Fractal Image Compression Based on Spatial Correlation And Hybrid Particle Swarm Optimization With Genetic Algorithm

[1]Gohar Vahdati, [2]Habib Khodadadi, [3]Mahdi Yaghoobi
[1,2,3]Computer Engineering Department
[1,2,3]Islamic Azad University, [1,3]Mashhad Branch, [2]Minab Branch
[1,3]Mashhad, [2]Minab, [1,2,3]Iran
[1]elham62.vahdati@gmail.com, [2] habibekhodadadi@gmail.com, [3]yaghobi@mshdiau.ac.ir

[4]Mohammad-R. Akbarzadeh-T.
Electrical Engineering Department
Ferdowsi University of Mashhad
Mashhad, Iran
[4]akbarzadeh@ieee.org

Abstract—Fractal image compression explores the self-similarity property of a natural image and utilizes the partitioned iterated function system (PIFS) to encode it. This technique is of great interest both in theory and application. However, it is time-consuming in the encoding process and such drawback renders it impractical for real time applications. The time is mainly spent on the search for the best-match block in a large domain pool. In this paper, a fractal image compression algorithm based on spatial correlation and hybrid particle swarm optimization with genetic algorithm (SC-PSOGA), is proposed to reduce the searching space. There are two stages for the algorithm: (1) Make use of spatial correlation in images for both range and domain pool to exploit local optima. (2) Adopt hybrid particle swarm optimization with genetic algorithm (PSO-GA), to explore the global optima if the local optima are not satisfied. Experiment results show that the algorithm convergent rapidly. At the premise of good quality of the reconstructed image, the algorithm saved the encoding time and obtained high compression ratio.

Keywords- Fractal image compression; particle swarm optimization; spatial correlation; genetic algorithm; PSO-GA.

I. INTRODUCTION

The idea of the fractal image compression (FIC) is based on the assumption that the image redundancies can be efficiently exploited by means of block self-affine transformations [1,2]. In 1988, Barnsley [3] proposed the idea of fractal image compression for the first time. The first practical fractal image compression scheme was introduced in 1992 by Jacquin [4]. One of the main disadvantages using exhaustive search strategy is the low encoding speed. Therefore, improving the encoding speed is an interesting research topic for FIC. So far, some improved approaches have been presented. Fisher and Wang et al. proposed their classification methods [5-6] based on the feature of the domain blocks, respectively. Truong et al.[7] 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. Some other researchers have combined fractal with other algorithms such as ant colony optimization [8], neural network [9], genetic algorithm[10-12], wavelet [13], etc. A schema genetic algorithm for fractal image compression is proposed in [14] to find the best self similarity in fractal image compression. Wu et al. [15] proposed a Spatial Correlation Genetic Algorithm (SC-GA), which speeded up the encoding time and increased compression ratio. PSO is an optimization algorithm having origins from evolutionary computation together with the social psychology principle [16-17].The mechanism of PSO algorithm generating the optimal or near-optimal solutions is a stochastic process. Though the searching process is stochastic and the grads information is ignored, the ability to pursue the optimal solutions of the particles and the collaboration among them ensure that the PSO algorithm can figure out the optimal or near-optimal solutions quickly. The formulae of the traditional PSO algorithm are simpler than those of the SA and the ACO algorithm, which means that the PSO algorithm can be more conveniently implemented. The population size of the PSO algorithm is smaller than those of the SA and GA algorithm, so the initialization of the populations is simpler using the PSO algorithm than that of the other intelligent optimization algorithms. In addition, the population size of the PSO algorithm is constant and the amount of the solutions is apt to be controlled in the optimization process. Although the traditional PSO algorithm has many advantages to resolve the optimization problems, the performance of it, such as the premature convergence and the stochastic stagnation, is heavily impacted by the principle and the parameters of the algorithm. The research [18] indicates that the traditional PSO algorithm will smoothly slip into the local near-optimal solutions when the optimization problem is relatively complex and it cannot jump over the obstruction. The drawback of PSO is that the swarm may prematurely converge. The underlying principle behind this problem is that, for the global best PSO, particles converge to a single point, which is on the line between the global best and the personal best positions. This point is not guaranteed for a local. Another reason for this problem is the fast rate of information flow between particles, resulting in the creation of similar particles with a loss in diversity that increases the possibility of being trapped in local optima. To overcome the limitations of PSO, hybrid algorithms with GA are proposed [19-21]. The basis behind this is that such a hybrid approach is expected to have merits of PSO with those of GA. The proposed hybrid PSO systems find a better solution without trapping in local maximum, and to achieve faster convergence rate. This is because when the PSO particles stagnate, GA diversifies the particle position even though the solution is worse. In PSO-GA, particle movement uses randomness in its search. Hence, it is a kind of stochastic optimization algorithm that can search a complicated and uncertain area. This makes PSO-GA more flexible and robust. Unlike traditional PSO, PSO-GA is more reliable in giving better quality solutions with reasonable computational time, since the hybrid strategy avoids premature convergence of the search process to local optima and provides better exploration of the search process [22].

