

Automatic Clustering Using an Improved Particle Swarm Optimization

R. J. Kuo and Ferani E. Zulvia

National Taiwan University of Science and Technology, Taipei, Taiwan

Email: rjkuo@mail.ntust.edu.tw, ve_zulvia@yahoo.co.id

Abstract—Unsupervised data clustering is an important analysis in data mining. Many clustering algorithms have been proposed, yet most of them require predefined number of clusters. Unfortunately, unavailable information regarding number of clusters is commonly happened in real-world problems. Thus, this paper intends to overcome this problem by proposing an algorithm for automatic clustering. The proposed algorithm is developed based on a population-based heuristic method named particle swarm optimization (PSO). It overcomes two main issues in automatic clustering, namely determining number of clusters and cluster centroid. In the automatic clustering using PSO (ACPSO), the exploration is conducted by particles comprising of two sections. Herein, time-varying tuning parameter is applied. Furthermore, sigmoid function is employed to handle infeasible solution. In addition, K-means is applied to adjust the cluster centroids. Method validation using four benchmark datasets reveals that TPSO outperforms other two previous methods namely DCPSO, DCPG, and DCGA. Overall, ACPSO has better accuracy and consistency.

Index Terms—automatic clustering, Particle swarm optimization, K-means.

I. INTRODUCTION

Clustering as one of important tasks in unsupervised data mining works in grouping instances into several clusters based on their similarity[1]. In recent years, clustering has been applied in wide-ranging areas such as engineering, computer science, biology and medical, social sciences and economics [2]. Due to its important role, many clustering methods have been proposed. These methods can be roughly divided into two categories, namely partition and hierarchical algorithm [3]. Partition algorithm divides a dataset into several clusters such that each instance is assigned into a cluster. There are many methods grouped as partition algorithm. Among them, K-Means algorithm is the most well-known method. It is a prototype-based clustering method which defines centroids as its cluster prototype. K-means is a very old clustering method, yet it still applicable due to its easy implementation and fast convergence [4]. Another type of clustering method is hierarchical clustering. Different with partition algorithm which divides data in one level, this algorithm constructs cluster step-by-step based on certain

proximity measurement. There are two basic approaches applied in hierarchical algorithms: agglomerative and divisive approaches. Agglomerative starts with defining each instance as a cluster. It then iteratively merges the closest two clusters until all instances grouped into one cluster. On the contrary, divisive hierarchical algorithm works by splitting a cluster into two smaller clusters until each cluster consists of an instance.

Although these methods are still widely applied, most of clustering methods require predefined number of clusters [5]. For any recognizable problems which have an exact number of clusters, these aforementioned methods are still applicable. However, lack information regarding number of cluster is commonly exists [6]in real-world problems. An appropriate number of clusters is an important parameter in clustering since it may affect the clustering results[7]. Thus, many researches work in developing clustering algorithm without requiring predefined number of clusters. This direction is known as automatic clustering or dynamic clustering. Clustering methods applied for automatic clustering basically works in determining the number of clusters based on some cluster validity measures [8, 9]. Recently, many heuristics methods such as genetic algorithm (GA) [5], [6], [10], [11], differential evolution (DE) [2], and tabu search [4] are developed to solve automatic clustering. Among these methods, PSO is a very promising algorithm especially for continuous problem such as clustering [12]. This study aims to develop PSO so that it can be applied for automatic clustering. The ACPSO algorithm proposed in this paper applies continuous PSO in for determining number of clusters and exploring cluster centroids. In this algorithm, each particle gives a result comprising of number of clusters and predicted clusters' centroids. It is then continued by applying K-means to adjust the centroid for each cluster.

The rest of this paper is organized as follows. Section II presents necessary background related to this study such as clustering algorithm and PSO. In Section III, the proposed PSO algorithm is discussed while Section IV gives experimental results involving several benchmark datasets for clustering. Finally, section V provides the concluding remarks.

II. BACKGROUND

This section briefly reviews literature study considered in this paper.

