Data Clustering Using Chaotic Particle Swarm Optimization

Li-Yeh Chuang, Yu-Da Lin, and Cheng-Hong Yang, *Member, IAENG*

Abstract—**Clustering is an important analysis tool employed in data statistics. Identification of the structure of large-scale data has become an increasingly important topic in data mining problem. We propose Gauss chaotic map particle swarm optimization (GaussPSO) method for clustering, which uses a Gauss chaotic map to adopt a random sequence with a random starting point as a parameter, the method relies on this parameter to update the positions and velocities of the particles. The Gauss chaotic map provides the significant chaos distribution to balance the exploration and exploitation capability of search process. This easy and fast function generates random seed processes, and further improves the performance of PSO due to its unpredictability. In the study, eight different clustering algorithms were extensively compared on six test data sets. The results indicate that the performance of the GaussPSO method is significantly better than the performance of other algorithms for data clustering problems.**

Index Terms—**Data Clustering, Particle Swarm Optimization, Gauss chaotic map.**

I. Introduction

A clustering technique is the process of grouping data from a set of objects. The objects within a cluster are similar to each other, but dissimilar to objects in other clusters. Clustering helps to identify some inherent structures that are presents in the objects. Clustering reflects the statistical structure of the overall collection of input patterns in the data because subsets of patterns and its particular problem have certain meanings [1]. A pattern can mathematically be represented as a vector in a multi-dimensional space.

Many studies have proposed the algorithms for clustering, such as K-means [2], genetic algorithm [3], particle swarm optimization [4-6]. The K-means algorithm is a popular clustering technique that has been successfully applied to many practical clustering problems [2]. However, the K-means algorithm is not convex and may contain many local minima since it suffers from several drawbacks based on its choice of initializations. Recent advancements in clustering algorithms introduced evolutionary computing methods such as genetic algorithms [3] and particle swarm optimization [4-6]. Genetic algorithms typically start with some candidate solutions to an optimization problem. These candidates then evolve towards a better solution through selection, crossover and mutation. The concept of PSO was designed to simulate social behavior via an exchange of information in practical applications. Many studies used PSO to cluster data within a multi-dimensional space and obtained outstanding results. However, the rate of convergence is insufficient when searching for global optima. Fan et al., [7] proposed to combine the Nelder–Mead simplex search method with PSO to improve it. The rationale behind it was that such a hybrid approach enjoys the merits of both PSO and the Nelder–Mead simplex search method. Kao et al. explored the applicability of a hybrid K-means algorithm, Nelder-Mead simplex search method, and particle swarm optimization (K–NM–PSO) to clustering data vectors [8].

PSO adopts a random sequence with a random starting point as a parameter, and relies on this parameter to update the positions and velocities of the particles. However, PSO often leads to premature convergence, especially in complex multi-peak search problems such as the clustering of high-dimensional data. We combined a Gauss chaotic map and particle swarm optimization in a new method named GaussPSO. Results of the conducted experimental trials on a variety of data sets taken from several real-life situations demonstrate that the proposed GaussPSO method is superior to the K-means, PSO, NM-PSO, K-PSO, and K-NM-PSO algorithms [8].

II. Method

A. Particle Swarm Optimization (PSO)

The original PSO method [9] is a population-based optimization technique, where a population is called a swarm. Every particle in the swarm is analogous to an individual "fish" in a school of fish. The swarm consists of *N* particles moving around a *D*-dimensional search space. Every particle makes use of its own memory and knowledge gained by the swarm as a whole to find the best solution. The value $pbest_i$ is introduced as the best previously visited position of the i_{th} particle; it is denoted $p_{\mathrm{i}} = (p_{i1}, p_{i2}, \ldots, p_{iD})$. The *gbest* value is the global best position of the all individual $pbest_i$ values; it is denoted $g = (g_1, g_2, \ldots, g_D)$. The position of the i_{th} particle is represented by $x_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$, x□ $(X_{\min}, X_{\max})^D$ and its velocity is represented by $v_i = (v_{i1}, v_{i2}, \ldots, v_{i\mathrm{D}})$, v□ $[V_{\min}, V_{\max}]^D$. The position and velocity of the i_{th} particle are updated by $pbest_i$ and *gbest* in the each generation. The update equations can be formulated as:

