Journal of Information Hiding and Multimedia Signal Processing ©2012 ISSN 2073-4212
Ubiquitous International Volume **3**, Number **3**, July **2012**

Bacterial Foraging Particle Swarm Optimization Algorithm Based Fuzzy-VQ Compression Systems

Hsuan-Ming Feng
Department of Computer Science and Information Engineering, National Quenoy University,
No. 1 University, Rd., Kin-Ning Vallage Kinmen, 892, Taiwan, R.O.C.
hmfeng@nqu.edu.tw

Ji-Hwei Horng
Department of Electrical Engineering, National Quenoy University,
No. 1 University, Rd., Kin-Ning Vallage Kinmen, 892, Taiwan, R.O.C.
horng@nqu.edu.tw

Shiang-Min Jou
Department of Electronic Engineering, Xiamen University,
Xiamen Fujian 361005; China
sammy331@pchome.com.tw

Received November, 2011; revised July, 2012

Abstract. *This study proposes a novel bacterial foraging swarm-based intelligent algorithm called the bacterial foraging particle swarm optimization (BFPSO) algorithm to design vector quantization (VQ)-based fuzzy-image compression systems. It improves compressed image quality when processing many image patterns. The BFPSO algorithm is an efficient evolutionary learning algorithm that manages complex global optimal codebook generation problems. The BFPSO algorithm combines bacterial foraging optimization (BFO) behavior with a particle swarm optimization (PSO) learning scheme to obtain fast convergence and self-adaptive learning benefits. The evolutionary BFPSO algorithm automatically designs appropriate parameters for fuzzy-VQ-based systems using a proper codebook selection machine. Computer simulation results of nonlinear image compression applications demonstrate the efficiency of the BFPSO learning algorithm. The differences between the proposed BFPSO learning scheme and the BFO- and LBG-based VQ learning methods demonstrate the superior image results produced by the proposed algorithm.*
Keywords: Vector quantization, Fuzzy logic, Image compression systems, Bacterial foraging particle swarm optimization, Evolutionary learning algorithm.

1. Introduction.

Vector quantization (VQ) image compression technology is important for establishing a significant codebook to represent features of large amounts of image data. It improves transmission and storage speeds of multimedia, video, medical diagnosis, and interpretation systems [7]. The VQ concept is an efficient data-encoding and -decoding method that has been widely used to develop several image compression application services [1-3, 11]. Codebook generation is a data-clustering process where training vectors are classified into specific code words based on the minimization of average distortion between training vectors and codebook vectors. In the data-encoding phase, determining the selected optimal codebook characterizes the whole image. Similar to the encoding phase, the VQ decoding process assigns a smaller, but proper, codebook to the related code vector index, which represents the whole original image well. The selected optimal codebook preserves the highest possible fidelity, that is, the smaller compressed image is the least distorted from the original image. Therefore, the compression rate must be as high as possible. If the codebook is much smaller than the original image data, a high image compression rate has been achieved. The smallest data set to be computed determines the rate of compression. This reduces the memory required and decreases the transmitted load of the channel bandwidth.

The main problem with image compression applications is that they must perform at the highest compression rate while achieving perfect recoverable images. To achieve these two goals, a VQ-based image compression system is used to extract the optimal codebook. K-means, such as Generalized Lloyd-type clustering algorithms, are a popular method of generating realizable codebooks. However, the gradient descent-type learning methodology is sensitive to the effect of initial parameter settings. The Generalized Lloyd-type learning scheme, called an LBG algorithm [18-19], is a recognized method of establishing a desired codebook. To execute an LBG learning scheme, let one codebook contain M code vectors for random selection in the initial condition. The Euclidean norm metric is then iteratively exploited to yield the appropriate codebook, which minimizes the average distortion between training image patterns and code vectors. The Euclidean norm metric and gradient descent-based iterative learning scheme of an LBG algorithm cannot contain global adaptive search capabilities. Its results are likely to stop at the local optimal solution. Generally, the hard-level measure-based LBG algorithm only accounts for point-to-point mapping to encode an image vector into one crisp code vector. Such LBG-based learning algorithms are used to solve specific image compression problems, where the distribution and shape of image patterns are poorly understood and well-separated, respectively. However, real image-processing patterns are always flexible and have irregular neighboring regions. The hard evaluation method cannot accomplish complex codebook design. It ignores the possibility that training image patterns belong to various code vectors.

Zadeh introduced fuzzy set theory in 1965 [9]. A visible fuzzy measure mechanism with a flexible membership function is manipulated to estimate the similarities between two training and testing data sequences. This research uses a typical Gaussian membership function to gently acquire the suitable codebook from training image patterns. Bezdek developed the fuzzy c-mean (FCM) in 1981 [5]. Its concepts of flexible shapes and different membership function degrees efficiently group data points into desired categories in the whole operating space. Because of the fuzzy soft estimator adaptation, several fuzzy-type VQ schemes have been used to achieve optimal codebooks [10, 14-17]. However, two main problems must be solved. Extensive knowledge is required to create the learning scheme, to select proper membership functions that are used to approach suitable codebooks. The other problem is that the best choice of possible parameter combinations is highly related to the specific shapes and boundaries of trained image patterns.

