A PSO-based Algorithm with Subswarm Using Entropy and Uniformity for Image Segmentation

[1]Jzau-Sheng Lin and [1]Shou-Hung Wu

Abatract

In the image segmentation field, it need several iterations to find optimization thresholds or cluster center to segment images. In this paper, we embedded a scheme based on maximum entropy and uniformity into the particle swarm optimization with subswarm structure named MEUPSOS to find the optimization threshold values iteratively. Instead of using the conventional PSO, swarm was divided into several subswarms for the purpose of getting local optimal solutions with a fitness function based on maximal entropy and uniformity. Additionally, just one-swarm particles were used to replace *k*-swarm (*k* is the number of threshold values) particles in order to upgrade the computation performance. Then, the local optimal solutions ware used to update global parameter in the global swarm. Through iterations updating the velocities and locations of particles, we can calculate the near optimal threshold values on an image based on the fitness functions of maximum entropy and uniformity. Finally, we can find that the proposed method can get more promising results than the other method.

Keywords- PSO; image segmentation; entropy; uniformity

1. Introduction

Segmentation in an image is an important step for the image processing. Great deals of literatures, based on region growing [1], edge detection [2], pixel classification [3-4], and histogram threshold [5], have been proposed in the research field of image segmentation. Region growing strategies start at known pixels and append all neighbors which are similar in gray level, color, texture, or other properties, to the known pixels in order to form a region. The local discontinuities are detect first and then connected to form complete boundaries in edge detection approaches. A pixel classification approach is one that classifies pixels based on global of local information such as their gray levels or colors in an image. In the traditional pixel classification approaches, they can be divided into two schemes such as supervised and unsupervised classification approaches. The supervised strategy classifies pixels into representative regions with known properties like number of regions or cluster centroids. Then these regions are merged into suitable clusters. The supervised method is so troublesome that it is time consuming. The unsupervised approach is a classification process that does not need to be experience cognitive. It just modifies cluster centers through several iterations to achieve the convergence condition in order to complete the classification.

Generally, a thresholding algorithm is one that determines optimal threshold values based on a certain criterion. A single threshold value is used for the whole image in a global thresholding method as well as several threshold values are used to segment a given image into subregions in a local thresholding. There are several optimal algorithms in thresholding segmentation have been proposed such as maximum entropy threshold [6] and Minimum error threshold [7-8].In order to find optimal threshold values and reduce the time consumption, biological intelligence algorithms, such as Artificial ant algorithm[9], Particle Swarm Optimization [10], Artificial Bee Colony [11-12] , Genetic Algorithm[13], and Bacterial Evolutionary[14], have been embedded into thresholding segmentation method to find global or near global optimal solution.

Artificial ant algorithm is used to choose a correct pathf or travelling in accordance with the pheromone. The quantity of pheromone on a path can be increased when ants go through the same path. In order not to fall into a local optimal solution, pheromones on all paths have to be reduced but the pheromone weight on the best path has to be increased in accordance with a fitness function.

Artificial Bee Colony (ABC) algorithm was proposed by Karaboga [11-12] for optimizing numerical problems. The algorithm simulates the intelligent foraging behavior of honey bee swarms. It is a near robust stochastic optimization algorithm.

Genetic algorithm is a heuristic algorithm commonly used binary encoding. It exchanges bits of information to find a better solution through iterative crossover step from their good parent generation. Additionally, the mutation mechanism also used to prevent trapped in local optimal solutions.

PSO method, belonging to the category of swarm optimization, is a population-based algorithm and first demonstrated as a stochastic optimization algorithm

[1]*Dept. of Computer Science and Information Engineering Nat'l Chin-Yi University of Technology Taichung City, TAIWAN*
jslin@ncut.edu.tw and happy_like92@yahoo.com.tw

by Eberhart and Kennedy [15]. It is modeled to simulate the social behavior of bird flocks and follow similar steps as evolutionary algorithms finding near-optimal solutions. During the last two decades, PSO has been popularly applied to several optimization problems such as minimax problems [16], integer programming problems [17], global optimization problems [18], and other applications in engineering [19-20].

