k}$ from qubit $[\alpha_{i,k}^t\ \beta_{i,k}^t]^T$ performs as:

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

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

3. Each binary instant x^0_i is evaluated to give some measure of its objective. In this step, the fitness of all binary solutions of $X(0)$ are evaluated.

4,5,6,7. In these steps the neighborhood set N_i of all binary solutions x^0_i 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^t_i has reached.

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

9. In the proposed algorithm, the size of the population is a function of the iteration number. In this step, $n(t)$, the size of the population in iteration t, is calculated as a function. The functions that used in this paper are:

Saw-tooth [3]: $n(t) = Round\left[\bar{n} - \frac{2A}{T-1}\left(t - T \times Round\,\frac{t-1}{T} - 1\right)\right]$

Inverse Saw tooth: $n(t) = Round\left[\bar{n} + \frac{2A}{T-1}\left(t - T \times Round\,\frac{t-1}{T} - 1\right)\right]$

Triangular: $n(t) = \mathrm{Round}\left[\bar{n} - A + 2A\max\left(\min\left(\frac{2\,\mathrm{mod}(t,T)}{T}, 2 - \frac{2\,\mathrm{mod}(t,T)}{T}\right), 0\right)\right]$

Sinusoid: $n(t) = Round\left[\bar{n} + A\sin\left(\frac{2\pi}{T}t\right)\right]$

Square: $n(t) = Round\left[\bar{n} + A - 2A \times Round\left(\frac{\mathrm{mod}(t,T)}{T}\right)\right]$

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 functional population, Round(.) is the round function (rounds its input to nearest integer), and mod(.,.) is modulus after division function. Fig 1 shows the functions which are used in this paper. The best values for T and A are found in the following of this section.

10. If $n(t)$, the size of the population in iteration t is greater than $n(t\text{-}1)$, 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 $n(t)$.
11. If $n(t)$, the size of the population in iteration t is smaller than $n(t\text{-}1)$, eliminate the q-individuals which have the worst observed solution, until the size of ring structured population reaches $n(t)$.
12. Observing the binary solutions $X(t)$ from $Q(t)$.
13. Evaluating the binary solutions $X(t)$.
14. The quantum individuals are updated using Q-gate.
15. The "**for**" loop is for all binary solutions x^t_i (i=1,2,…,S) in the population.
16. Finding the neighbors of the binary solution located on the location i.
17. Find the best possible solution in the neighborhood N_i, and store it to x.
18. If x is fitter than B_i, store x to B_i.

Table 1. the best parameters for the proposed FSQEA. The best results are the best ones.