Particle swarm optimization (PSO) is an evolutionary computation and intelligent swarm technique that was inspired by the social behavior of birds flocking or fish schooling. Eberhart and Kennedy introduced it in 1995 [6]. PSO is learned from a scenario, and some ill-defined nonlinear VQ problems are created to be solved by the PSO learning algorithm [2]. Two factors establish the swarm-like computation learning machineXthe personal best solution for a particle and the global best solution for a particle in a whole swarm. Because of its simple learning strategy, the common PSO learning algorithm is likely to stop at the local optimal solution. Bacterial foraging optimization (BFO) was introduced in 2002 [8]. This learning procedure contains probabilistic training concepts by simulating the natural behavior of Escherichia coli. Foraging strategy patterns inspired the development of BFO to successfully achieve desired parameter settings. BFO mimics the behavior of E. coli bacteria, which exhibit chemo taxi, swarming, tumbling, reproductive, elimination, and dispersal activities. The complexity of BFO encouraged researchers to work toward its simplification and faster convergence. The novel bacterial foraging particle swarm optimization (BFPSO) learning algorithm integrates the benefits of the BFO global search ability and the PSO fast convergence learning machine to minimize their weaknesses. The population-based BFPSO learning scheme solves ill-defined, nonlinear, complex, multidimensional optimization problems [3-4, 8, 13]. This study proposes a BFPSO algorithm to design the proper fuzzy inference to acquire an optimal codebook. The BFPSO algorithm is an autonomous fuzzy rule tuning mechanism that creates appropriate parameters for a fuzzy system to avoid the time-consuming trial-and-error method. The selected fuzzy systems with desired linguistic rules achieve a large image-compression rate. This study uses the novel BFPSO algorithm to establish the VQ-based image-compression fuzzy system to reduce trial-and-error parameter-tuning time wastage. The evolutionary BFPSO learning algorithm, directed by a specific fitness function, is an efficient method of obtaining approximate optimal codebooks in a large, complex image space.

2. Fuzzy VQ System Design.

Figure 1 shows a diagram of the BFPSO algorithm-based fuzzy-VQ image compression system. The original image patterns were applied to the fuzzy-VQ-based analysis scheme. The organized fuzzy-based inference vector quantization (FVQ) system contained N-input training vector $\underline{x}$ and M-codebook in the part of jth codebooks (C_j), which were generated by the following fuzzy inference machine:

$$R^i : IF\ \underline{x}\ is\ ME_i\ THEN\ \underline{x}\ belong\ to\ C_{(j_1 j_2,\ldots,j_n)}\ with\ CF = CF_{(j_1 j_2,\ldots,j_n)} \tag{1}$$

where $j \in \{1, 2, \ldots, M\}$; the form of vector $\underline{x} = (x_1, x_2, \ldots, x_n)$ is the input pattern; and ME_i is the fired fuzzy inference machine with membership function combinations in the principle part of the fuzzy inference system for the related ith fuzzy rules (R^i). The selected $C_{(j_1 j_2,\ldots,j_n)}$ is the code vector classification in the consequent part of the fuzzy inference system. n and M are the number of membership functions and fuzzy codebooks, respectively. CF is the certainty value in this inference. The FVQ procedure uses $\underline{x}$ to extract the proper $C_{(j_1 j_2,\ldots,j_n)}$ th code vector with the evaluated grade of the corresponding ith fuzzy rule R_j. This paper uses the following equations to find the mathematical fuzzy inference:

$$ME_i\left(\underline{x}\right) = HC_{(j,1)}\left(x_1^t\right) \times HC_{(j,2)}\left(x_2^t\right) \times \cdots \times HC_{(j,n)}\left(x_n^t\right) \tag{2}$$

$HC_{(j,n)}$ is the th fuzzy set for the jth fuzzy rule in the fuzzy system. It is described as

$$HC_{(j,i)}\left(x_j; c_{ji}, \gamma\right) = \exp\left(-\left(\frac{\left(x_i - c_{ji}\right)^2}{\gamma^2}\right)\right) \tag{3}$$

where c_{ji} is the centered position of the Gaussian-type membership function and γ is the distributed factors. The concept shows that the training image patterns are easily obtained by fuzzy partition metrics with different membership function parameter settings. Thus, the developed membership function acts as a soft evaluator to forecast the similarity between image data $\underline{x}_i$ and the jth code vector. In the proposed fuzzy VQ scheme, the number of code vectors is the same as that of the fuzzy rules. In the fuzzy inference procedure, the certainty value CF_j^t of the associated jth code vector is determined in advance. Available membership function contours are set with feasible shapes and different center positions to approximate proper fuzzy partitions of the fuzzy system to represent the complex codebook design. The goal of this learning machine is to obtain the maximal membership function value grade while the selected code vector approaches the most similar image pattern. The $C_{(j_1 j_2,\ldots,j_n)}$th setting is an active code vector when it contains the maximal CF_j^t value. The formula for this is

$$CF_X^t = MAX\left\{CF_1^t, CF_2^t, \cdots, CF_M^t\right\}. \tag{4}$$

This inference scheme for the fuzzy VQ-based image compression system can be used to adaptively estimate the most codebooks from variant image data with the maximal certainty value selection. The flowchart in Fig. 1 shows that the recognized codebook segmentation is transformed to reconstruct the image patterns.

The aim of fuzzy VQ-based inference systems is approbatory codebook design to achieve the desired image. This is achieved by several membership functions flexibly distributed in an n-dimensional space to form the desired fuzzy inference system. To efficiently extract the optimal or near-optimal codebook, the proposed BFPSO learning algorithm produces suitable fuzzy inference system parameters-guided by the fitness function. Therefore, the appropriate parameter selections for the evolutionary learning algorithms are required to create the fuzzy inference system in this search space. The evolutionary BFPSO-based learning algorithm is used to determine the proper parameter set in the search space. The next section presents the relevant discussion.