In this paper, a fractal image compression algorithm based on spatial correlation and hybrid particle swarm optimization with genetic algorithm, is proposed. Results show that proposed algorithm reduced the coding time and retained high compression ratio under the premise of good image quality. The rest of the paper is organized as follows. We introduce the conventional fractal image coding scheme in Section II. Section III describes the PSO-GA algorithm. Section IV describes Fractal Image Compression based on spatial correlation and PSO-GA in this paper. Some experimental simulations are performed in Section V to verify the improvement of our proposed algorithm. Finally, a conclusion is made in Section VI.

II. FULL SEARCH FRACTAL IMAGE COMPRESSION

The fundamental idea of fractal image compression is the Iteration Function System (IFS) in which the governing theorems are the Contractive Mapping Fixed-Point Theorem and the Collage Theorem [5]. For a given gray level image of size $N \times N$, let the range pool R be the set of the $(N/L)^2$ non-overlapping blocks of size L×L which is the size of encoding unit. Let the contractivity of the fractal coding be a fixed quantity of 2. Thus, the domain pool makes up the set of $(N-2L+1)^2$ overlapping blocks of size (2L×2L). For the case of 256×256 image with 8×8 coding size, the range pool contains 1024 blocks of size 8×8 and the domain pool contains 58081 blocks of size 16×16. For each range block v in R, one searches in the domain pool D to find the best match, i.e., the most similar domain block. The parameters describing this fractal affine transformation form the fractal compression code of v. At each search entry, the domain block is first down-sampled to 8×8 and denoted by u. Let the set of down-sampled domain blocks be denoted by D. The down-sampled block is transformed subject to the eight transformations in the Dihedral on the pixel positions. If the origin of u is assumed to locate at the center of the block, the eight transformations $T_k : k = 0, \dots, 7$ can be represented by the following matrices:

$$T_0 = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix} \quad T_1 = \begin{bmatrix}1 & 0\\0 & -1\end{bmatrix} \quad T_2 = \begin{bmatrix}-1 & 0\\0 & 1\end{bmatrix}$$

$$T_3 = \begin{bmatrix}-1 & 0\\0 & -1\end{bmatrix} \quad T_4 = \begin{bmatrix}0 & 1\\1 & 0\end{bmatrix} \quad T_5 = \begin{bmatrix}0 & 1\\-1 & 0\end{bmatrix}$$

$$T_6 = \begin{bmatrix}0 & -1\\1 & 0\end{bmatrix} \quad T_7 = \begin{bmatrix}0 & -1\\-1 & 0\end{bmatrix} \tag{1}$$