A. Clustering Algorithm

Data clustering is a data analysis method which mining important information from a dataset by grouping data into several clusters. In clustering, similar data points are grouped into the same cluster while different data points are put into different clusters. Generally, data clustering method is grouped two different categories: hierarchical and partition method[3]. Hierarchical clustering works by either merging two clusters or splitting a cluster. Hierarchical algorithm which performs clusters merging is called agglomerative hierarchical algorithm. This method starts with treat each individual data point as a cluster. It then iteratively merges two nearest clusters until only one cluster left. Oppositely, divisive hierarchical algorithm splits a cluster until each cluster is consists of only one data point. Another direction in clustering is partition algorithm. Partition clustering divides dataset into several clusters in one level. Some methods grouped as partition clustering are K-means, fuzzy C-Means, DBSCAN, EM, etc[3].The above mentioned various unsupervised clustering method requires the user to predetermine number of clusters. However, for some unknown dataset, information regarding appropriate number of clusters is very limited. As results, clustering algorithm for automatic clustering becomes a new promising research direction in clustering.

In terms of clustering's objectives, the aims of data clustering are minimize similarity among instance in one cluster and maximize dissimilarity between two clusters. Recently, many researches proposed many different ways for evaluating clusters. For instances DB index which minimize average similarity [13], PBM index which tries to find small number of cluster which are well separated [9], $CS(k)$ [8], and VI [14]. Basically, these measurements are ratio between the two clustering's objectives.

B. Particle Swarm Optimization

PSO is a population-based search algorithm which simulates bird behavior in finding food. In PSO, birds represented as particle are employed to perform solution exploration. Particle at any initial position moves to other position with a certain velocity and direction. In every iteration velocity is adjusted based on particle best solution and social best solution. The velocity of each particle is calculated according to formula shown in (1), while particle's position is updated using (2). This concept makes PSO becomes different with other heuristic methods.

$$v_i^{t+1} = w \times v_i^t + c_1 \times r_1 \times (pbest_i - x_i^t) + c_2 \times r_2 \times (gbest - x_i^t) \quad (1)$$

$$x_i^{t+1} = v_i^t + x_i^t \quad (2)$$

where v_i^t and x_i^t are velocity and position of particle i at iteration t , respectively. The notation w is defined as inertia weight. This is a mechanism to control the exploration and exploitation abilities of the particle and also to eliminate the need for velocity clamping [15]. The cognitive component $c_1 \times r_1 \times (pbest_i - x_i^t)$ resembles the

individual memory of the particle best position, $pbest_i$, whereas $c_2 \times r_2 \times (gbest - x_i^t)$ is the best social position $gbest$. PSO has shown superior performance in solving optimization problem [16]. Recently, this method is also applied for data clustering [17], [18].

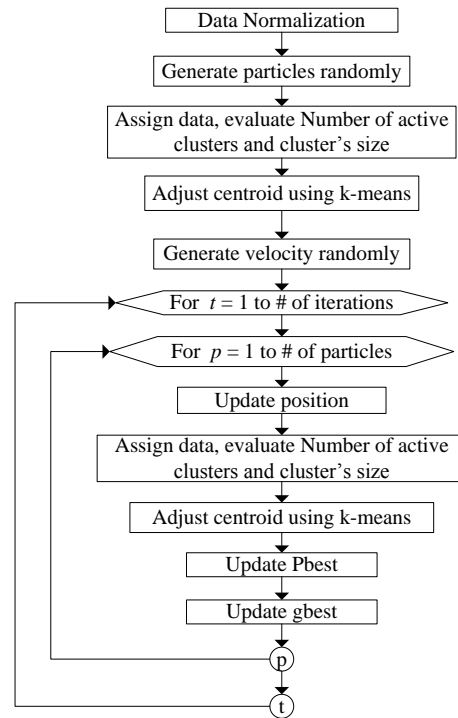


Figure 1. ACPSO algorithm

III. METHODOLOGY

Initially introduced for simulating social behavior, PSO has become a very well-known optimization method. In data mining subject especially data clustering, PSO has been widely applied. In this study, PSO algorithm is extended so that it can overcome automatic clustering problem. Generally, there are two main issues in solving automatic clustering, namely determining cluster number and finding cluster centroids. In order to accommodate these two objectives, the ACPSO employs a particle which comprises of two sections. The first section is responsible for representing number of clusters while the other section informs cluster centroids. Exploration step is conducted towards these two sections gradually. After updating a section, infeasibility constraint will be checked. It covers feasible number of clusters and cluster's size. In addition, K-means is performed to update the second section of each particle based on active clusters obtained by its first section. Complete algorithm proposed in this study is illustrated in Fig 1.