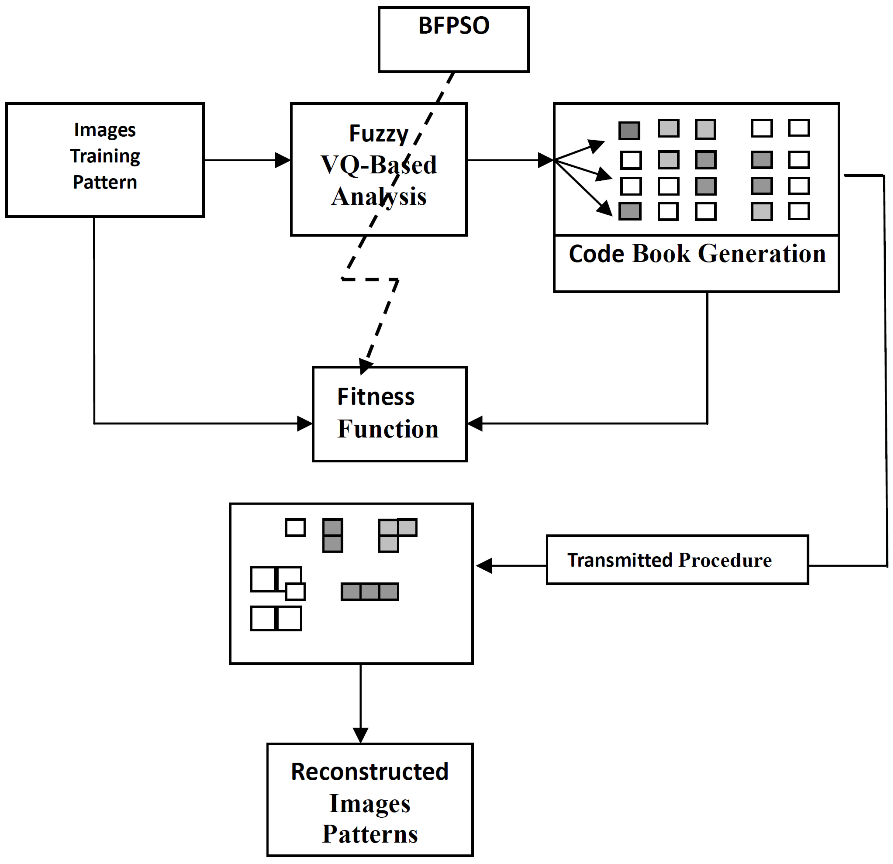


Figure 1. Diagram of the BFPSO fuzzy-VQ compressed image system design.

3. **Optimal Fuzzy-VQ System Design with the BFPSO Learning Algorithm.**

This study uses the BFPSO learning algorithm to develop a fuzzy VQ-based image compression system. The BFPSO algorithm combines the natural behavior from the BFO and PSO strategies to achieve the desired objective. The PSO learning scheme mimics and simulates intelligent swarm behavior with a computer. The PSO learning algorithm exchanges heuristic information from knowledge and experience, which is better at discovering superior offprints. In the PSO learning cycle, each particle contains related position value X and velocity value Y, which are regulated by the two best values Pbest_x and Gbest_x, respectively. Here, Pbest_x is the best solution for an individual particle, which has obtained the highest fitness value thus far. The selected Gbest_x is the best value achieved by all particles. The benefit of the PSO learning algorithm is simple and efficient; the relevant Pbest_x and Gbest_x values teach particle velocity in each learning step. The new velocity for each particle is updated using the following equation [4]: $v_{p,d}(k+1) = v_{p,d}(k) +$

$$\alpha_1(k+1)\left(Pbest - x_{p,d}(k+1) - x_{p,d}(k)\right) + \alpha_2(k+1)\left(Gbest - x_{p,d}(k+1) - x_{p,d}(k)\right) \quad (5)$$

where $v_{p,d}$ is the velocity of the pth particle for the dth dimensional variable; the value of the pth particle position in the dth dimension is $x_{p,d}$; p is the particle number; k is the current state; $k+1$ is the next time step; and $\alpha_1(k+1)$ and $\alpha_2(k+1)$ are selected random numbers between 0 and 1.

If the velocity of the particle is obtained, the particle location is modified in the next time step.

$$x_{p,d}(k+1) = x_{p,d}(k) + v_{p,d}(k+1) \quad (6)$$

Based on the benefits of the PSO and BFO learning algorithms, the BFPSO solves the nonlinear complex codebook selection problem in VQ-based image compression. The bionic BFO is a novel evolutionary learning algorithm; it mimics the behavior of E. coli bacteria to improve searching. BFO consists of four stages-taxis, dispersal, reproduction, and elimination-to identify the nearest optimal parameter set. The next three sections present a discussion on the BFO concept.

3.1. **Bacterium Taxis.**

The action of taxis is reasonable natural behavior that occurs while the E. coli bacterium is stimulated by its surroundings. The E. coli bacterium usually uses two methods to move-swimming and tumbling. In the swimming cycle, an opposite force is applied to the bacterium, which allows the flagellum to rotate in a counterclockwise direction. This creates an opposite force, with the swimming action randomly pushing the bacterium cell into different positions. In the tumbling cycle, the bacterium is idle. The two methods can change its direction. The movement of the ith bacterium is described by

$$P^s(f+1, g, u) = P^s(f, g, u) + C(s) * V(f) \quad (7)$$

where $P^s(f, g, u)$ is the sth location of the bacterium at the fth chemotactic, gth reproductive, and uth elimination steps. $C(s)$ is the length of one walking cycle. Here, it is defined as a small constant value. $V(f)$ is the direction angle of the fth chemotactic step; its default value is set at a range of $[0, 2\pi]$.

