Color Quantization Using Modified Artificial Fish Swarm Algorithm

Danial Yazdani[1], Hadi Nabizadeh[2], Elyas Mohamadzadeh Kosari[3], and Adel Nadjaran Toosi[4]

[1,2] Department of Electronic, Computer and Information Technology, Azad University of Qazvin, Iran, {d_yazdani, h_nabizadeh}@qiau.ac.ir

[3]Department of Computer Engineering, Faculty of Engineering, Ferdowsi University of Mashhad, Mashhad, Iran, elyas.kosari@stu-mail.um.ac.ir

[4]Department of Computer Science and Software Engineering, University of Melbourne, Melbourne, Australia, adeln@csse.unimelb.edu.au

Abstract. Color quantization (CQ) is one of the most important techniques in image compression and processing. Most of quantization methods are based on clustering algorithms. Data clustering is an unsupervised classification technique and belongs to NP-hard problems. One of the methods for solving NP-hard problems is applying swarm intelligence algorithms. Artificial fish swarm algorithm (AFSA) fits in the swarm intelligence algorithms. In this paper, a modified AFSA is proposed for performing CQ. In the proposed algorithm, to improve the AFSA's efficiency and remove its weaknesses, some modifications are done on behaviors, parameters and the algorithm procedure. The proposed algorithm along with other multiple known algorithms has been used on four well known images for doing CQ. Experimental results comparison shows that the proposed algorithm has acceptable efficiency.

Keywords: Color quantization, compression, artificial fish swarm algorithm, data clustering.

1 Introduction

One of the available challenges in image processing is high color variety in pixels. Therefore, usually a decreasing technique of color variety is used as a preprocessing for different works in graphic and image processing applications. By decreasing the number of colors, it can decrease image file size to conserve storage space, reduce time for transmission, and reduce computation. Color Quantization (CQ) is one of the most famous techniques of decreasing the numbers of colors which is applied in image compression [1], graphic [2] and image processing [3,4].

CQ process is done in two steps [5]. In the first step, a codebook is constructed. In this step, it has to be determined how many colors have to be decreased at first. In fact, the number of considered colors is the number of codewords in the codebook. Each codeword represents a color and its index in the codebook that each of these codewords is representative of multiple colors on the original image. After constructing the codebook by means of a CQ method, the image is encoded by that and every pixel just would possess index number of its representative color in the codebook. On the second step, each image is decoded by its corresponding codebook. From the view of compression, CQ is taken into account as a lossy compression method in which some information are lost. Indeed, after performing CQ on an image, most of pixels cannot have their primary colors anymore [5].

Natural images often have many colors and determining a proper codebook is a challenging problem for decreasing the colors. Generally, CQ techniques could be categorized into two classes: first category consists of those methods independent from image, which specify a comprehensive codebook regardless of any specific image. In these methods, first, a fixed codebook is produced by using a training set, and then all images are encoded and decoded by means of this codebook. Second category contains dependent techniques to image, in which for every image a codebook is built. Usually, in these methods, every codebook is built based on color distribution in a specific image. Thereafter, to use it for decoding the image, the codebook with encoded image is transferred. Dependent image techniques are slower than independent image techniques, but obtained results from the former have higher quality than the latter [6].

In CQ, the main goal is to obtain an appropriate codebook. If the codebook is not proper, the resulted image from CQ has much disharmony with the original image and distortion increases between the original image and decoded one. One of the applied approaches for producing codebook based on the color distribution is using clustering algorithms like k-means [6] and FCM [7]. Clustering is an unsupervised classification technique in which datasets that are usually data vectors in multi dimensional space. Data vectors are divided into some clusters based on a similarity criterion. After performing a clustering algorithm, each of data vectors of dataset is assigned to one of clusters. Clustering process is done with respect to a specific similarity criterion such that assigned data to a cluster are more similar than other data in other clusters. The way of using clustering algorithms in CQ is such that, first, colors histogram is produced for the original image and after that, clustering according to the color distribution among pixels is done. The number of cluster centers in clustering algorithms is determined equal to the number of decreased colors in the codebook. In clustering process, cluster centers contain smaller set of colors. Other colors with respect to difference between their color numbers and the numbers of cluster center colors become a member of one of the clusters. That is, each of colors becomes a member of a cluster that its center color is more similar than other cluster centers. One of the applied methods for clustering is use of the swarm intelligence algorithms such as particle swarm optimization [8], and artificial fish swarm algorithm [9].