The eight transformed blocks are denoted by u_k, $k = 0,1,\dots,7$, where $u_0 = u$. The transformations T_1 and T_2 correspond to the flips of u along the horizontal and vertical lines, respectively. T_3 is the flip along both the horizontal and vertical lines. T_4, T_5, T_6, and T_7 are the transformations of T_0, T_1, T_2, and T_3 performed by an additional flip along the main diagonal line, respectively. In fractal coding, it is also allowed a contrast scaling p and a brightness offset q on the transformed blocks. Thus, the fractal affine transformation Φ of u(x,y) in D can be expressed as:

$$\Phi\begin{bmatrix}x\\y\\u(x,y)\end{bmatrix} = \begin{bmatrix}a_{11} & a_{12} & 0\\a_{21} & a_{22} & 0\\0 & 0 & p\end{bmatrix}\begin{bmatrix}x\\y\\u(x,y)\end{bmatrix} + \begin{bmatrix}t_x\\t_y\\q\end{bmatrix} \tag{2}$$

Where the 2×2 sub-matrix $\begin{bmatrix}a_{11} & a_{12}\\a_{21} & a_{22}\end{bmatrix}$ is one of the Dihedral transformations in (1) and (t_x, t_y) is the coordinate of the domain block in the domain pool. In each search entry, there are eight separate MSE computations required to find the index d such that

$$d = argmin\{MSE((p_k u_k + q_k), v): \quad k = 0,1,\dots,7\} \tag{3}$$

Where

$$MSE(u,v) = \frac{1}{L^2}\sum_{i,j=0}^{L-1}(u(i,j=0) - v(i,j))^2 \tag{4}$$

Here, p_k and q_k can be computed directly as

$$p_k = \frac{\left[L^2\langle u_k, v\rangle - \sum_{i=0}^{L-1}\sum_{j=0}^{L-1} u_k(i,j)\sum_{i=0}^{L-1}\sum_{j=0}^{L-1} v(i,j)\right]}{\left[L^2\langle u_k, u_k\rangle - \left(\sum_{i=0}^{L-1}\sum_{j=0}^{L-1} u_k(i,j)\right)^2\right]} \tag{5}$$

$$q_k = \frac{1}{L^2}\left[\sum_{i=0}^{L-1}\sum_{j=0}^{L-1} v(i,j) - p_k \sum_{i=0}^{L-1}\sum_{j=0}^{L-1} u_k(i,j)\right] \tag{6}$$

As u runs over all of the 58081 domain blocks in D to find the best match, the terms $t_x$and t_y in (2) can be obtained. Together with d and the specific p and q corresponding this d, the affine transformation (2) is found for the given range block v. In practice, t_x, t_y, d, p, and q can be encoded using 8, 8, 3, 5, and 7 bits, respectively, which are regarded as the compression code of v. Finally, as v runs over all of the 1024 range blocks in R, the encoding process is completed.

III. HYBRID PARTICLE SWARM OPTIMIZATION WITH GENETIC ALGORITHM

A. Particle Swarm Optimization

PSO is a population-based algorithm for searching global optimum. The original idea of PSO is to simulate a simplified social behavior. It ties to artificial life, like bird flocking or fish schooling, and has some common features of evolutionary computation such as fitness evaluation.For example, PSO is like a GA in that the population is initialized with random solutions. The adjustment toward the best individual experience (PBEST) and the best social experience (GBEST) is conceptually similar to the crossover operation of the GA. However, it is unlike a GA in that each potential solution, called particle, is ‘‘flying” through hyperspace with a velocity. Moreover, the particles and the swarm have memory, which does not exist in the population of the GA [16-17]. PSO is initialized with a swarm including N random particles. Each particle is treated as a point in a D-dimensional space. The i-th particle is represented as $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$, x_{ij} is limited in the range $[a_j, b_j]$.The best previous position of the i-th particle (PBEST), is represented as $p_i = (p_{i1}, p_{i2}, \dots, p_{iD})$ .The best particle among all the particles in the population (GBEST), is represented by $p_g = (p_{g1}, p_{g2}, \dots, p_{gD})$. The velocity of particle i is represented as $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$.