3.2. **Swarming Dispersal.**

The objective of the E. coli bacterium is to extract the best parameter setting in the search space. Cell-to-cell communication allows every bacterium to disperse its own message to others. Each bacterium also releases a repellent function signal for others to be at a minimal distance from its current position. Based on this dispersal behavior in a suitably sized swarm, bacterium position can be approximated as the area that contains the best nutrients. Cell-to-cell swarming dispersal action is described by

$$J_\alpha^s = (P, P^s(f,g,u))$$

$$= -d_{attract}\exp\left[-\overline{w}_{attract}\sum_{x=1}^{q}(p_{opi}-p_x^s)^2\right] + h_{repellent}\exp\left[-\overline{w}_{repellent}\sum_{x=1}^{q}(p_{opi}-p_x^s)^2\right] \quad (8)$$

where q is the number of bacteria; $d_{attract}$ and $\overline{w}_{attract}$ are the attractive coefficients; $d_{repellent}$ and $\overline{w}_{repellent}$ are repellent coefficients; and p_{opi} is the best solution in the current learning cycle.

3.3. **Reproduction and Elimination.** Bacteria have a natural tendency to gather in nutrient-rich areas through an activity called bacterium taxis. After bacterium taxis and dispersal procedure steps, reproduction and elimination tasks improve performance in similarly sized swarms. Generally, 50% of bacteria are reproduced by fitness value evolution. In this learning cycle, half a bacteria population is high performance and is set in a live state to reproduce. Elimination action kills the other half. These vigorous bacteria are selected into the swarm population regeneration pool. Therefore, existing bacteria are more likely to remain near optimal positions. Swarm dispersion occurs after a certain number of reproduction and elimination procedures. Other influences change the life of the bacteria population suddenly or gradually. This process prevents local minimal trapping events. The novel evolutionary BFPSO learning algorithm is developed for use in a fuzzy VQ-based image compression system. Figure 2 shows the flowchart of the proposed learning algorithm and the following section discusses the BFPSO learning steps.

BFPSO 1) The required BFPSO parameters must be selected. These are bacteria size, input variables, number of parameters, swinging unit length, number of chemotactic learning loops, number of reproductive learning loops, attractive and repellent coefficients, and codebook size. The learning rate (c1, c2) and inertia factor for the PSO learning scheme must be selected. Swarm position is randomly generated to form the initial fuzzy VQ-based image compression system.

BFPSO 2) The proper fitness function must be defined. The fitness function correctly measures the distortion between the original image and the selected codebook data. The proposed fitness function is

$$Fit\_VQ = \frac{k_0}{\sum_{i=1}^{N}\sum_{j=1}^{M} S_{ij}\cdot\|\underline{x}_i - c_j\|^2}, \quad (9)$$

where N is the number of training vectors, and positive constant $k_0$is selected by the designer. The definition of $S_{ij}$is described by

$$S_{ij} = \begin{cases} 1 & if\ CF(\underline{x}_i, c_j) = Arg\ \ MAX\{CF_1, CF_2, \cdots, CF_M\} \\ 0 & other \end{cases} \quad (10)$$

where, $\underline{x}_i$ and c_j are the ith original image pattern and the selected jth code vector, respectively. Less error between the original image pattern and the code vector produces less distortion. The objective of the defined fitness unction is to produce minimal error, that is, $MIN\left\{\sum_{i=1}^{N}\sum_{j=1}^{M} S_{ij}\cdot\|\underline{x}_i - c_j\|^2\right\}$.

Therefore, parameters are selected to achieve the maximal fitness value, which is defined by the following formula:

$$MAX\left(Fit\_VQ\right). \tag{11}$$

BFPSO 3) Execute the bacterium taxis learning cycle using (7).

BFPSO 4) Run the swarm dispersal learning cycle using (8).

BFPSO 5) Evaluate and select the personal best solution ($pBest$) using the following equation [5]:

$$pBest_p^{t+1} = \begin{cases} Y_p^{t+1} & if \quad F\left(Y_p^{t+1}\right) \geq F\left(pBest_p^t\right) \\ pBest_p^t & if \quad F\left(Y_p^{t+1}\right) < F\left(pBest_p^t\right) \end{cases}. \tag{12}$$

BFPSO 6) Evaluate and select the global best solution ( ) using the following equation:

$$gBest^{t+1} = \begin{cases} pBest_p^{t+1} & if \quad Fit\_VQ\left(pBest_p^{t+1}\right) \geq Fit\_VQ(gBest^t) \\ gBest^t & if \quad Fit\_VQ\left(pBest_p^{t+1}\right) < Fit\_VQ\left(gBest^t\right) \end{cases}. \tag{13}$$

BFPSO 7) Regulate the related position in the PSO learning cycle using (5) and (6).

BFPSO 8) Execute the reproduction and elimination procedure.

BFPSO 9) Repeat Steps 2-8 until g=G.

BFPSO 10) Select the best optimal codebook using the proposed fuzzy inference analysis with the global best parameters set ($gBest$) to generate the desired fuzzy VQ-based image compression system.