The convergence of traditional PSO algorithm is so fast that it easily falls into the local optimal solution. In order to resolve this problem, the main framework of this paper divides the solution space into *n* subspaces named subswarms in the PSO scheme. The optimal threshold value was obtained in accordance with the local maximum entropy for the best position ever attained by a proper particle and the global maximum uniformity indicated the best position ever encountered by all particles in the subswarms. The experimental results show that promising thresholds can be obtained by means of maximum entropy and global maximum uniformity which are embedded into fitness function of particles.

2. Relative algorithms

2.1 Maximum-entropy Thresholding

A maximum entropy threshold method, proposed by Jaynes [21], used the concept of Shannon entropy to image segmentation. It finds an appropriate grayscale threshold value which can divide pixels into the background and objects. And, let variances of pixels within these two regions be the largest amount, *L* represent the maximum range of the grayscale images and *t* is any one of the gray level value in the solution space. $H(t)$, represented the entropy value of an image, can be calculated as follows:

$$H(t) = H_{object}(t) + H_{background}(t) = -\sum_{i=0}^{t} \frac{p_i}{p_t} \ln \frac{p_i}{p_t} - \sum_{j=t+1}^{L-1} \frac{p_j}{1-p_t} \ln \frac{p_j}{1-p_t} \quad (1)$$

and

$$P_t = \sum_{i=0}^{t} P_i \text{ and } P_i = h(i)/N, \quad (2)$$

where $h(i)$ is the total pixels with the graylevel value i as well as N is the total number of pixels in an image. Therefore, we can find an optimal threshold t which deduces maximum entropy as follows:

$$\phi = \underset{0<t<L-1}{Arg\max} H(t) \quad (3)$$

To find the optimal multi-threshold values $t_1, t_2, \cdots, t_n$ in the multi-thresholding problem, Eq. (1) can be modified as

$$H(t_1, t_2, \cdots, t_n) = H(t_1) + H(t_2) + \cdots + H(t_n) = -\sum_{i=0}^{t_1-1} \frac{P_i}{P_{t_1}} \ln \frac{P_i}{P_{t_1}} - \sum_{j=t_1}^{t_2-1} \frac{P_j}{P_{t_2}} \ln \frac{P_j}{P_{t_2}} - \cdots - \sum_{k=t_n}^{L-1} \frac{P_k}{P_L} \ln \frac{P_k}{P_L} \quad (4)$$

where, $P_{T_1} = \left(\sum_{i=0}^{t_1-1} P_i\right), P_i = h(i)/N, P_{T_2} = \left(\sum_{j=t_1}^{t_2-1} p_j\right), P_j = h(j)/N$,

and $P_L = \left(\sum_{k=t_n}^{L-1} p_k\right), P_k = h(k)/N$.

Therefore, the fitness function, shown as in Eq. (5), is also defined the maximum entropy such that

$$Fit = \underset{0<t<t_n}{Arg\max} H(t_1, t_2, \cdots, t_n) \quad (5)$$

2.2 Uniformity

The uniformity is to evaluate the performance estimation for an image through thresholding processing. The greater the threshold value, the better the uniformity. The uniformity is defined as follows:

$$u = 1 - 2 * \frac{\sum_{j=0}^{k} \sum_{i \in R_j} \left(f_i - m_j\right)^2}{N * \left(f_{\max} - f_{\min}\right)^2} \quad (6)$$

Uniformity value is ranged between 0 to 1. Where k is the number of thresholding values. i, f_i and m_j are the number of pixels, graylevel of pixel i, and the average graylevel in a cluster, respectively. $f_{\max}$ and $f_{\min}$ are the maximum and minimum graylevel in the target image.

2.3 Particle Swarm Optimization

PSO, proposed by Kennedy and Eberhart [15], is a stochastic optimization algorithm that is also a population-based strategy to take advantage of a population (named *swarm*) of individuals (called *particles*) and search promising areas in the search space. Initially, PSO algorithm generates a group of particles, in which a particle is defined by a position and a velocity. Each particle moves with an adaptable velocity based on a fitness function within the search space, and retain the best position it ever visited in the past. The velocity decides the moving direction and distance for a particle. In PSO, two optimal constraint parameters *gbest* (global parameter indicated the best position ever encountered by all particles) and *pbest* (local parameter for the best position ever attained by a proper particle) are used to update the velocity for all particles. Finally, an approximating optimal solution can be obtained through previous actions iteratively.