	Saw-Tooth			Inverse-Saw			Sinusoid			Square			Triangular			QEA
	A	*T*	Best	*A*	*T*	Best	*A*	*T*	Best	*A*	*T*	Best	*A*	*T*	Best	Best
KR1	0.4	25	406.7	0.2	100	407.51	0.4	100	407.23	0.2	250	**407.67**	0.1	50	407.18	387.74
KR2	0.9	100	412.62	0.4	100	412.8	0.2	50	**413.05**	0.4	500	412.59	0.4	100	412.72	407.43
KP1	0.2	50	**556.69**	0.1	50	556.69	0.2	50	556.69	0.2	25	556.69	0.2	50	556.69	517.66
KP2	0.4	50	406.44	0.4	25	**407.56**	0.2	25	407.19	0.4	250	407.49	0.1	25	405.35	388.88
Trap	0.2	250	82.6	0.4	500	83.7	0.2	500	83.7	0.2	500	**84.4**	0.1	500	84.2	79.737
f_1	0.2	100	44932	0.6	100	47227	0.9	100	**47678**	0.4	100	45464	0.2	100	45973	32471
f_2	0.2	100	-1420	0.4	100	**-1274**	0.2	50	-1398	0.2	100	-1419	0.2	250	-1374	-2281
f_3	0.2	250	-17.07	0.4	100	**-16.88**	0.2	50	-16.98	0.2	25	-17.00	0.4	100	-17.00	-17.24
f_4	0.2	500	-22.85	0.2	250	**-17.44**	0.4	100	-21.22	0.2	250	-21.33	0.4	100	-20.74	-47.744
f_5	0.2	100	-1.0e5	0.4	500	-78259	0.4	500	-94385	0.4	250	**-3.8e4**	0.4	100	-9.0e4	-2.05e5
f_6	0.2	100	-22786	0.4	250	**-18903**	0.4	250	-21103	0.2	100	-2.2e4	0.4	250	-2.1e4	-49138
f_7	0.2	250	32.38	0.4	250	**35.40**	0.4	100	32.36	0.4	500	32.45	0.2	50	33.33	19.33
f_8	0.2	250	50.17	0.4	250	**53.64**	0.4	250	52.28	0.2	100	50.67	0.2	100	51.36	37.49
f_9	0.2	500	-2.5e5	0.4	250	**-1.94e5**	0.2	100	-2.27e5	0.2	100	-2.3e5	0.2	100	-2.3e5	-5.69e5
f_{10}	0.2	250	-3.55	0.4	500	**-2.99**	0.2	100	-3.33	0.2	250	-3.38	0.4	100	-3.3591	-5.5741
f_{11}	0.2	25	-162.19	0.2	500	-158.75	0.2	100	-161.56	0.2	250	-159	0.2	500	-163.13	**-143.63**
f_{12}	0.2	250	-7.1e6	0.4	500	**-6.22e6**	0.4	500	-7.8e6	0.2	100	-7.7e6	0.4	500	-7.1e6	-2.54e7
f_{13}	0.2	100	-39280	0.4	250	**-31939**	0.2	500	-37418	0.2	500	-36826	0.4	100	-34912	-1.10e5
f_{14}	0.6	25	-0.0057	0.6	25	-0.004	0.6	25	-0.009	0.9	50	**-0.001**	0.9	25	-0.0058	-1.13

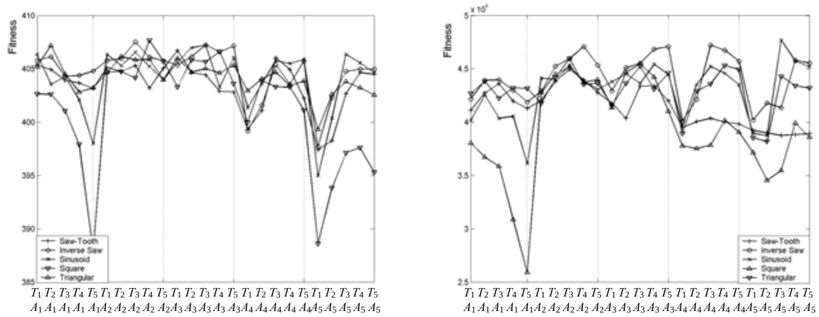

Fig. 2. parameter setting of FSQEA for T and A for (a) Knapsack Problem Penalty 1 (b) Generalized Schwefel Function 2.26 for several functions for the population. The parameters are set to $T_1 \dots T_5$=(25,100,250,500,1000) and $A_1 \dots A_5= \bar{n} \times$(0.1,0.2,0.4,0.6,0.9)

The proposed functions for the population have 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 algorithm has two cycles: exploration cycle and exploitation cycle.

3. Finding the best parameters

As it seen in Fig. 1, the proposed functions have some parameters that are A, the amplitude and T the period of the functions. In order to find the best values for these parameters some experiments are performed. Because fractal image compression is a time consuming algorithm and finding the best parameters for the algorithm needs to execute the algorithm for several parameters and for several times, it is not possible to set the parameters of the proposed algorithm for the fractal image compression problem. So the best parameters for the proposed algorithm are found for some benchmark functions like Knapsack Problem, Trap Problem and 14 numerical benchmark functions (see Appendix). Fig. 2 shows the finding of the best parameters for the proposed FSQEA for Knapsack Problem penalty type 1 and Generalized Schwefel Function 2.26. The best parameters for the Knapsack problem, Trap Problem and 14 numerical benchmark functions are found similar to the Fig 2. The best parameters and the best functions for the size of the population are summarized in Table 1. According to Table 1 the Inverse Saw-Tooth function has the best results for 11 benchmark objective functions, the Square for 3 benchmark functions, sinusoid for 2 functions, saw-tooth function for 1 benchmark function, and Triangular with no

p_x	p_y	p_t

Fig 3. The structure of the q-individuals. p_x shows the horizontal position of domain block, p_y shows the vertical position of domain block and p_t shows the transformation.