4. **Illustrated Studies.**

Lena and Peppers Examples: The evolutionary BFPSO algorithm was developed with the self-learning ability to automatically generate the fuzzy codebook and produce a good image compression system. This experiment used the original gray-scale image patterns, Lena and Peppers-both 256 x 256 pixels-as test material. The peak-signal to noise ratio (PSNR) was used to evaluate the reconstructed image quality. The PSNR is defined as follows:

$$PSNR = 10\log_{10}\frac{255^2}{\frac{1}{N\times N}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left(f\left(i,j\right)-\hat{f}\left(i,j\right)\right)^2}dB, \tag{14}$$

where $N \times N$ is the size of the original image and $f\left(i,j\right)$ and $\hat{f}\left(i,j\right)$ are the gray-level pixel values of the original and reconstructed images, respectively. Two original images, Lena and Peppers, were used to demonstrate the adaptive capacities of the BFPSO, BFO, and PSO learning methods using the VQ machine. The experiment was conducted using different-sized codebook design problems, such as M=4, 8, 16, 32, 64, 128, and 256. The BFPSO, BFO, and PSO learning algorithms were executed 3 times on the images. Figures 3 and 4 show the evaluated PSNR of various codebook sizes of the original Lena and Peppers images, respectively. The software simulation showed that the performance disparity among BFO, PSO, and BFPSO becomes larger as the codebook size increases. The other obvious circumstance in the representation of these two examples can be found in this study. The three PSO and BFO simulationsXwhich used the same training dataXwere stuck in the local optimal codebook selection when they encountered variant initial conditions. The BFO learning algorithm learns by natural selection and tends to eliminate poor solutions. Because it incorporates the PSO search engine, the proposed BFPSO learning algorithm improved PSO and BFO local problems, retaining a higher PSNR, even when several image patterns for training different-sized code vectors were encountered. Most training codebooks produced by the BFPSO learning scheme achieved perfect image compression results in three running cycles. It decrypted the ability of the BFPSO learning algorithm in the VQ-based fuzzy image compression scheme, which contained a powerful and adaptive facility to manage the complex codebook design problem under various initial conditions. Table 1 shows that the BFPSO algorithm produces higher PSNRs than the PSO and BFO learning methods for Lena and Peppers images.

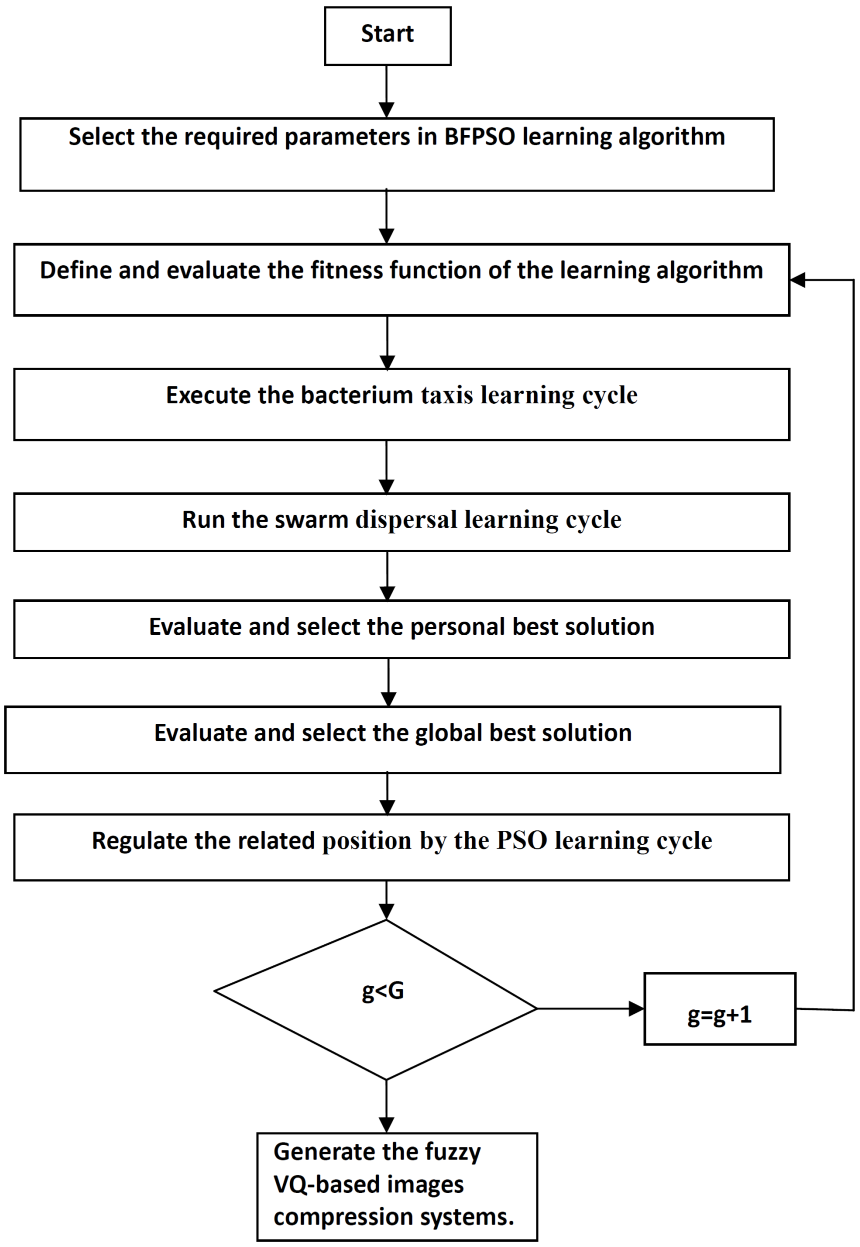

Start
Select the required parameters in BFPSO learning algorithm
Define and evaluate the fitness function of the learning algorithm
Execute the bacterium taxis learning cycle
Run the swarm dispersal learning cycle
Evaluate and select the personal best solution
Evaluate and select the global best solution
Regulate the related position by the PSO learning cycle
g<G
g=g+1
Generate the fuzzy VQ-based images compression systems.

Figure 2. Flowchart of the BFPSO-based fuzzy-VQ image compressed system.