After finding the aforementioned two best values, the particle updates its velocity and position according to the following equations:

$$v_{id} = v_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}) \quad (7)$$

$$x_{id} = x_{id} + v_{id} \quad (8)$$

where d is the d-th dimension of a particle, c_1 and c_2 are two positive constants called learning factors, r_1 and r_2 are random numbers in the range of [0, 1].The population size is first determined, and the position and velocity of each particle are initialized. Each particle moves according to (7) and (8), and the fitness is then calculated.

Meanwhile, the best positions of each particle and the swarm are recorded. Finally, as the stopping criterion is satisfied, the best position of the swarm is the final solution.

B. Genetic Algorithm

Genetic algorithm is a biologically motivated search method mimicking the natural selection and natural genetics. It is capable of finding the near-optimal solution since the candidate solutions will not get stuck at the local optima. Therefore, GA is especially efficient when the search space of a problem has very rough landscape riddled with many local optima. GA is suitable to fractal image compression because the search of the best match is highly related to such characteristics [15]. The elementary operations of GA include selection, crossover, and mutation.

C. Hybrid PSO with GA

In this model the initial population of GA is assigned by solutions of PSO. The total numbers of iterations are equally shared by PSO and GA. First half of the iterations are run by PSO and the solutions are given as initial population of GA. Remaining iterations are run by GA.

IV. Fractal Image Compression based on Spatial Correlation and Hybrid PSO with GA (SC-PSOGA)

The proposed method is implemented in two stages. The first stage makes full use of spatial correlations in images to exploit local optima. It can reduce the searching space of the similar matching domain pool, and shorten the optimal searching time. The second stage is operated on the whole image to explore more adequate similarities if the local optima are not satisfied. Let r_i be the range block to be encoded, $0 \le i < 1024$. Denote the neighbor range blocks of r_i, as depicted in Figure1, by r^H, r^V, r^D and $r^{D'}$ which have been encoded. Assume $d^{H(1)}$, $d^{V(1)}$, $d^{D(1)}$ and $d^{D'(1)}$ are the corresponding matched domain blocks, respectively. Now, one will restrict the searching space of r_i to $d^{H(1)}$, $d^{V(1)}$, $d^{D(1)}$, $d^{D'(1)}$ including some domain blocks in the relative directions. For example, $d^{H(1)}$ is the mapped domain block of r^H which is in the horizontal direction of r_i. Thus one expands the searching space in the horizontal direction (S^H) to $d^{H(0)}$, $d^{H(1)}$, $d^{H(2)}$ and $d^{H(3)}$ as depicted in Figure 1. Similarly, $d^{V(1)}$, $d^{D(1)}$and $d^{D'(1)}$ are expanded to S^V, S^D, $S^{D'}$, according to their corresponding directions.

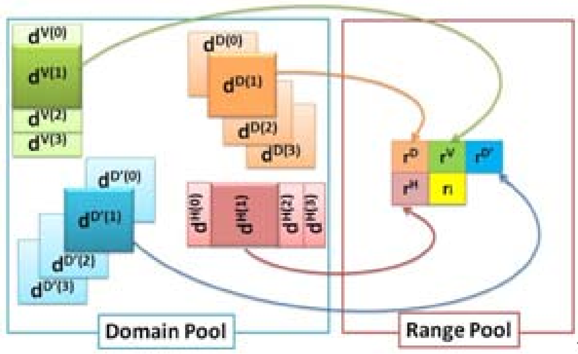

Figure 1. The searching space of the current range block in the first stage.