a) GA b) FSQEA

Fig 4. The comparison between the proposed FSQEA and GA for Lena. The size of the picture is 256×256, the size of range blocks is 8×8 and the size of domain block is 16×16.

objective function, so the best function for the size of the population is Inverse Saw-Tooth function. Only for one objective function the best result is reached by original version of QEA and the proposed algorithm improves the performance of QEA for most of the objective functions. In order to find the best parameters for the proposed algorithm Table 2 shows the median and standard deviation of the best parameters for 5 proposed functions. According to this table the best amplitude for the proposed functions in 0.2 and best period T is 100.

4. FSQEA for Fractal Image Compression

The proposed FSQEA for fractal image compression searches among the domain blocks to find the best domain block and the best transformation for each range block. For each range block, QEA searches among all the domain pool too find the best domain block and the best transformation. Fig 3 shows the coding method for the q-individuals in the proposed method. 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°. The size of p_x and p_y part of each q-individual depends on the size of the picture and the size of the p_t part is 3 bit. The gray coding is used for the coding of the individuals.

4. Experimental Results

Section 3 shows that the best function for FSQEA is the Inverse Saw-Tooth, the best parameters for the proposed algorithm are 0.2 for the amplitude and 100 for *T* period of the function. This section experiments the proposed algorithm and compares the proposed algorithm with the performance of GA in fractal image compression. We examine the proposed algorithm on images Lena, Pepper and Baboon with the size of 256×256 and gray scale. The size of range block is considered as 8×8 and the size of domain block is considered as 16×16. In order to compare the quality of results, the PSNR test is performed:

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

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

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 on the proposed algorithm and GA. The number of iterations for all the experiments is 200. According to table 3 the proposed algorithm improves the performance of fractal image compression for all the experimental results.

Table 2. Median and Standard deviation of the best parameters for the proposed FSQEA.

	A		*T*	
	Mean	Std	Mean	Std
Saw-Tooth	0.2	0.184	100	144
Inverse-Saw	0.4	0.124	250	177
Sinusoid	0.2	0.18	100	179
Square	0.2	0.17	250	168
Triangular	0.2	0.18	100	161

Table 3. comparison between the proposed algorithm and GA.

Picture	Method	Population size	MSE computations	PSNR
Lena	Full Search	-	59,474,944	28.85
	FSQEA	15	3,072,000	27.44
		20	4,096,000	27.93
		25	5,120,000	28.32
		30	61,440,000	28.51
	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
	FSQEA	15	3,072,000	29.53
		20	4,096,000	29.12
		25	5,120,000	28.81
		30	61,440,000	28.17
	GA	15	3,072,000	29.14
		20	4,096,000	28.92
		25	5,120,000	28.64
		30	6,144,000	28.11
Baboon	Full Search	-	59,474,944	20.04
	FSQEA	15	3,072,000	19.31
		20	4,096,000	19.14
		25	5,120,000	18.94
		30	61,440,000	18.52
	GA	15	3,072,000	19.17
		20	4,096,000	19.02
		25	5,120,000	18.65
		30	6,144,000	18.41

6. Conclusion

This paper proposes a Functional Sized population QEA for fractal image compression. The proposed functional sized population QEA has some parameters and this paper finds the best parameters for the proposed algorithm. Since the fractal image compression is a time consuming algorithm, and finding the best parameters needs several run of algorithm for several times, some benchmark functions are used to find the best parameters for the proposed FSQEA. Finally the experimental results on Lena picture show an improvement on fractal image compression. The time complexity of the proposed FSQEA is equal to original version of QEA because the average size of the population for FSQEA is equal to QEA and the number of function evaluations for both of algorithms is equal.

5. Appendix

In this section two combinatorial optimization problems, Trap problem and Knapsack problem, and 14 function optimization problems are discussed to evaluate the proposed SRQEA.