Figures 5(a) and 6(a) show the original Lena and Peppers images used in the second experiment, respectively. The BFPSO, BFO, and PSO learning algorithms were applied to the Lena image with codebook sizes of 128 and 64. Figures 5(b)5(g) show the software simulation results of the BFPOS, BFO, and PSO learning algorithms. In a similar experiment, the original Peppers image was also used to test the performance of the BF-PSO, BFO, and PSO methods with codebook sizes of 128 and 64. Figures 6(b)6(g) show the simulation results. The results show that the proposed BFPSO leaning algorithm improves performance in complex fuzzy-VQ codebook design problems.

5. **Conclusion.** This research develops a set of available membership functions to produce appropriate fuzzy partitions for recognizing perfect codebooks with fuzzy soft inference analysis decisions. It attempts to improve adaptable fuzzy image compression systems in complex search spaces. Based on the integration of the heuristic BFO and PSO learning schemes, the novel BFPSO algorithms efficiently tune variable parameter combinations of the fuzzy VQ-based image compression system. Results show that the proposed BFPSO learning method produces better results than the commonly used BFO and PSO learning algorithms. In real-world image compression system design, information received is always vague and variant. It is difficult for the BFO and PSO learning schemes to produce perfect results in such variant conditions. Local optimization and bad codebook problems occur in such conditions. The BFPSO learning scheme improves performance in the generated fuzzy VQ-based image compression applications. The software simulation shows that the algorithm is adaptable and capable of performing in a vague and variant environment. It produces better codebooks to reconstruct compressed images with high fidelity.

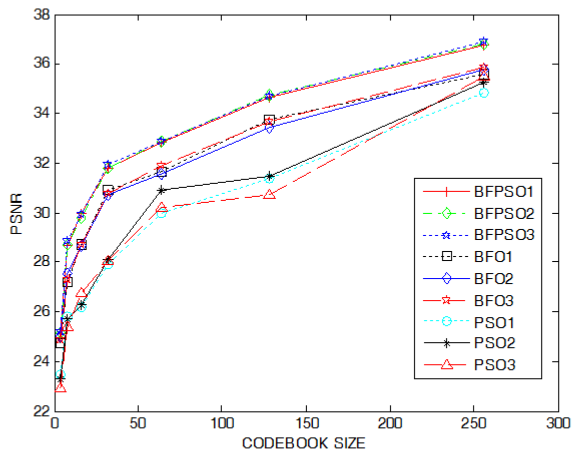


FIGURE 3. PSNR in dB versus the size of the codebook for "Lena".

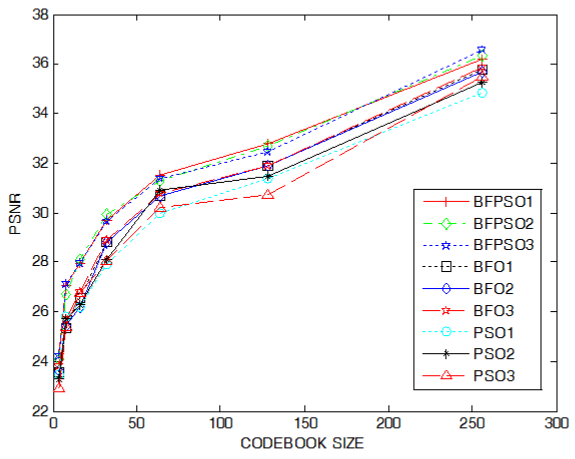


FIGURE 4. PSNR in dB versus the size of the codebook for "pepper".

6. **Acknowledgment.** This research was partly supported by the National Science Council of the Republic of China under contract NSC-93-2218-E-507-001 and NSC-100-2221-E-507-004.