A. Solution Representation

In terms of solution representation, ACPSO adopts a particle representation applied in several previous researches[2], [19]. A particle comprises of two sections. Section one represents number of clusters while section two gives information regarding cluster centroid or data points belongs to the corresponding cluster. Through this

scheme, each particle has its own exploration path so that the search space can be explored wider and deeper.

Supposes data clustering is conducted towards a given dataset $D = \{d_i | i = 1, \dots, S\}$ where S is number of data points and it consists of M attributes. Let the appropriate number of clusters is an integer number in range $[2, N_{\max}]$ where N_{\max} is predicted maximum number of clusters. A particle p in iteration t symbolized as X_p^t comprises of $N_{\max} + M \times N_{\max}$ bits as illustrated in Fig. 2. Herein, all bits are continuous number within $[0,1]$. The first N_{\max} bits represents active or inactive clusters. Cluster k in particle p is an active cluster if the k^{th} bit in X_p^t is greater than or equal to 0.5. Otherwise, the corresponding cluster is inactive. Furthermore, the rest bits correspond to cluster centroid. The initial particle and velocity are generated randomly.

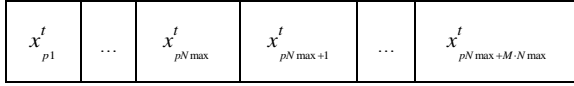


Figure 2. Solution Representation

B. Particle Updating

Each bit i in a particle is updated using (3) and (4). Furthermore, an additional procedure is embedded to guarantee feasibility of the next particle. In this additional procedure, if result from (4), X_p^t , is outside $[0,1]$, sigmoid function is employed to transform the current result to the feasible region.

$$v_{pi}^{t+1} = w^t \times v_{pi}^t + c_1 \times r_1 \times (y_{pi} - x_{pi}^t) + c_2 \times r_2 \times (z_i - x_{pi}^t) \quad (3)$$

$$x_{pi}^{t+1} = x_{pi}^t + v_{pi}^{t+1} \quad (4)$$

$$x_{pi}^{t+1} = \begin{cases} \text{sigmoid}(x_{pi}^{t+1}) & , \text{ if } x_{pi}^{t+1} \notin [0,1] \\ x_{pi}^{t+1} & , \text{ otherwise} \end{cases} \quad (5)$$

where:

v_{pi}^t : velocity of the i^{th} bit in particle p for iteration t .

y_{pi} : the i^{th} bit in particle best p .

y_{pi} : the i^{th} bit in global best.

r_1 and r_2 : random number $N(0,1)$.

In updating the velocity, two parameters are required. There are inertia weight w , and learning rate c_1 and c_2 . ACPSO applies time-varying inertia weight. Herein, the inertia weight linearly decreases based on the (6).

$$w = w_{\max} \frac{(w_{\max} - w_{\min})}{T} \times t \quad (2)$$

where w_{\max} and w_{\min} are maximum and minimum inertia weight, respectively. The T is total iteration and t is current iteration

```

Given particle  $X_p^t = \{x_{pi}^t, \dots, x_{pN_{\max}+M \cdot N_{\max}}^t\}$ ,  $N_p$  is active
clusters in  $X_p^t$ , and  $|N_p|$  is number of active clusters.
If  $|N_p| < 2$ 
    Generate integer number  $i, j \in \{1, \dots, N_{\max}\}, i \neq j$ 
    if  $x_{pi}^t < 0.5$ 
         $x_{pi}^t = x_{pi}^t + 0.5$ 
         $|N_p| = |N_p| + 1$ 
    end if
    if  $x_{pj}^t < 0.5$ 
         $x_{pj}^t = x_{pj}^t + 0.5$ 
         $|N_p| = |N_p| + 1$ 
    end if
end if
    
```