Assume a search space with n dimension, $N \subset R^n$ is a swarm consisting of M particles. Then the position and velocity for the i-th particle on an n-dimension coordinate are defined

$$S_i = (x_{i1}, x_{i2}, ..., x_{in})^T \in N \quad (7)$$

and

$$V_i = (v_{i1}, v_{i2}, ..., v_{in})^T \in N \quad (8)$$

The best previous position (*pbest*) encountered by the i-th particle in N is defined as

$$pbest_i = (p_{i1}, p_{i2}, ..., p_{in})^T \in N . \quad (9)$$

In the meantime, the best previous position (*pbest*) among all the individuals of the swarm can be obtained by

$$gbest = \max(pbest_1, pbest_2, ..., pbest_M) \quad (10)$$

Therefore, the velocity $V_i(k+1)$ and position $S_i(k+1)$ for the i-th particle in the swarm at iteration k+1 can be defined as following equations:

$$\begin{aligned} V_i(k+1) = wV_i(k) + &\\ c_1 \times rand_1(pbest_i(k) - S_i(k)) &\\ + c_2 \times rand_2(gbest(k) - S_i(k)) & \end{aligned} \quad (11)$$

$$S_i(k+1) = S_i(k) + V_i(k+1) \quad (12)$$

where $i = 1,2,...,M$ is the index of particles, c_1 and c_2 are cognitive and social parameters, $rand_1$ and $rand_2$ are random numbers uniformly distributed between 0 and 1, and w, the weighting function, is defined as

$$w = w_{\max} - \frac{w_{\max} - w_{\min}}{k_{\max}} \times k \quad (13)$$

In Eq. (11), the cognitive component $pbest_i(k) - S_i(k)$ represents the particles own experience as to where the best position is while the social component $gbest(k) - S_i(k)$ stands for the belief of the entire swarm as to where the best solution is during iteration k. In Eq. (13), $w_{\max}$ is the initial weight as well as $w_{\min}$ and $k_{\max}$ are defined the weight and the number of the last iteration respectively. The higher the value of iteration, the smaller the weight w will be. In the proposed MEUPSOS, Eq (10) is modified as

$$subbest_k = \max(pbest_1, pbest_2, ..., pbest_i) \quad (14)$$

$$gbest = \max(subbest_1, subbest_2, ..., subbest_k) \quad (15)$$

3. The proposed PSO based on maximum entropy and uniformity

In this paper the constraints with maximum entropy and uniformity were embedded into the PSO algorithm with subswarm structure (named MEUPSOS). Swarm in the MEUPSO was divided into several subwarms in order to find several local optimal solutions to speed up global or near global threshold values being obtained. In order to upgrade the computation performance, we just used only one-swarm particles for all subswarms instead of one-population particles for each swaem.

In the proposed MEUPSOS, we added local optimal value named *subbest* for each subswarm. First, we used the entropy-based fitness function to update the parameter *pbest*. Then parameters *pbest*s were used to update *subbest* in a subswarm. Finally, *gbest* was updated with *subbest*s in the global swarm. The detail flowchart of the proposed MEUPSOS is shown as in Figure 1. In Figure 1, the processing flow went through two sub flowcharts named (A) and (B) to calculate the entropies for updating *subbest*s and compute uniformity to adjust the global parameter *gbest*. These two sub flowcharts are shown as in Figure 2 and Figure 3.

The main purpose in Figure 2 is to calculate the entropy-based fitness function of particles in every subswarm and updating parameters *pbest* with maximum entropy. Then, *pbest*s were used to update *subbest* in a subswarm. In Figure 3, parameter *gbest* was updated by using of the best *subbest* with uniformity-based fitness function in subswarms. Additionally, the threshold values were updated with the better uniformity.