Artificial fish swarm algorithm (AFSA) was presented by Li Xiao Lei in 2002 [10]. This algorithm is a technique based on swarm behaviors that was inspired from social behaviors of fish swarm in nature. AFSA works based on population, random search, and behaviorism. This algorithm has been used in optimization applications, such as clustering [11, 12], machine learning [13, 14], PID control [15], data mining [16], image segmentation [17].

In this paper, a modified AFSA is proposed as a color quantizer. In the modified AFSA, we try to remove weaknesses of standard AFSA to increase the algorithm efficiency. Then, this algorithm is configured to perform CQ. In order to comparison, the proposed algorithm along with three other clustering algorithms is used for performing CQ on 4 well-known images. Comparing qualitative efficiency of the algorithms confirms the competence of the proposed algorithm. The remainder of the paper is organized as follows: in section 2, standard AFSA will be described and in section 3, the proposed algorithm will be presented. Section 4 studies the experiments and analyzes the results. Final section concludes the paper and outlines possibilities of the future works.

2 Artificial Fish Swarm Algorithm

In the underwater world, fish can find areas that have more foods rather than their current area, which is done with individual or swarm search by fishes. According to this characteristic, artificial fish (AF) model is represented by prey, free_move, swarm and follow behaviors. AFs search the problem space by those behaviors. The environment, which AF lives in, substantially is solution space and other AF's domain. Food consistence degree in water area is AFSA objective function. Finally, AFs reach to a point, which its food consistence degree is maxima (global optimum).

In AFSA, AF perceives external concepts with sense of sight. Current position of AF is shown by vector $X=(x_1, x_2, \ldots, x_n)$. The *visual* is equal to length of sight field of AF in each dimension and X_v is a position in *visual* where the AF wants to go. Then if X_v has better food consistence than current position of AF, it goes one step toward X_v which causes change in AF position from X to X_{next}, but if the current position of AF is better than X_v, it continues searching in its *visual* area. *Food consistence* in position X is fitness value of this position and is shown with $f(X)$. The step is equal to maximum length of the movement. The distance between two AFs which are in X_i and X_j positions is shown by $Dis_{ij}=\| X_i-X_j\|$ (Euclidean distance). AF model consists of two parts of variables and functions. Variables include X (current AF position), *step* (maximum length step), *visual* (length of sight field), *try-number* (the maximum test interactions and tries) and crowd factor δ ($0<\delta<1$). Also functions consist of prey behavior, free move behavior, swarm behavior and follow behavior. In each step of optimization process, AF looks for locations with better fitness values in problem search space by performing these four behaviors based on algorithm procedure [10, 14, 15].

3 Proposed Algorithm

In this section, a modified artificial fish swarm algorithm called MAFSA is presented. Then, MAFSA is configured as a color quantizer. Generally, modifications which are imposed on standard AFSA structure include: removing two parameters step and crowd_factor, adding contraction coefficient parameter to the algorithm, removing blackboard, changing visual parameter value during algorithm execution, changing in follow and prey behaviors, removing swarm behavior and changing in the procedure of algorithm execution. In the following, modified AFSA algorithm is described. First, modified AFSA behaviors are explained:

3.1 Prey behavior

This behavior is an individual behavior that each AF performs independently and performs a local search around itself. Every AF by performing this behavior attempts try-number times to move to a new position with better fitness. Here, it is supposed that AF *i* is in position $\vec{X}_i$ and wants to perform prey behavior. In prey behavior, following steps are done:

1) AF *i* considers a goal position $\vec{X}_T$ in its *visual* by means of Eq. (1), then evaluates its fitness. *d* shows dimension number and Rand generates a random number by uniform distribution in [-1, 1].

$$X_{T,d} = X_{i,d} + Visual \times Rand_d(-1,1) \quad \textbf{(1)}$$

2) If fitness value of position $\vec{X}_T$ is better than fitness value of the current position of AF *i*, position of AF *i* is updated by Eq. (2).

$$\vec{X}_i(t+1) = \vec{X}_i(t) + \left(\vec{X}_T - \vec{X}_i(t)\right) \times Rand(0,1) \quad \textbf{(2)}$$