Figure 3. Procedure for evaluating number of clusters

```

Given particle  $X_p^t = \{x_{pi}^t, \dots, x_{pN_{\max}+M \cdot N_{\max}}^t\}$ ,  $N_p$  is active
clusters in  $X_p^t$ ,  $|N_p|$  is number of active clusters, and  $|x_{pi}^t|$  is size
of cluster  $x_{pi}^t$ 
For  $i = 1$  to  $N_{\max}$ 
    if  $x_{pi}^t \geq 0.5$ 
        if  $|x_{pi}^t| < 2$  and  $|N_p| > 2$ 
             $x_{pi}^t = x_{pi}^t - 0.5$  (deactivate  $x_{pi}^t$ )
        if  $|x_{pi}^t| = 1$ 
            reassign data to another cluster which has the shortest
            distance
        end if
         $|N_p| = |N_p| - 1$ 
    else if  $|x_{pi}^t| < 2$  and  $|N_p| \leq 2$ 
        Choose any data points  $d_q, d_r \in D, q \neq r$ 
        Let  $x_{pi}^t$  is original cluster of  $d_q$  and  $d_r$ 
        Assign data  $d_j, d_k$  to cluster  $x_{pi}^t$ 
        Update  $|x_{pi}^t| = |x_{pi}^t| + 2$ 
        Update  $|x_{pj}^t| = |x_{pj}^t| - 2$ 
    end if
end if
end for
    
```

Figure 4. Procedure for evaluating cluster's size

C. Avoiding Infeasible Solution

Existence of infeasible solution is commonly happened in any heuristics methods. For automatic clustering, infeasible solution is occurred if number of active clusters is less than two and cluster's size less than two. Thus, repairing mechanism is performed after particle updating. In this mechanism, if a particle has less than two number of clusters, two random bits are chosen and the corresponding clusters are activated. As showed in Fig. 3, if the chosen bit is currently an active cluster, the final

number of active clusters in this particle is two. Otherwise, there are three active clusters.

Furthermore, all data points are assigned into the nearest active cluster. Afterward, cluster's size is evaluated. A cluster is valid if it has more than two data points. Thus, the next procedure is evaluating cluster's size using a procedure described in Fig. 4. According to this procedure, if invalid cluster exists while number of active clusters in particle p is greater than two, the invalid cluster is eliminated and data point belongs to this cluster is moved to another cluster. However, if number of active clusters is only two, choose any random data points, and move these data into the invalid cluster so that it becomes valid cluster.

D. Fitness Function

In this study, cluster validity is simply measured using VI proposed by Turi[14]. Basically, VI improved standard ratio between intra and inter cluster by integrating with a multiplier function. Intra cluster is defined as cluster compactness measurement. It has to be minimized. Whereas inter cluster is distance between clusters centroid. It is defined as minimum distance between two clusters instead of average distance among clusters. Turi[14] found that standard ratio between intra and inter cluster may lead to small number of clusters such as two or three. Thus, a penalty should be added towards too small number of clusters.

$$intra = \frac{1}{S} \times \sum_{i \in N_p} \sum_{d_j \in X_{pi}} \|d_j - c_i\|^2 \quad (7)$$

$$intra = \min_{i,j \in N_p, i \neq j} \{ \|c_i - c_j\|^2 \} \quad (8)$$

$$VI = (\alpha \times N(\mu, \sigma) + 1) \times \left(\frac{intra}{inter} \right) \quad (9)$$

where α is a constant number, π is 3.14, and $N(\mu, \sigma)$ is Gaussian function as given in (10) with mean μ and standard deviation σ calculated in (10). The $N(\mu, \sigma)$ is a function for specified number of clusters, $|N_p|$.

$$N(\mu, \sigma) = \frac{1}{\sqrt{2 \times \pi \times \sigma^2}} \times \exp \left(- \frac{\left(\left| N_p \right| - \mu \right)^2}{2 \times \sigma^2} \right) \quad (10)$$