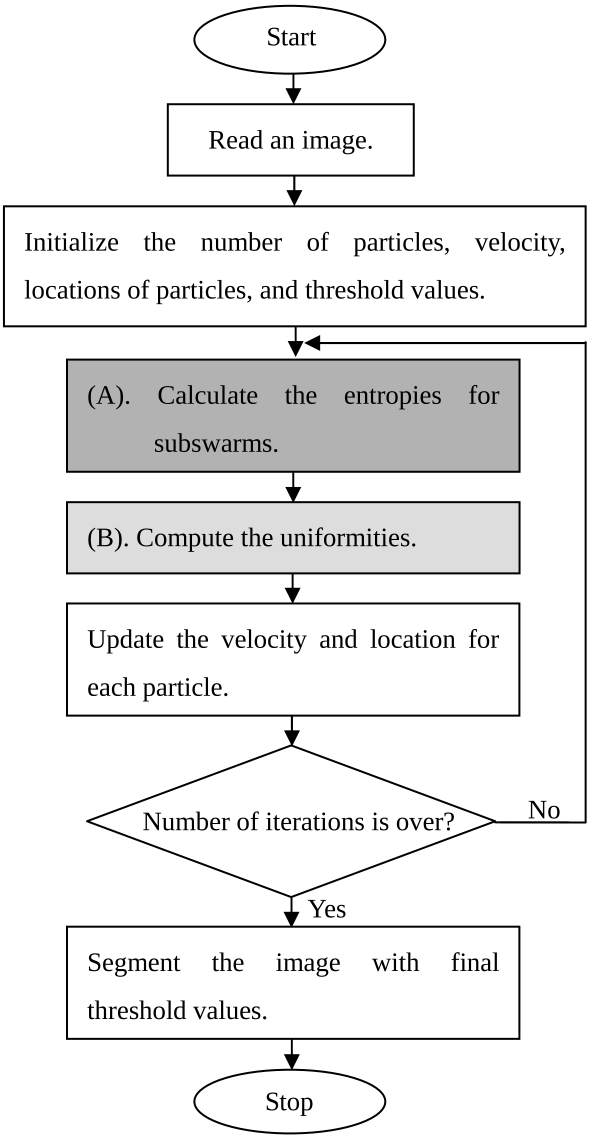


Figure 1: Flowchart of the proposed MEUPSO.

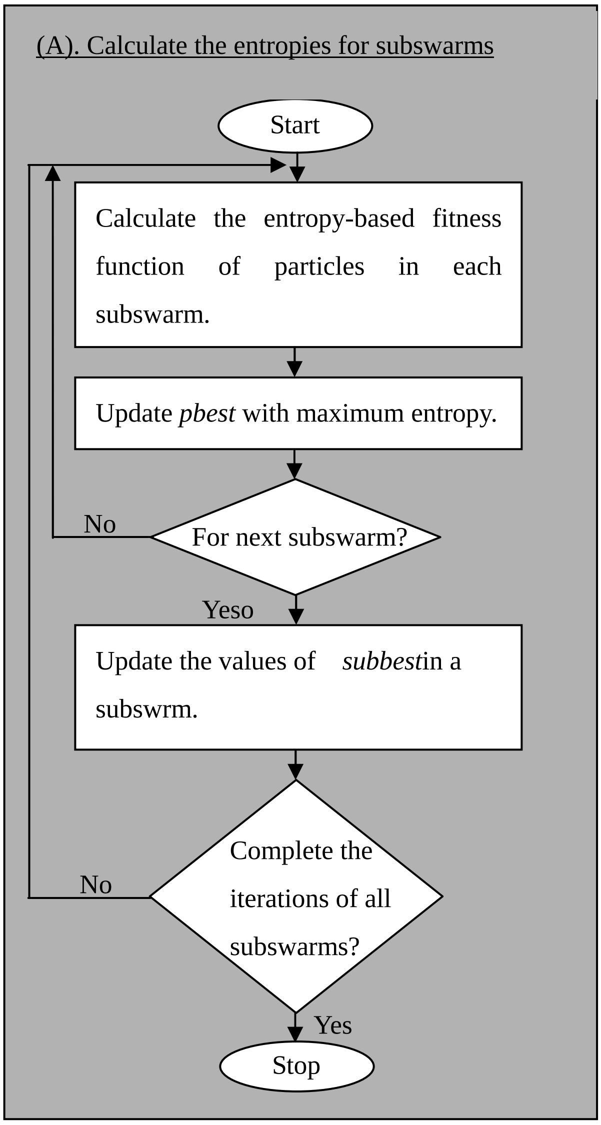


Figure 2: Flowchart of block (A) in Figure 1.