L.Y. Chuang is with the Department of Chemical Engineering, I-Shou University , 84001 , Kaohsiung, Taiwan (E-mail: chuang@isu.edu.tw)
Y.D. Lin is with the Department of Electronic Engineering, National Kaohsiung University of Applied Sciences, 80778, Kaohsiung, Taiwan (E-mail: e0955767257@yahoo.com.tw)
C.H. Yang is with the Department of Electronic Engineering, National Kaohsiung University of Applied Sciences, 80778, Kaohsiung Taiwan (phone: 886-7-3814526#5639; E-mail: chyang@cc.kuas.edu.tw). He is also with the Network Systems Department, Toko University, 61363, Chiayi, Taiwan. (E-mail: chyang@cc.kuas.edu.tw).

$$v_{id}^{new} = w \times v_{id}^{old} + c_1 \times r_1 \times \left(pbest_{id} - x_{id}^{old}\right) + c_2 \times r_2 \times \left(gbest_d - x_{id}^{old}\right) \quad (1)$$

$$x_{id}^{new} = x_{id}^{old} + v_{id}^{new} \quad (2)$$

where r_1 and r_2 are random numbers between (0, 1), c_1 and c_2 control how far a particle will move in once generation, v_{id}^{new} and v_{id}^{old} denote, respectively, the velocities of the new and old particle, x_{id}^{old} is the current particle position, and x_{id}^{new} is the updated particle position. The inertia weight w controls the impact of the previous velocity of a particle on its current one; w is designed to replace $V_{\max}$ and adjust the influence of previous particle velocities on the optimization process. An acceptable trade-off between exploration and exploitation is essential. One of the most important considerations in PSO is how to effectively balance the global and local search abilities of the swarm, because the proper balance of global and local search over the entire run is critical to the success of PSO [10]. In general the inertia weight decreases linearly from 0.9 to 0.4 throughout the search process [11]. The relevant equation can be written as:

$$w_{LDW} = \left(w_{\max} - w_{\min}\right) \times \frac{Iteration_{\max} - Iteration_i}{Iteration_{\max}} + w_{\min} \quad (3)$$

where $w_{\max}$ is 0.9, $w_{\min}$ is 0.4 and $Iteration_{\max}$ is the maximum number of allowed iterations.

B. Gauss Chaotic Map Particle Swarm Optimization (GaussPSO)

The Gauss chaotic map is similar to a quadratic transformation in the sense that it allows a complete analysis of the qualitative and quantitative properties of chaos. It provides the continued fraction expansion of numbers, which is an analogy to the shift transformation corresponding to the quadratic iterator. This shift transformation satisfies the properties of chaos — dense periodic points, mixing and sensitivity [12]. We used these characteristics of the Gauss chaotic map and adaptive action to avoid entrapment of the PSO in a local optimum.

In PSO, the parameters w, r_1 and r_2 the key factors affecting the convergence behavior. The factors r_1 and r_2 control the balance between the global exploration and the local search ability. An inertia weight w that linearly decrease from 0.9 to 0.4 throughout the search process is usually adopted [11]. A Gauss chaotic map is a frequently used chaotic behavior map that can quickly generate chaotic sequences. The sequences can easily be stored so there is no need to store long sequences. In Gauss chaotic map PSO (GaussPSO), sequences generated by the Gauss chaotic map substitute the random parameters r_1 and r_2. The parameters r_1 and r_2 are modified based on the following equation.

$$Gr(x) = \begin{cases} 0, & Gr(x) = 0 \\ Frac(\frac{1}{x}) = \frac{1}{x} \bmod 1, & Gr(x) \in (0,1) \end{cases} \quad (4)$$

The velocity update equation for GaussPSO can thus be formulated as:

$$\begin{aligned} v_{id}^{new} = w \times v_{id}^{old} &+ c_1 \times Gr_1 \times \left(pbest_{id} - x_{id}^{old}\right) \\ &+ c_2 \times Gr_2 \times \left(gbest_d - x_{id}^{old}\right) \end{aligned} \quad (5)$$

where Gr is a function based on the results of the Gauss chaotic map with values between 0.0 and 1.0. The pseudo-code of the GaussPSO is shown below.