Steps 1 and 2 are repeated *try-number* times. By executing above steps, an AF can update its position at most try-number times.
AF moves as a random percentage of the distance between its current position and goal position at each movement. Also, it is possible that none of its attempts for finding better positions is efficacious. If AF *i* couldn't move toward better positions by performing two mentioned steps (*try_number* times), it moves with a random step in its *visual* by means of Eq. (3):

$$X_{i,d}(t+1) = X_{i,d}(t) + Visual \times Rand_d(-1,1) \quad \textbf{(3)}$$

In MAFSA, by executing Eq. (3) on AF, it is attempted to preserve swarm diversity, but it wouldn't be used for the best AF of swarm because this behavior may result in worse position for an AF. Therefore, the best AF of swarm wouldn't lose its position even when it doesn't find better position in its neighborhood and just displaces when it could find better position in its visual. Thereafter, in this condition, the best found position during previous iterations by swarm is the best AF's position since at each iteration, the best AF of swarm changes its position when it moves toward a better position. Consequently, MAFSA doesn't require to blackboard anymore.

3.2 Follow behavior

As it was mentioned in subsection 3-1, the best AF of swarm locates in the best found position so far by swarm. In follow behavior, each of AF moves one step toward the best AF of swarm by Eq. (4):

$$\vec{X}_i(t+1) = \vec{X}_i(t) + \frac{\vec{X}_{Best} - \vec{X}_i(t)}{Dis_{i,Best}} \times \left[Visual \times Rand(0,1)\right] \quad \textbf{(4)}$$

Where $\vec{X}_i$ is position vector of AF *i* which performs follow behavior and $\vec{X}_{Best}$ is the position vector of the best AF of swarm. Hence, AF *i* moves as a random percentage of *visual* in each dimension toward the best AF of swarm. Indeed, after that an AF finds more food, other members follow it to reach more food, too. Performing follow behavior of the best AF of swarm causes increase in convergence rate of swarm and helps to keep integrity of AF of a swarm. This behavior is a group behavior and interactions between members of swarm are done globally among them. Thus, this behavior can also perform the duty of swarm move (keeping swarm integrity) in standard AFSA since it can keep AF in a swarm and prevent from swarm splitting in problem space. As a result, swarm behavior is eliminated in MAFSA. To execute follow behavior in standard AFSA, at each iteration of algorithm execution, it has to calculate Euclidean distance between all AF with each other (for detecting neighbors of each AF), which is of heavy computational load. But follow behavior in MAFSA has lower computational load while it is very effective in increasing the convergence rate of the algorithm.

3.3 Modified AFSA Procedure

In MAFSA, each of prey and follow behaviors are done for each of AF at every iteration. In MAFSA, first all AF perform prey behavior and their position are updated based on prey behavior execution procedure. Then, follow behavior is performed and all members except the best AF of swarm move to a new position in direction of moving toward the best found position by swarm. At the end of each of iteration of MAFSA algorithm, *visual* value is updated for AF to make a balance between global search and local search abilities [13, 14]. To reach this goal, *visual* has to be large at first, such that AF converge to their goals fast and perform global search well. Simultaneously with swarm convergence toward goal, *visual* decreases gradually until AF with small *visual* could get better results by doing an acceptable local search around goal.

For this purpose, *visual* is multiplied by a positive number less than one at each of iterations, which this number can be determined with different approaches [14]. In this paper, to decrease *visual*, a random number generator with uniform distribution is applied in the considered interval that is given in Eq. (5):

$$Visual(t+1) = Visual(t) \times \left(L_{Low} + \left(Rand \times \left(L_{High} - L_{Low}\right)\right)\right) \tag{5}$$

In Eq. (5), at each of iterations, *visual* is obtained randomly with respect to this parameter in previous iteration. L_{low} and L_{high} are lower bound and upper bound of change percentage of *visual* to previous iteration respectively and Rand is random number generator with uniform distribution in interval [0, 1]. Therefore, *visual* is a random percentage of its value in previous iteration between L_{low} and L_{high}. For this reason, L_{high} should be considered a number less than one. Pseudo code of MAFSA is represented in figure 1.

```
MAFSA:
for each Artificial Fish i ∊ [1 .. N]
    initialize xi
endfor
repeat:
    for each Artificial Fish i ∊ [1 .. N]
       flag[i]=0;
       for counter=1 to try_number
          Obtain X_T with Eq.  1 and Calculate f(X_T)
          if  f(X_T) ≤  f(X_i) then
             apply Eq.  2
             flag[i]=1;
          endif
        endfor
        if flag[i]==0 then
           apply Eq.  3
        endif
    endfor
   for each Artificial Fish i ∊ [1 .. N]
       apply Eq.  4
   endfor
   Update Visual according Eq.  5
until stopping criterion is met
```

Fig. 1. MAFSA pseudo code

3.4 MAFSA Configuration for CQ

In this section, MAFSA configuration is discussed as a color quantizer. Application of images is usually for observing by an individual. The eye is very good at interpolation, that is, the eye can tolerate some distortion. The eye has more acuity for luminance (gray scale) than chrominance (color). This is why we will concentrate on compressing gray scale (8 bits per pixel) image.