IV. EXPERIMENTAL RESULT

Algorithm validation is conducted using four clustering datasets, namely iris, wine, glass, and aggregation. The results are then compared with some previous automatic clustering methods proposed earlier. For comparison purposes, we choose three other methods which have similar basic algorithm. They are DCPSO [12], DCPG[7], and DCGA[7]. DCPSO uses binary PSO to find number of cluster. DCPSO uses a pool of centroid as reference for all particles. Thus, each bit refers to a centroid in a pool of centroid. DCPG is an automatic clustering algorithm which combine PSO and GA for exploring appropriate number of clusters[7]. Result obtained by PSO and GA is then given to K-means algorithm to find the cluster centroid for each cluster. Furthermore, DCGA is an automatic clustering algorithm based on GA.

In this study, ACPSO is built in C++ and the computational experiments are performed on a PC with 2GB of RAM and 2801 MHz CPU speed. Since the aims of validity is to compare method performance, outer factor such as parameter setting is controlled so that it is set as the same with the one used by competitor algorithms. The parameter setting used by ACPSO is mentioned in Table I. The maximum number of clusters is defined as $\sqrt{s/2}$ where s is a total data point in a dataset.

TABLE I. PARAMETER SETTING

Parameter	Value
Iteration	200
population size	20
Learning factors	1.49
Inertia weight	0.9 to 0.4

TABLE II. NUMBER OF CLUSTERS OBTAINED BY ALL ALGORITHMS AVERAGE (STANDARD DEVIATION)

	Actual	ACPSO	DCPG	DCPSO	DCGA
Iris	3	3.6667 (1.2843)	3.0000 (0.0000)*	3.0666 (0.2494)	3.0333 (0.1795)
Wine	3	4.9333 (0.8277)	3.7667 (0.4229)*	3.6333 (0.6046)	3.9333 (0.6289)
Glass	6	6.0000 (0.3714)*	5.9333 (0.7717)	5.4333 (0.6155)	5.8000 (1.4696)
Aggregation	6	6.0000 (0.0000)*	6.1000 (0.4147)	6.1428 (1.0694)	N/A

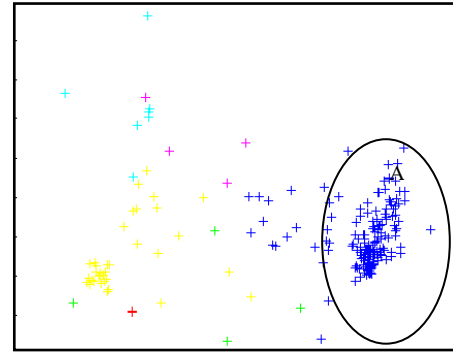
*: best result

For verification, comparison is made based on the average and standard deviation of number of clusters, VI and accuracy. According to Table II DCPG has better result than ACPSO, DCPSO, and DCGA for iris and wine dataset in determining number of clusters. It shows by average number of clusters which approximates the actual number of clusters. However, for glass and aggregation datasets, ACPSO obtains more precise number of clusters.

As showed by VI index summarized in Table III,

ACPSO outperforms DCPG, DCPSO, and DCGA for glass and aggregation datasets. From the standard deviation, ACPSO proves that it has more stable result since it has smaller standard deviation in terms of number of clusters, VI and also accuracy. Furthermore, according to the accuracy listed in Table IV, ACPSO is the best algorithms compare with DCPG and DCPSO. It achieves better accuracy for all datasets although the VI values for iris and wine are not better than those obtained by DCPG

and DCPSO. This result shows that ACPSO can give better result than DCPG and DCPSO especially for higher number of clusters. ACPSO divides iris dataset into 3 and 4 clusters. For the four clusters, one of clusters is a small cluster with number of data less than ten. Similar result is also happened for wine dataset. Therefore, although the number of clusters is not exactly equal to the actual number of clusters, the accuracies for both iris and wine are good since only small number of data points missed clustered. In glass dataset, these three algorithms only can achieve around 40% accuracy. It is caused by these three algorithm grouped data point based on distance between data points and cluster centroid. Thus, closer data points are grouped into a cluster. As illustrated in Fig. 5, there is a region A consisted of more than one cluster. By ACPSO, DCPG, and DCPSO which apply Euclidean distance as similarity measurement, data points within this region A are grouped into one cluster since they are closer to each other. Thus, it causes lower accuracy. Similarity measurement is an important issue in clustering.