Thus, the searching space of r_i is limited to

$$S = \{S^H, S^V, S^D, S^{D'}\} \quad (9)$$

In this case, the expansion width is said to be 4, which can be set to other values according to the trade-off between the encoding speed and the bit rate [7]. It should be noted that some of these neighbors and their extended domain blocks might not exist. They are considered whenever they are applicable. To avoid large gaps between this local minimum and the global minimum obtained through the baseline method, one pre-defines a threshold T. If the local minimum exceeds this threshold , PSO-GA algorithm, will be invoked. The detailed steps of the proposed improved algorithm (SC-PSOGA) are given as follows:

(1) Let i = 0.

(2) Define the searching space S as depicted in Figure 1.

(3) Perform the first stage of the proposed algorithm on r_i. Let $d_s^{(i)}$ be the best-matched block in this stage. If $MSE(d_s^{(i)}, r_i) < T$, record the fractal code and go to step (5).

(4) Perform the second stage of the proposed algorithm (PSO-GA), on r_i. Let $d_f^{(i)}$ be the best-matched block in this stage. record the fractal code.

(5) Let i = i + 1, If i is equal to 1024, then stop, otherwise go to step (2).

if the fractal codes come from the first stage, the range block r_i is called ‘‘hit” block, which indicates that the local optima can satisfy the demand. For a hit block, fewer bits are required to record the offset of the domain block instead of the 16-bit absolute position. This will improve the compression ratio. Let N_R and N_H denote the number of range blocks and hit blocks, respectively. For hit blocks, 2 + B_w bits are required to record the relative positions, where B_w denotes the number of bits to represent the expansions. Let B_k, B_p and B_q denote the number of bits required to represent the orientation, the contrast and the brightness, respectively. Then the bit rate (bit per pixel) can be computed directly in terms of the number of hit blocks as

$$\mathrm{bpp} = \frac{N_H(1 + (2 + B_w) + B_k + B_p + B_q)}{N_{TP}} + \frac{(N_R - N_H)(1 + B_A + B_k + B_p + B_q)}{N_{TP}} \quad (10)$$

where N_{TP} is the total number of pixels in the image. Note that one bit is required to indicate if the block is a hit block. The detailed design of PSO-GA is summarized as follows: As discussed in Section II, the parameters t_x, t_y, d, p, and q constitute the fractal code. In PSO method, we encode the particle as (t_x, t_y), which is the position of the domain block. The quantities p and q can be calculated from (5) and (6), and k is searched separately. At each search entry, all of the eight Dihedral transformations in (1) are performed and the best index d in (3) can be obtained. The fitness value of a particle is defined as the minus of the minimal MSE produced from eight Dihedral transformations, i.e., $-\mathrm{MSE}((p_d u_d + q_d),v)$. The steps of PSO algorithm are summarized as follows:

1. Initialize the parameters of PSO.
2. For each particle (t_x, t_y), fetch the domain block at (t_x,t_y) in the image. Sub-sample the block and denote it by u.
3. For each Dihedral transformation, compute u_k, k = 0,…, 7. Calculate the contrast scaling p_k and brightness offset q_k. Find the fitness of the particle corresponding to the best parameter d as given in (3).
4. Update the PBEST(p_i) and the GBEST(p_g) if required.
5. If stopping criterion is satisfied, then stop.
6. For each particle, update the velocity and the position according (7) and (8). Go to step 2.

(1) Chromosome formation:

Since the fractal encoding scheme utilize the PIFS to encode every range block, one takes the absolute position (t_x, t_y) of a domain block and the dihedral transformation d to constitute a chromosome. A chromosome is 19 bits in length as shown in Figure 2, in which 3, 8 and 8 bits are allocated for d and (t_x, t_y), respectively.

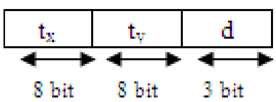

Figure 2. Structure of a chromosome

(2) 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) Initial population:

Chromosomes are initialized by solutions of PSO.

(4) Selection:

Select two individuals into the mating pool to take part in the genetic operation in term of roulette wheel method.

(5) Crossover:

Perform one-point crossover on the selected chromosomes in the mating pool with probability p_c.