As mentioned before, the goal of this paper is to solve CQ problem as a clustering problem. First, a dataset has to be determined that clustering has to be done on it. In this problem, dataset consists of all pixels' values. In gray scale images, every pixel has an 8 bit color characteristic, so data are one dimensional. Then, it has to be specified a fitness function for clustering which MAFSA should optimize. In this paper, to find optimal values of cluster centers which their number has been predetermined, one of the most known clustering criteria called sum of intra cluster distances is used [18]. Eq. (6) is a function which calculates sum of intra cluster distances that according to it, the best clustering is the one when this function's value is minimum.

$$J(C_1, C_2, ..., C_K) = \sum_{i=1}^{K} \left(\sum_{X_j \in C_i} \| Z_i - X_j \| \right) \tag{6}$$

This function is used as a fitness function for MAFSA algorithm and is considered as a minimizing problem. In Eq. (6), the Euclidean distance between each data vector in a cluster and the centroid of that cluster is calculated and summed up. Here, we have K clusters C_i ($1 \leq i \leq K$) that each of N data vectors X_j ($1 \leq j \leq N$) are clustered on the basis of distance from each of these cluster centers Z_i ($1 \le i \le K$). Data vectors belong to a cluster that their Euclidean distance from its cluster center is less than their Euclidean distance from other cluster centers. Therefore, MAFSA goal is to determine cluster centers which minimize Eq. (6), and consequently optimal cluster centers are determined. In fact, according to clustering conditions of CQ, Eq. (6) shows the sum of differences between original image's pixel color numbers and decoded image's. Codebook is determined by using final result of clustering. Indeed, the codebook contains cluster centers and their indices. Each cluster center is a one dimensional vector that consists of gray scale color values. Hence, after completing the codebook, every pixel of the original image is transformed into cluster center index in the codebook that it belongs to (encoding). To represent the image again, each pixel takes its corresponding cluster center color values with respect to its encoded value (decoding).

Since data and cluster centers are one dimensional and there are K clusters (the number of codewords in the codebook), so every AF has to represent K cluster centers. As a result, each AF is K dimensional or has K components in its vector. As a matter of fact, each of components includes one of colors which are supposed to be considered as the replacement of some more similar colors to it.

In MAFSA, first, AFs are initialized randomly in the problem space. Therefore, every AF consists of K initial random cluster centers which displace these cluster centers in the problem space by means of MAFSA behaviors and their goal is to determine cluster centers in a way that Eq. (6) to be minimizes as a fitness function. At last, the codebook would be the same as obtained cluster centers from MAFSA.

4 Experiments

Experiments are done on 4 well-known images which are mostly used for measuring the efficiency of CQ algorithms. These images are *Barbara*, *Boat*, *Lenna* and *Pepper* that their size is 512*512 pixels. Figure 2 shows applied images in this paper.

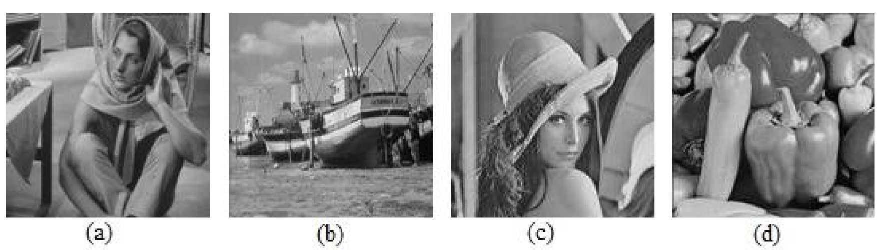

Fig. 2. Applied images in this paper.