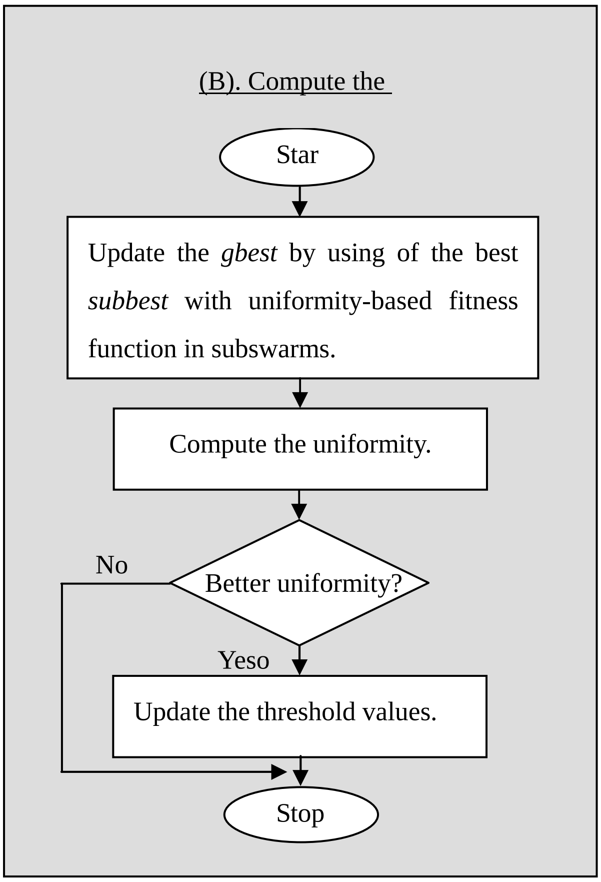
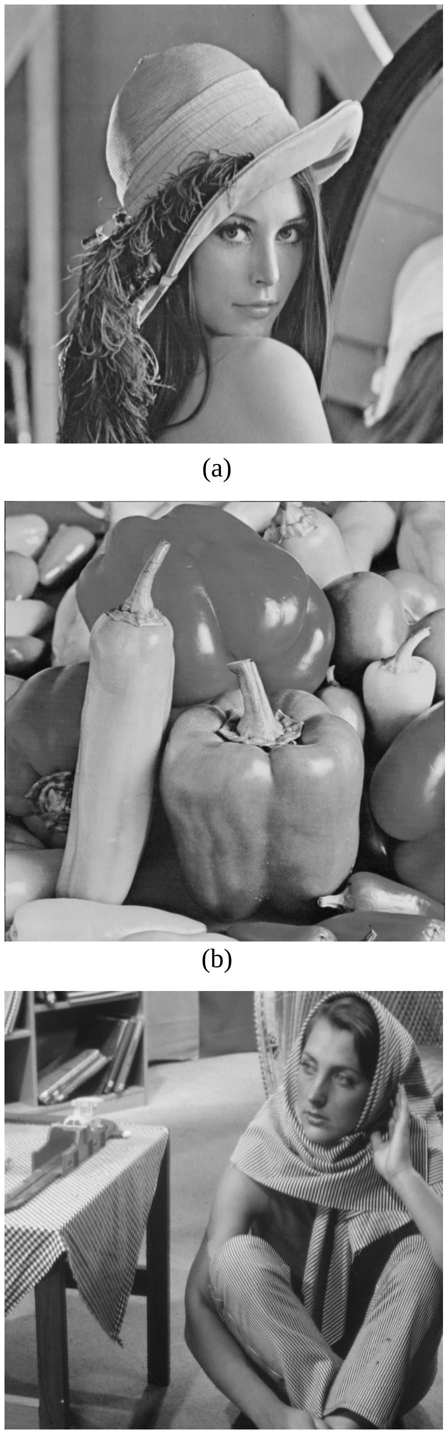


Figure 3: Flowchart of block (B) in Figure 1.