(6) Mutation:

The mutation operation with probability Pm is applied to the temporary offspring in order to generate the offspring for the next generation. Here the goal of the mutation is to maintain the diversity of the population and to avoid pre-maturity.

(7) Stopping criterion:

The goal of stopping criterion is to stop the GA evolutionary process. Here, a pre-set number of iterations is adopted to stop the evolution.

V. EXPERIMENTAL RESULTS

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 our experiments, the MSE threshold (T) values are set to be 100, 300 and 80, for the images Lena, Baboon, and Pepper, respectively. The coefficients of PSO are set heuristically as C_1=1.3 and C_2=1.4.

Table I, shows the various parameters of different algorithms used in this current investigation. Table II, shows the experimental results on the hybrid PSO-GA method, Full Search method and Traditional GA method.

Table III, shows some comparative results for both our method and SC-Full Search method[7]. According to Tables II and III, the proposed algorithm improves the performance of fractal image compression for all the experimental results.

TABLE I. DIFFERENT PARAMETERS OF HYBRID PSO-GA METHOD AND PROPOSED METHOD

method		Population size	iteration	Pc	Pm
PSO-GA	***PSO***	20	20	-	-
	GA	20	20	0.6	0.005
SC-PSOGA	***PSO***	30	30	-	-
	GA	30	30	0.6	0.005

TABLE II. THE COMPARISON OF HYBRID PSO-GA ALGORITHM AND TRADITION GA METHOD TOGETHER WITH FULL SEARCH METHOD

Image	Method	PSNR	Speed-up rate
Lena	***Full Search***	28.91	1
	PSO-GA	27.53	69.66
	Traditional GA	26.23	54.35
Pepper	***Full Search***	29.84	1
	PSO-GA	28.29	68.36
	Traditional GA	27.10	53.99
Baboon	***Full Search***	20.15	1
	PSO-GA	19.75	76.05
	Traditional GA	19.47	53.62

TABLE III. THE COMPARISON OF PROPOSED METHOD (SC - PSOGA) WITH SC- FULL SEARCH METHOD

Image	Method	PSNR	Time	Speed up rate	Hit block	bpp
Lena	*Full Search*	28.91	3135	1	-	0.4844
	SC – full search	27.94	1326	2.36	581	0.3936
	SC - PSOGA	27.24	44	71.25	584	0.3931
Pepper	*Full Search*	29.84	3145	1	-	0.4844
	SC – full search	29.12	1220	2.57	584	0.3931
	SC – PSOGA	28.23	40	78.62	589	0.3922
Baboon	*Full Search*	20.15	2966	1	-	0.4844
	SC - Full Search	19.90	1502	1.97	304	0.4443
	SC - PSOGA	19.66	73	40.63	303	0.4445

VI. CONCLUSION

In this paper, a fractal image compression method based on spatial correlation and hybrid particle swarm optimization with genetic algorithm, is proposed. There are two stages for the algorithm. The first stage exploits local optima by making use of the spatial correlation between neighboring blocks. If the local optima are not satisfied, the second stage of the algorithm is carried out in order to explore further similarities from the whole image. Since the searching space in the first stage is much smaller, so the coding time is reduced. Such a method can speed up the encoder and also preserve the image quality. Moreover, the compression ratio can also be improved since only relative positions are recorded in the first stage of the algorithm.

REFERENCES

[1] J. Crilly, R.A. Earnshaw, H. Jones, “Fractals and Chaos,” Springer-Verlag, New York, 1991.

[2] H.O. Peitgen, J.M. Henriques, L.F. Penedo, “Fractals in the Fundamental and Applied Sciences,” Elsevier Science Publishing Company Inc., New York, 1991.

[3] M.F. Barnsley, “Fractal Everywhere,” Academic, New York, 1988.

[4] A.E. Jacquin, “Image coding based on a fractal theory of iterated contractive image transformations,” IEEE Transactions on Signal Processing, vol. 1, pp. 18-30, 1992.