(a) ACPSO clustering
Figure 5. Glass dataset

V. CONCLUSION

Automatic clustering is new direction in data clustering. Different with conventional data clustering method which requires predefined number of clusters, automatic clustering works in both determining appropriate number of clusters and finding best centroid for each cluster. Due to its importance especially in new real-world problems, many researchers studied the automatic clustering yet the exact solution for automatic clustering has not been found. Thus, in this research a new automatic clustering method is proposed. The proposed method named ACPSO extends a well-known heuristic method, PSO.

ACPSO applies continuous PSO for both determining number of clusters and finding the cluster centroids. These two objectives are represented in a particle which is comprised of two sections. In exploring the search space, sigmoid function is applied for transforming infeasible solution into feasible solution. Moreover, K-means is performed by each particle to adjust its second section representing clusters centroids. This algorithm is verified using four benchmark datasets in clustering. The results are compared with three previous methods, namely DCPG, DCGA, and DCPSO. The results show that ACPSO outperforms DCPG, DCGA and DCPSO especially in terms of accuracy and consistency. As the conclusion, ACPSO shows more promising results than DCGA, DCPGA, and DCPSO especially for higher number of clusters. Besides, the results also reveal that obtaining better VI does not ensure higher accuracy. Thus, if the main concern is accuracy, other fitness functions that are more related to the accuracy can be applied. Furthermore, similarity measurement also gives significant influence on the results. Further researches can consider the appropriate similarity measurement for the dataset so that it can obtain better result.

TABLE III. VI OBTAINED BY ALL ALGORITHMS AVERAGE (STANDARD DEVIATION)

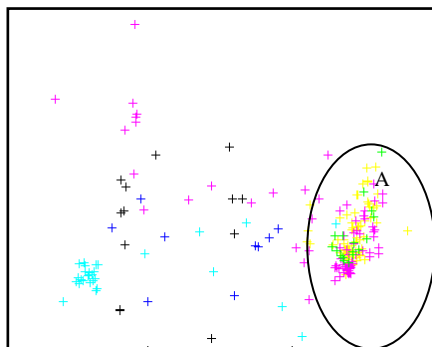
	ACPSO	DCPG	DCPSO	DCGA
Iris	0.3716 (0.0227)	0.3586 (0.0143)*	0.3715 (0.0227)	0.3742 (0.0247)
Wine	0.6248 (0.0101)	0.5813 (0.0238)*	0.5965 (0.0355)	0.6152 (0.0066)
Glass	0.4183 (0.0115)*	0.4699 (0.0483)	0.5390 (0.0849)	0.4436 (0.0423)
Aggregation	0.3986 (0.0001)*	0.4146 (0.0247)	0.4178 (0.0214)	N/A

*: best result

TABLE IV. ACCURACY OF ALL ALGORITHMS AVERAGE (STANDARD DEVIATION)

	ACPSO	DCPG	DCPSO	DCGA
Iris	0.8833 (0.0060)*	0.8634 (0.0526)	0.8663 (0.0609)	N/A
Wine	0.9393 (0.0175)*	0.8438 (0.0716)	0.8695 (0.0743)	N/A
Glass	0.4735 (0.0027)*	0.4714 (0.0319)	0.4571 (0.0411)	N/A
Aggregation	0.8833 (0.0143)*	N/A	N/A	N/A

*: best result



(a) Actual clustering

REFERENCES

- [1] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognition Letters*, vol. 31, pp. 651-666, 2010.
- [2] S. Das, A. Abraham, and A. Konar, "Automatic kernel clustering with a multi-elitist particle swarm optimization algorithm," *Pattern Recogn. Lett.*, vol. 29, pp. 688-699, 2008.
- [3] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*, Boston: Pearson Education, Inc., 2006.