The most important measurement criteria for CQ algorithms efficiency include mean squared error (*MSE*) and peak signal to noise ratio (*PSNR*) [5,6]. *MSE* is usually used for assessing distortion between the original image and resulted image from CQ. Let the original image x have n pixels. *MSE* is computed by Eq. (7):

$$MSE = \frac{1}{n} \sum_{i=1}^{n} \left(x_i - \hat{x}_i \right)^2 \quad \textbf{(7)}$$

Where, $\hat{x}$ is the obtained image after performing CQ. *MSE* represents the average distortion and lesser value of it shows better efficiency of CQ algorithm. *PSNR* is a standard way for evaluating fidelity between the original image and the obtained image from CQ. *PSNR* is calculated by Eq. (8):

$$PSNR = 10\log_{10}\left(\frac{m^2}{MSE} \right) \quad \textbf{(8)}$$

Where, *m* is the largest amount which a pixel can take that is 255 in gray scale images. *PSNR* is measured in decibels (*dB*) and the larger value of it shows better efficiency of CQ method. The proposed algorithm along with standard AFSA, PSO and k-means is used for performing CQ on 4 mentioned images. PSO parameters are adjusted with respect to [18] and Forgy initializing method is used for k-means [6]. Population size is considered 5 times the number of problem space dimensions for standard AFSA and MAFSA [18, 9]. That is, population size is 5 times the number of codebooks' colors. Based on multiple experiments which have been done, visual, try-number, L_{high} and L_{low} are 10, 10, 1 and 0.95, respectively. Standard AFSA's parameters are adjusted according to [9]. Experiments are repeated 50 times and average of obtained *PSNR* and *MSE* from 4 algorithms on 4 images are represented in table 1. In this table, each image has been compressed with rates 8:3, 8:4 and 8:5 which their colors have been decreased to 8, 16 and 32, respectively. The best result is shown by bold face for each case. As it is observed, MAFSA has achieved better results in all cases. MAFSA has achieved better results than standard AFSA because of not having the weaknesses of standard AFSA specially imbalance between global search and local search [14]. In fact, AF perform global and local search well in MAFSA and generate a codebook by decreasing the sum of intra cluster distances which decreases distortion in decoded image. Therefore, obtained images from the proposed algorithm would have more fidelity with the original image.

According to results of table 1, generally, standard AFSA has less efficiency than PSO. But MAFSA has achieved better efficiency than PSO by improving different parts of standard AFSA. Figure 3 shows Lenna and Peppers images whose colors have been decreased to 8 colors and have been compacted by rate 8:3.

Fig. 3. Two decoded images with 8 colors.

On the whole, experimental results show that compressed images by means of generated codebook by the proposed algorithm are of higher quality than other algorithms.

Table 1. MSE and PSNR comparison of the quantization methods.

Image	Compression Ratio	Criteria	Std-AFSA	K-means	PSO	MAFSA
Lenna	8:3	MSE	29.81	28.16	26.04	**22.48**
		PSNR	33.38	33.69	33.97	**34.61**
	8:4	MSE	8.93	8.29	8.69	**6.27**
		PSNR	38.62	39.03	38.76	**40.16**
	8:5	MSE	3.71	2.78	3.34	**1.71**
		PSNR	42.43	43.70	42.93	**45.80**
Barbara	8:3	MSE	25.48	24.94	25.81	**22.78**
		PSNR	34.07	34.19	34.02	**34.55**
	8:4	MSE	9.99	8.37	9.87	**6.48**
		PSNR	38.14	38.95	38.20	**40.01**
	8:5	MSE	3.83	3.50	3.01	**1.61**
		PSNR	42.29	42.97	43.39	**46.05**
Boat	8:3	MSE	26.79	23.99	24.91	**23.39**
		PSNR	33.85	34.32	34.22	**34.44**
	8:4	MSE	10.14	9.11	10.61	**6.78**
		PSNR	38.15	38.64	37.93	**39.82**
	8:5	MSE	4.18	2.62	3.90	**2.06**
		PSNR	42.17	44.05	42.28	**44.97**
Pepper	8:3	MSE	29.58	30.04	32.55	**28.90**
		PSNR	33.42	33.38	33.01	**33.52**
	8:4	MSE	10.43	9.84	10.93	**7.10**
		PSNR	37.95	38.22	37.77	**39.61**
	8:5	MSE	4.56	2.83	3.11	**1.86**
		PSNR	41.57	43.66	43.21	**45.43**

5 Conclusion

In this paper, a modified artificial fish swarm algorithm was proposed. In the proposed algorithm, it has been attempted to remove standard AFSA's weaknesses and algorithm to be able to reach acceptable and good results. The proposed algorithm is utilized in CQ application and its efficiency is compared qualitatively with efficiency of standard AFSA, PSO and k-means. In this study, images are compressed only with respect to the number of their colors. Experimental results show that obtained images from the proposed algorithm are of higher quality than obtained results from other tested algorithms. However, the proposed algorithm has more complexity than other tested algorithms. Reducing complexity is issue that merits further research.