4. Experimental results

In order to show the performance, all simulations are executed with the interpreter language of MATLAB in a personal computer. The performance for the proposed MEUPSOS was compared with HCOCLPSO [11]. The original 512x512 test images (Lena, Pepper, and Barbara) are shown as in Figure 4. The segmented results with 6 clusters by the proposed MEUPSOS are shown as in Figure 5. A comparative study of uniformity for the proposed MEUPSOS and HCOCLPSO is shown as in Table I. From Table I, we can find that the uniformity by the proposed MEUPSOS is better than those got by the other algorithm HCOCLPSO.



(a)

(b)

(c)

Figure 4: Original 512x512 images; (a) Lena, (b) Pepper, and (c) Barbara.

(a)

(b)

(c)

Figure 5: Segmented results with 6 clusters by the proposed MEUPSOS

Table 1: A comparative study of uniformity for the proposed MEUPSOS and HCOCLPSO

Image	k	Uniformity	
		Proposed MEUPSOS	HCOCLPSO
Lena	2	0.9889	0.9883
	3	0.9899	0.9886
	4	0.9904	0.9893
	5	0.9920	0.9903
Pepper	2	0.9878	0.9876
	3	0.9890	0.9875
	4	0.9918	0.9914
	5	0.9916	0.9914
barbara	2	0.9898	0.9893
	3	0.9911	0.9901
	4	0.9925	0.9911
	5	0.9921	0.9919

Although the object values of entropy obtained by the proposed MEUPSOS are smaller than those got by the HCOCLPSO with a tiny value, the used memory resource in the proposed MEUPSOS are less than the HCOCLPSO. No matter what the number of threshold values is defined in advance by the proposed MEUPSOS. If an image was segmented *k* clusters and number of particles is *m*, the memory resource was **4*(*m*+*k*)+2** (= 102 locations, if *m*=20 and *k*=5.) for the proposed MEUPSOS and **(4*m+2)*k** (= 410 locations, if *m*=20 and *k*=5.) for HCOCLPSO. Therefore, the computation performance in the proposed MEUPSOS is better than the HCOCLPSO.

Table 2: A comparative study of entropy for the proposed MEUPSOS and HCOCLPSO

Image	k	Entropy			
		Proposed MEUPPSO		HCOCLPSO	
		Optimal thresholds	Object value	Optimal thresholds	Object value
Lena	2	88, 165	17.76	97, 164	17.8130
	3	72, 125, 179	22.0278	82, 126, 175	22.0993
	4	70, 117, 163, 197	25.8766	64, 97, 138, 179	25.9864
	5	53, 86, 123, 160, 207	29.1873	63, 94, 128, 163, 194	29.7348
Pepper	2	70, 146	18.0233	72, 144	18.0280
	3	56, 109, 164	22.4350	53, 102, 155	22.4694
	4	50, 98, 146, 196	26.3321	53, 98, 143, 191	26.5234
	5	35, 71, 106, 141, 191	30.3217	42, 74, 109, 149, 191	30.3483
barbara	2	90, 169	18.2646	98, 163	1832765
	3	72, 135, 187	22.6534	76, 127, 178	22.7182
	4	60, 111, 168, 208	26.6182	61, 100, 143, 187	26.7712
	5	66, 99, 135, 170, 208	30.5861	57, 92, 129, 166,199	30.6238

TABLE 3: A comparative result of memory resources for the proposed MEUPSOS and HCOCLPSO

# of Clusters	# of particles：m	Memory Resource			
		Proposed MEUPSOS		HCOCLPSO	
k	P_best	m*2	Needed 4*(m+k)+2	m*2*k	Needed (4*m+2)*k
	G_best	2		2*k	
	position	m		m*k	
	velocity	m		m*k	
	Sub_swarm	4*k		0	

5. Conclusions

In this paper, we proposed a PSO algorithm by maximum entropy and uniformity with subswarm structure named MEUPSOS for image segmentation. In order to upgrade the computation performance, the number of particles was used with a fixed value no matter how the threshold values were changed. Although tiny poor object values were obtained, the proposed MEUPSOS can get better uniformity and computation performance than those obtained by the HCOCLPSO for the segmented images.

Acknowledgment

This work was supported by the National Science Council, TAIWAN, under the Grants NSC100-2221-E-167-030 -

References

[1] T.Pavlidis, and YTLiow, “Integrating region growing and edge detection,” IEEE Trans. on Pattern Anal. Mach. Intell., vol. 12, pp. 225-233, 1990.