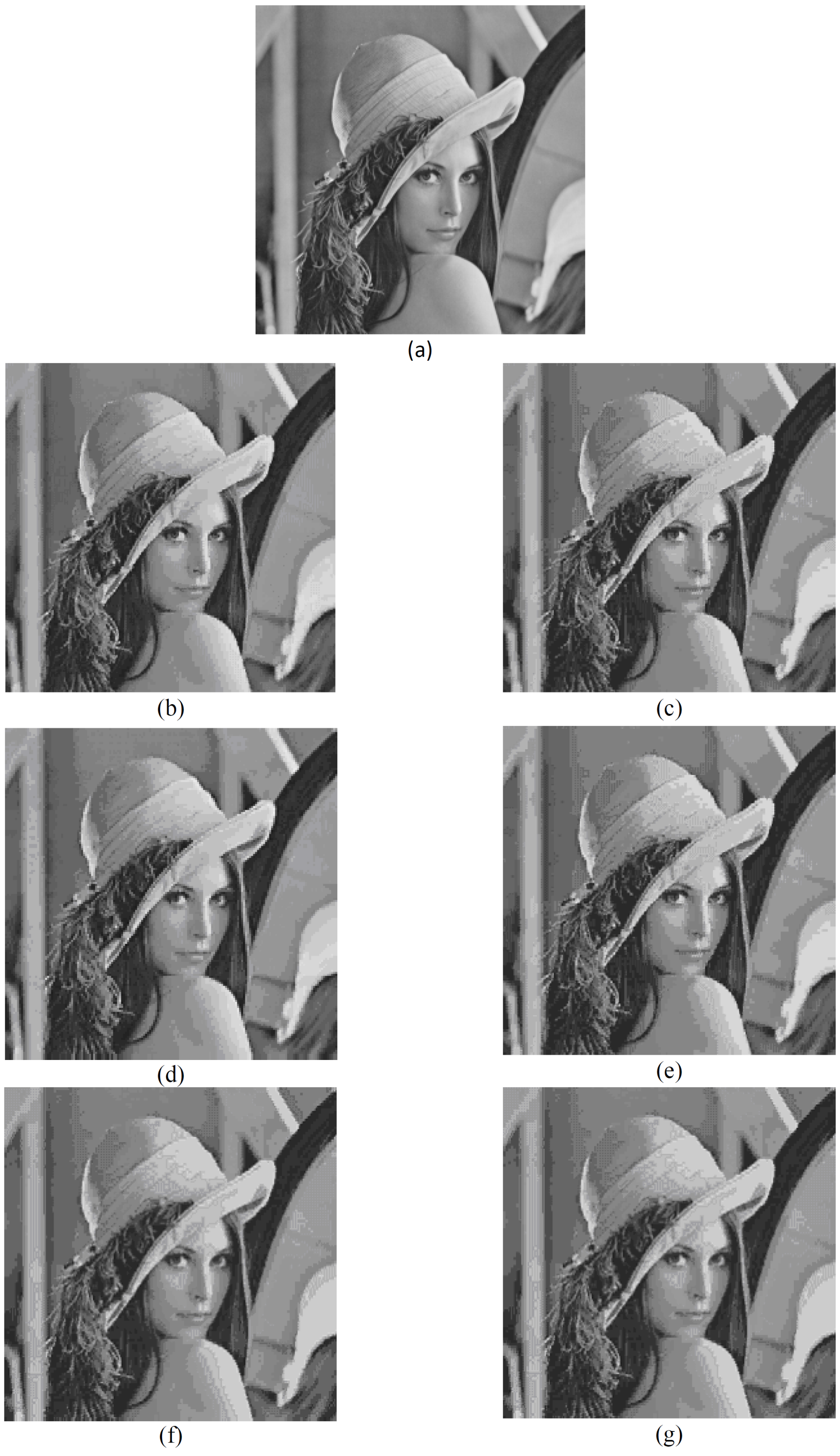
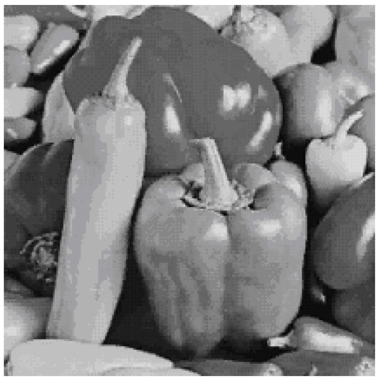
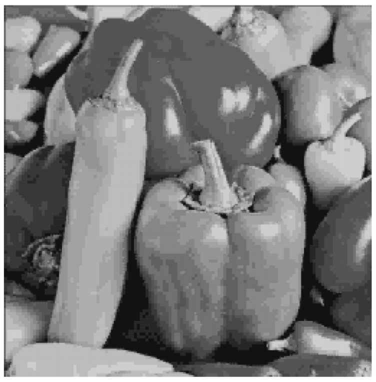
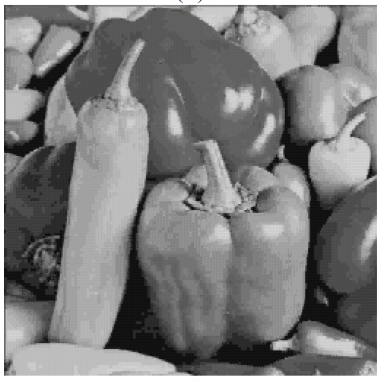


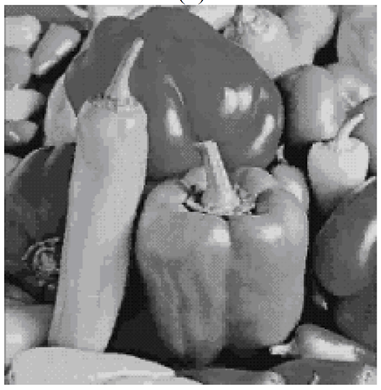
Figure 5. Comparison results of the training images (a) Original Lena image (b) BFPSO reconstruction image M=128. (c) BFO reconstructed image in M=128. (d) PSO reconstructed image in M=128. (e)BFPSO reconstruction image in M=64. (f) BFO reconstructed image in M=64. (g) PSO reconstructed image in M=64

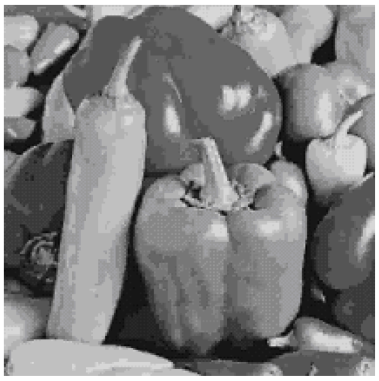
(a)


(b)


(c)


(d)


(e)


(f)

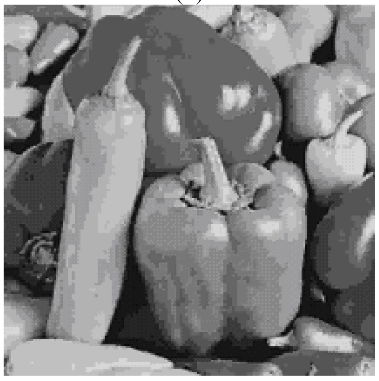

(g)

FIGURE 6. Comparison results of the training images, (a) Original Peppers image (b) BFPSO reconstruction image M=128. (c) BFO reconstructed image in M=128. (d) PSO reconstructed image in M=128. (e)BFPSO reconstruction image in M=64. (f) BFO reconstructed image in M=64. (g) PSO reconstructed image in M=64.

Table 1. Performance (PSNR) comparisons of BFPSO, BFO and PSO. (codebook sizes=4-256).