[5] Y. Fisher, “Fractal Image Compression,” Theory and Application, Springer Verlag, Berlin, 1995.

[6] Z. Wang, D. Zhang, Y. Yu, “Hybrid image coding based on partial fractal mapping,” Signal Process. Image Commun, vol. 15, no. 9, pp. 767-769, 2000.

[7] T.K. Truong, C.M. Kung, J.H. Jeng, M.L. Hsieh, “Fast fractal image compression using spatial correlation,” Chaos, Solitons and Fractals, vol. 22, pp. 1071-1076, 2004.

[8] J. Li, D. Yuan, Q. Xie, C. Zhang, “Fractal Image Compression by Ant Colony Algorithm, ” In : Proceedings of the 9th International Conference for Young Computer Scientists, IEEE , pp. 1890-94, 2008.

[9] Y. Chakrapani, K. Soundera Rajan, “Implementation of fractal image compression employing artificial neural networks ,” World Journal of Modelling and Simulation ISSN 1 746-7233, vol. 4, no. 4, pp. 287-295, 2008.

[10] L. Vences, I. Rudomin, “Genetic algorithms for fractal image and image sequence compression ,‘’ Proc. Comput. Vis, pp. 35–44, 1997.

[11] S.K. Mitra, C.A. Murthy, M.K. Kundu, “Technique for fractal image compression using genetic algorithm,” IEEE Trans. Image Process, vol. 7, no. 4, pp. 586–593, 1998.

[12] F. MOHAMED, B. AOUED, “Speeding Up Fractal Image Compression by Genetic Algorithms,” Multidimensional Systems and Signal Processing, vol.16, pp. 217-236, 2005.

[13] Y. Iano, F. Silvestre , A. L. Mendes Cruz, “A Fast and Efficient Hybrid Fractal-Wavelet Image Coder,” IEEE TRANSACTIONS ON IMAGE PROCESSING, vol.15, no.1, pp. 98-105, 2006.

[14] M. S. Wu, J. H. Jeng, J. G. Hsieh, “Schema genetic algorithm for fractal image compression,” Engineering Applications of Artificial Intelligence, vol. 20, pp. 531-538, 2007.

[15] M. S. Wu , W. C. Teng, J. H. Jeng, J. G. Hsieh, “Spatial correlation genetic algorithm for fractal image compression”, Chaos, Solitons and Fractals, vol. 28, pp. 497-510, 2006.

[16] J. Kennedy, R.C. Eberhart, “Particle swarm optimization, “ in: Proceedings of IEEE International Conference on Neural Networks, Perth, Australia, vol. 4, pp. 1942–1948, 1995.

[17] R.C. Eberhart, J. Kennedy, “A new optimizer using particle swarm theory,” in: Proceedings of IEEE International Symposium on Micro Machine and Human Science, Nagoya, Japan, pp. 39–43, 1995.

[18] F. Bergh, A. P. Engelbrecht, “A study of particle swarm optimization particle trajectories,” Information Sciences, vol. 17, no. 8, pp. 937-71, 2006.

[19] Q. Niu, B. Jiao, X. Gu, “Particle swarm optimization combined with genetic operators for job shop scheduling problem with fuzzy processing time,” Applied Mathematics and Computation, vol. 205, pp. 148-158, 2008.

[20] R. Ding, H. Dong, X. Feng, G. Yin, “A Hybrid Particle Swarm Genetic Algorithm for Classification,” International Joint Conference on Computational Sciences and Optimization, pp. 301-305, 2009.

[21] Y. Gao, Z. Ren,” Adaptive Particle Swarm Optimization Algorithm With Genetic Mutation Operation,” Third International Conference on Natural Computation (ICNC2007), 2007.

[22] K. Premalatha , A.M. Natarajan, “Hybrid PSO and GA for Global Maximization,” Int. J. Open Problems Compt. Math., vol. 2, no. 4, pp. 597-608, 2009.