[2] K. S. Fu, and J. K. Mu, “A survey on image sementation,” Pattern Recognition, vol. 13, pp. 3-16, 1983.

[3] K. S. Cheng, J. S. Lin, and C. W. Mao, “The application of competitve Hopfield neural network on medical image segmentatin,” IEEE Trans. on Medical Imaging, vol. 15, pp. 560-567, 1996.

[4] J. S. Lin “Segmentation of medical images through a penalized fuzzy Hopfield network with moments preservation,” J. of the Chinese Institute of Eng., vol. 23, pp. 633-643, 2000.

[5] P. K.Sahoo, S. Soltani, A. K. C. Wong, and Y. C. Chen, “A survey of thresholding techniques,” Comput. Vision, Graphic, and Imge Processing, vol. 41, pp. 233-260, 1988.

[6] M. Mairta, and A. Chatterjee “A hybrid cooperative-comprehensive learning based PSO algorithm for image segment using multilevel threshilding,” Expert System with Applications, vol. 34, pp. 1341-1350, 2008.

[7] J. Kittler, J. Illingworth, “Minimum error threshold,” Pattern Recognition, vol. 19, lssue 1,pp. 41-47,1986.

[8] D. Liu, J. Yu, “Otsu method and K-means,” Hybrid Intelligent System,” vol. 1, pp. 344-349, 2009.

[9] S. H. Liu, J. S. Lin, and Z. S. Lin, “An Annealed Ant System to the Shortest-path Network Problem,” Int. J. of Electrical Engineering, vol. 13, pp. 209-217, 2006.

[10] H. Tang, C. Wu, L. Han, and X. Wang, “Image segment based on Improved PSO” Int. Conf. on Computer and Communication Technologies in Agriculture Engineering, pp. 191-194, 2010.

[11] D. Karaboga, B. Basturk, “Artificial bee colony (abc) optimization algorithm for solving constrained optimization problems,” LNCS: Advances in Soft Computing: Foundations of Fuzzy Logic and Soft Computing, vol. 4529, Springer-Verlag, 2007, pp. 789-798,.

[12] D. Karaboga, B. Basturk, “A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm,” J. of Global Optimization, vol. 39, pp. 459-171, 2007.

[13] M. Awad, K.Chehdi, and A. Nasri, “Multi-component image segmentation using a hybrid dynamic genetic algorithm and fuzzy C-means,” Image Processing, IET, vol. 3, pp. 52-62, 2009.

[14] M. Hanmandlu, O. P. Verma, N. K. Kumar, and M. Kulkarni, “A novel optimal fuzzy system for color image enhancement using bacterial foraging,” IEEE Trans. on Instrumentation and Mmeasurement, vol. 58, pp. 2867-2879, 2009.

[15] R. C. Eberhart and J. Kennedy, “A new optimizer using particle swarm theory,” in Proc. 6th Symp. Micro Machine and Human Science, Nagoya, Japan, pp. 39-43, 1995.

[16] E. C. Laskari, K. E. Parsopoulos, and M. N. Vrahatis, “Particle swarm optimization for minimax problems,” in Proc. IEEE 2002 Congr. Evolutionary Computation, Honolulu, pp. 1582-1587, 2002.

[17] E. C. Laskari, K. E. Parsopoulos, and M. N. Vrahatis, “Particle swarm optimization for integer programming,” in Proc. IEEE 2002 Congr. Evolutionary Computation, Honolulu, pp. 1576-1581, 2002.

[18] K. E. Parsopoulos, and M. N. Vrahatis, "On the computation of all global minimizers through particle swarm optimization," IEEE Trans. on Evolutionary Computation, vol. 8, pp. 211-224, 2004.

[19] M. G. Omran and A. P. Engelbrecht, "A Color Image Quantization Algorithm Based on Particle Swarm Optimization," Informatica, vol. 29, pp. 261-269, 2005.

[20] M. A. Abido, "Optimal design of power system stabilizers using particle swarm optimization," IEEE Trans. Energy Conversion, vol. 17, pp. 406-413, 2002.

[21] E. T. Jaynes, "On the rotational of maximum entropy methods," IEEE proceedings, vol. 70, pp. 939-952, 1982.