		PSNR (dB)						
		M=4	M=8	M=16	M=32	M=64	M=128	M=256
Lena	BFPSO1	25.1278	28.6772	29.9667	31.8082	32.8171	34.6736	36.7636
	BFPSO2	25.1558	28.7111	29.7890	31.7966	32.8827	34.7582	36.7999
	BFPSO3	25.1858	28.8652	29.9293	31.9298	32.8982	34.7081	36.8985
	BFO1	24.7591	27.2317	28.7358	30.9031	31.6652	33.7891	35.5891
	BFO2	25.0125	27.5326	28.6532	30.7431	31.5891	33.4531	35.7821
	BFO2	24.8925	27.3219	28.7341	30.7582	31.8741	33.6791	35.8912
	PSO1	24.5743	27.6632	28.5034	30.7186	31.7560	33.1356	35.3152
	PSO2	24.4219	26.2834	28.7666	30.4075	31.9876	33.2536	35.6289
	PSO3	24.6917	26.8508	27.8919	30.1687	31.0968	33.0495	35.4331
Peppers	FPSOVQ1	24.3101	27.0548	27.9566	29.7152	31.5165	32.7846	36.1881
	FPSOVQ2	24.1040	26.7158	28.1078	29.9262	31.2796	32.6928	36.3401
	FPSOVQ3	24.2119	27.1208	27.9797	29.6373	31.3762	32.4581	36.5651
	BFO1	23.5671	25.3416	26.5812	28.7891	30.6721	31.8915	35.7812
	BFO2	23.6178	25.4531	26.1758	28.7914	30.6731	31.9123	35.6816
	BFO2	23.7812	25.7189	26.7891	28.8912	30.7831	31.8759	35.8901
	PSO1	23.4530	25.8228	26.1730	27.9213	29.9923	31.3953	34.8264
	PSO2	23.2803	25.7112	26.2970	28.1265	30.8934	31.4531	35.2823
	PSO3	22.9181	25.4152	26.7489	28.0495	30.2175	30.7114	35.5142

REFERENCES

[1] C. H. Lin and C. Y. Yamg, Multipurpose watermarking based on blind vector quantization (BVQ), *Journal of Information Hiding and Multimedia Signal Processing*, vol. 2, pp. 236-246, 2011.

[2] H. M. Feng, C. Y. Chen and F. Ye, Evolutionary fuzzy particle swarm optimization vector quantization learning scheme in image compression, *Journal of Expert Systems with Applications*, vol. 32, pp. 213-222, 2007.

[3] H. M. Feng and J. H. Horng, VQ-Based fuzzy compression systems designs through bacterial foraging particle swarm optimization algorithm, *Proc. of the 5th International Conference on Genetic and Evolutionary Computing*, pp. 256-259, 2011.

[4] H. M. Feng, Autonomous rule-generated fuzzy systems designs through bacterial foraging particle swarm optimization algorithm, *Proc. of International Conference on Electric and Elctronics*, vol. 98, pp. 19-28, 2011.

[5] J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, New York, USA, Plenum Press, 1981.

[6] J. Kennedy, and R. C. Eberhart, Particle Swarm Optimization, *Proc. of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942-1948, 1995.

[7] K. L. Oehler and R. M. Gray, Combining image compression and classification using vector quantization, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 17, pp. 461-473, 1995.

[8] K. M. Passino, Biomimicry of bacterial foraging for distributed optimization and control, *Journal of IEEE Control Systems*, vol. 22, pp. 52-67, 2002.

[9] L. A. Zadeh, Fuzzy sets *Journal of Information and Control*, vol. 8, pp. 338-353, 1965.

[10] N. B. Karayiannis and P. I. Pai, Fuzzy vector quantization algorithms and their application in image compression, *IEEE Trans. Image Processing*, vol. 4, pp. 1193-1201, 1995.

[11] P. Puranik, P. Bajaj, A. Abraham, P. Palsodkar and A. Deshmukh, Human perception-based color image segmentation using comprehensive learning particle swarm optimization, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 2, pp. 227-235, 2011.

[12] T. Hofmann and J. M. Buhmann, Competitive learning algorithms for robust vector quantization, *IEEE Trans. Signal Processing*, vol. 46, pp. 1665-1675, 1998.

[13] W. M. Korani, H. T. Dorrah and H. M. Emara, Bacterial foraging oriented by Particle Swarm Optimization strategy for PID tuning, *Proc. of IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp. 445-450, 2009.

[14] W. Pedrycz and K. Hirota, Fuzzy vector quantization with the particle swarm optimization: A study in fuzzy granulation-degranulation information processing, *Journal of Signal Processing*, vol. 87, pp. 2061-2074, 2007.

[15] W. Xu, A. K. Nandi and J. Zhang, Novel fuzzy reinforced learning vector quantisation algorithm and its application in image compression, *Proc. of IEE Vision, Image and Signal Processing*, vol. 150, pp. 292-298, 2003.

[16] X. Kong and R. Wang, G. Li, Fuzzy clustering algorithms based on resolution and their application in image compression, *Journal of Pattern Recognition*, vol. 35, pp. 2439-2444, 2004.

[17] X. L. Liu, R. J. Li and P. Yang, A bacterial foraging global optimization algorithm based on the particle swarm optimization, *IEEE International Conference on Intelligent Computing and Intelligent Systems*, pp. 22-27, 2010.

[18] Y. Linde, A. Buzo and R. Gray, An algorithm for vector quantization design, *IEEE Trans. Communications*, vol. 28, pp. 84-94, 1980.

[19] Z. M. Lu and Y. N. Li, Image compression based on mean value predictive vector quantization, *Journal of Information Hiding and Multimedia Signal Processing*, vol. 1, pp. 172-178, 2010.