References

1. Yang. C.K, Tsai. W.H.: Color Image Compression Using Quantization, thresholding, and Edge Detection Techniques all Based on the Moment-Preserving Principle. In: Pattern Recognition Letters 19, pp. 205--215, (1998)
2. Wang. S, Cai. K, Lu. J,Liu. X, Wu. E. Real-time coherent stylization for augmented reality. In:The Visual Computer 26, pp. 445-455, (2010)
3. Deng. Y, Manjunath. B.: Unsupervised Segmentation of Color–Texture Regions in Images and Video. In: IEEE Transactions on Pattern Analysis and Machine Intelligence 23, pp. 800--810, (2001)
4. Sertel. O, Kong. J, Catalyurek. U.V., Lozanski. G, Saltz. J.H, Gurcan. M.N.: Histopathological Image Analysis Using Model-Based Intermediate Representations and Color Texture: Follicular Lymphoma Grading. In: Journal of Signal Processing Systems 55, pp. 169--183, (2009)
5. Sayood. K.:Introduction to Data Compression. In: Morgan Kaufmann, 3th Edition, (2006)
6. Celebi. M.E.: Improving the Performance of K-means for Color Quantization. In: Journal of Image and Vision Computing 29, pp. 26--271, (2011)
7. Schaefer. G, Zhou. H.: Fuzzy Clustering for Color Reduction in Images. In: Telecommunication Systems 40, pp. 17--25, (2009)
8. Tsai, C.Y., Kao, I.W.: Particle Swarm Optimization with Selective Particle Regeneration for Data Clustering. In: Journal of Expert Systems with Applications 38, pp. 6565--6576. (2011)
9. Yazdani, D., Golyari, S., Meybodi, M. R.: A New Hybrid Approach for Data Clustering. In: 5th International Symposium on Telecommunication (IST), pp. 932--937, Tehran (2010)
10. Li, L.X., Shao, Z.J., Qian, J.X.: An Optimizing Method Based on Autonomous Animate: Fish Swarm Algorithm. In: Proceeding of System Engineering Theory and Practice, pp. 32--38. (2002)
11. Hi, S., Belacel, N., Hamam, H., Bouslimani, Y.: Fuzzy Clustering with Improved Artificial Fish Swarm Algorithm. In: International Joint Conference on Computational Sciences and Optimization 09, pp. 317--321, Hainan (2009).
12. Xiao, L.: A Clustering Algorithm Based on Artificial Fish school. In: 2nd International Conference on Computer Engineering and Technology, pp. 766--769, Chengdu (2010)
13. Yazdani, D., Golyari, S., Meybodi, M.R.: A New Hybrid Algorithm for Optimization Based on Artificial Fish Swarm Algorithm and Cellular Learning Automata. In: 5th International Symposium on Telecommunication (IST), pp. 932--937, Tehran (2010)
14. Yazdani, D., Nadjaran Toosi, A., Meybodi, M.R.: Fuzzy Adaptive Artificial Fish Swarm Algorithm. In: 23th Australian Conference on Artificial Intelligent, Adelaide (2010)
15. Luo, Y., Zhang, J., Li, X.: The Optimization of PID Controller Parameters Based on Artificial Fish Swarm Algorithm. In: IEEE International Conference on Automation and Logistics, pp. 1058--1062, Jinan (2007)
16. Zhang, M., Shao, C., Li, M., Sun, J.: Mining Classification Rule with Artificial Fish Swarm. In: 6th World Congress on Intelligent Control and Automation, pp. 5877--5881, Dalian (2006)
17. Li, C.X., Ying, Z., JunTao, S., Qing, S.J.: Method of Image Segmentation Based on Fuzzy C-means Clustering Algorithm and Artificial Fish Swarm Algorithm. In: International Conference on Intelligent Computing and Integrated Systems (ICISS), Guilin (2010)
18. Kao, Y.T., Zahara, E., Kao, I.W.: A Hybridized Approach to Data Clustering. In: Journal on Expert System with Applications 34, pp.1754-1762 (2008)