- [4] S.-M. Pan and K.-S. Cheng, "Evolution-Based Tabu Search Approach to Automatic Clustering," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, pp. 827 - 838, 2007.
- [5] D.-X. Chang, X.-D. Zhang, C.-W. Zheng, and D.-M. Zhang, "A robust dynamic niching genetic algorithm with niche migration for automatic clustering problem," *Pattern Recognition*, vol. 43, pp. 1346-1360, 2010.
- [6] Y. Liu, X. Wu, and Y. Shen, "Automatic clustering using genetic algorithms," *Applied Mathematics and Computation*, vol. 218, pp. 1267-1279, 2011.
- [7] R. J. Kuo, Y. J. Syu, Z.-Y. Chen, and F. C. Tien, "Integration of particle swarm optimization and genetic algorithm for dynamic clustering," *Information Sciences*, vol. 195, pp. 124-140, 2012.
- [8] C.-H. Chou, M.-C. Su, and E. Lai, "A new cluster validity measure and its application to image compression," *Pattern Anal. Appl.*, vol. 7, pp. 205-220, 2004.
- [9] M. K. Pakhira, S. Bandyopadhyay, and U. Maulik, "Validity index for crisp and fuzzy clusters," *Pattern Recognition*, vol. 37, pp. 487-501, 2004.
- [10] H. He and Y. Tan, "A two-stage genetic algorithm for automatic clustering," *Neurocomputing*, vol. 81, pp. 49-59, 2012.
- [11] L. Y. Tseng and S. Bien Yang, "A genetic approach to the automatic clustering problem," *Pattern Recognition*, vol. 34, pp. 415-424, 2001.
- [12] M. Omran, A. Salman, and A. Engelbrecht, "Dynamic clustering using particle swarm optimization with application in image segmentation," *Pattern Analysis & Applications*, vol. 8, pp. 332-344, 2006.
- [13] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, pp. 224-227, 1979.
- [14] R. H. Turi, "Clustering-based colour image segmentation," Ph.D, School of Computer Science and Software Engineering, Monash, Australia, 2001.
- [15] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," presented at the 7th International Conference on Evolutionary Programming VII, 1998.
- [16] P. Wei, W. Kang-ping, Z. Chun-guang, and D. Long-jiang, "Fuzzy discrete particle swarm optimization for solving traveling salesman problem," in *Fourth International Conference on Computer and Information Technology*, 2004, pp. 796-800.
- [17] C. Ching-Yi and Y. Fun, "Particle swarm optimization algorithm and its application to clustering analysis," in *IEEE International Conference on Networking, Sensing and Control*, 2004, vol. 2, pp. 789-794.
- [18] D. W. van der Merwe and A. P. Engelbrecht, "Data clustering using particle swarm optimization," in *Congress on Evolutionary Computation*, 2003, vol. 1, pp. 215-220.
- [19] S. Das, A. Abraham, and A. Konar, "Automatic clustering using an improved differential evolution algorithm," *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 38, pp. 218-237, 2008.



R. J. Kuo received the MS degree in Industrial and Manufacturing Systems Engineering from Iowa State University, Ames, IA, in 1990 and the PhD degree in Industrial and Management Systems Engineering from the Pennsylvania State University, University Park, PA, in 1994. Currently, he is the Distinguished Professor in the Department of Industrial Management

at National Taiwan University of Science and Technology, Taiwan. His research interests include architecture issues of computational intelligence and their applications in data mining, electronic business, customer relationship management, supply chain management and decision support systems.



Ferani Eva Zulvia was born in Cirebon, Indonesia on April 23, 1987. She received her B.S degree in Mathematics from Sriwijaya University, Indonesia, M.Eng degree in Industrial Engineering, Bandung Institute of Technology, Indonesia and MBA degree in Industrial Management, National Taiwan University of Science and Technology, Taiwan. Currently, she

is a doctoral student in department of Industrial Management, National Taiwan University of Science and Technology in Taiwan. Her research interest includes heuristic methods and its application in data mining and logistic distribution