Trap problem is defined as:

$$f(x) = \sum_{i=0}^{N-1} \text{Trap}(x_{5i+1}, x_{5i+2}, x_{5i+3}, x_{5i+4}, x_{5i+5}) \quad (2)$$

Where N is the number of traps and

$$\text{Trap}(x) = \begin{cases} 4 - \text{ones}(x), & \text{if ones}(x) \le 4 \\ 5 & \text{if ones}(x) = 5 \end{cases} \quad (3)$$

Where the function "ones" returns the number of ones in the binary string x. Trap problem has a local optimum in $(0,0,0,0,0)$ and a global optimum in $(1,1,1,1,1)$.

Knapsack problem is a well-known combinatorial optimization problem which is in class of NP-hard problems [7]. Knapsack problem can be described as selecting various items x_i (i=1,2,…,m) with profits p_i and weights w_i for a knapsack with capacity C. Given a set of m items and a knapsack with capacity C, select a subset of the items to maximize the profit $f(x)$:

$$f(x) = \sum_{i=1}^{m} p_i x_i \quad , \quad \sum_{i=1}^{m} w_i x_i \le C \, .$$

This paper considered:

$w_i = \text{R}(1, v)$, $p_i = \text{R}(1, v)$

Where $\text{R}(\cdot,\cdot)$ is a uniform random number generator and v=10.

The use of QEA for solving Knapsack problem is described in [7].

The objective functions which are used here are f_1:Schwefel 2.26 [6], f_2:Rastrigin [6], f_3:Ackley [6], f_4:Griewank [6], f_5:Penalized 1 [6], f_6:Penalized 2 [6], f_7:Michalewicz [7], f_8:Goldberg [2], f_9:Sphere Model [6], f_{10}:Schwefel 2.22 [6], f_{11}:Schwefel 2.21 [6], f_{12}:Dejong [7], f_{13}:Rosenbrock [2], and f_{14}:Kennedy [2].

6. References

1. 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.
2. M.-H. 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.
2. V. K. Koumousis, C. P. Katsaras. "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.
3. 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.

4. 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.
5. 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.
6. W. Zhong, J. Liu, M. Xue, L. Jiao, "A Multi-agent Genetic Algorithm for Global Numerical Optimization," *IEEE Trans. Sys, Man and Cyber*. vol. 34, pp. 1128–1141, 2004.
7. A.-R. Khorsand, M.-R. Akbarzadeh-T. “Quantum Gate Optimization in a Meta-Level Genetic Quantum Algorithm”. *IEEE International Conference on Systems, Man and Cybernetics*, 2005.
8. M.-H. Tayarani- H, M.-R. Akbarzadeh-T. “A Cellular Structure and Diversity Preserving operator in Quantum Evolutionary Algorithms”. *IEEE World Conference on Computational Intelligence* 2008.
9. K. Han, J. Kim, “Quantum-inspired evolutionary algorithm for a class of combinatorial optimization,” IEEE Trans. on Evolutionary Computation, vol. 6, no. 6, 2002.
10. Xi'an Jiaotong Univ “An improved genetic algorithm of solving IFS code of fractal image” IEEE *3rd international conference on signal processing* 1996.
11. Xiao Chen, Guang-Xi Zhu, and Yaoting Zhu, “Fractal image coding method based on genetic algorithms” *International Symposium on Multispectral Image Processing* 1998.
12. 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.
13. Lu Xun Yu Zhongqiu “The application of GA in fractal image compression”, 3rd *IEEE World Congress on Intelligent Control and Automation*, 2000.
14. Gafour, A. Faraoun, K. Lehireche, “A Genetic fractal image compression”. *ACS/IEEE International Conference on Computer Systems and Applications*, 2003.
15. Faraoun Kamel Mohamed, Boukelif Aoued, “Speeding Up Fractal Image Compression by Genetic Algorithms” Springer Journal of Multidimention systems and sigenl processing. Vol 16, No 2, 2005.
16. 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.
17. Ming-Sheng Wu, Wei-Chih Teng, Jyh-Horng Jeng and Jer-Guang Hsieh, “Spatial correlation genetic algorithm for fractal image compression” Elsevier Journal of Chaos, Solitons and Fractals. Volume 28, Issue 2, pp 497-510, 2006.
18. 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.