GaussPSO Pseudo-Code

```
Begin
Initialize particle swarm
While (number of iterations, or the stopping criterion is not met)
  Evaluate fitness of particle swarm
  For $n$ = 1 to number of particles
    Find $pbest$
    Find $gbest$
    For $d$ = 1 to number of dimension of particle
      Update the position of particles via equations 5 and 2
    Next $d$
  Next $n$
  Update the inertia weight value via equation 3
  Update the value of $Gr$ via equation 4
Next generation until stopping criterion
End
```

C. Application of the PSO algorithm

a) Initial particle swarm

The 3×N particles are randomly generated with an individual position and velocity in the solution space. The generated position for the ith particle is defined as x_i (x_i□ $\{x_{i1}, x_{i2}, \ldots, x_{in}\}$) and the velocity is defined as v_i ($v_i \in \{v_{i1}, v_{i2}, \ldots, v_{in}\}$), where n is the number of particles. Every particle is composed of K center positions for each cluster, where K is the anticipated number of clusters. N is computed as follow:

$$N = K \times d \quad (6)$$

where d is the data set dimension. For example, a possible encoding of a particle for a two-dimensional problem with three clusters is illustrated in Fig. 1. The three cluster centers in the particle X$_i$ are randomly generated as X$_1$= (2.5, 2.7, 4.5, 5, 1.2, 2.2). The particle dimension is N = 6, i.e., K=3, d=2 and the population size is 18.

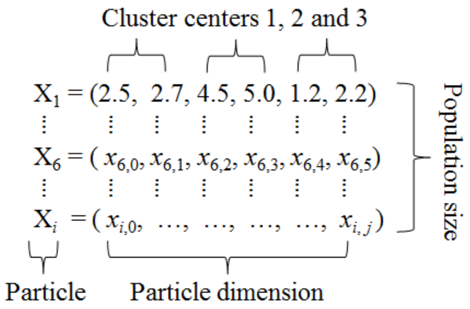
Cluster centers 1, 2 and 3
X1 = (2.5, 2.7, 4.5, 5.0, 1.2, 2.2)
X6 = (x6,0, x6,1, x6,2, x6,3, x6,4, x6,5)
Xi = (xi,0, …, …, …, …, xi,j)
Population size
Particle
Particle dimension

Fig. 1. Encoding of particles in PSO

b) Grouping the data vectors for every particle

All data set are grouped into K clusters according to the data vectors on the basis of the Euclidean distance measurement. A matrix $x_i = (C_1, C_2, \ldots, C_j, .., C_K)$, where C_j represents the jth cluster centroid vector and K is the number of clusters, is used to calculate the distance as the length between the data vector and the centroid vector of the respective cluster in every particle; the calculation is described in equation 7. Each data vector is assigned to the cluster with the shortest distance.

$$\mathrm{D}(x_p \cdot z_j) = \sqrt{\sum_{i=1}^{d} (x_{pi} - z_{ji})^2} \qquad (7)$$

$$z_j = \frac{1}{n_j} \sum_{\forall x_p \in c_j} x_p \qquad (8)$$

c) Particle fitness evaluation

The fitness value of each particle is computed by the following fitness function. The fitness value is the sum of the intra-cluster distances of all clusters. This sum of distances has a profound impact on the error rate.

$$fitness = \sum \left| X_j - Z_i \right|, i = 1, \ldots, K, \; j = 1, \ldots, n \qquad (9)$$

where K and n are the numbers of clusters and data sets, respectively. Z_i is the cluster center of i and X_j is the data point j.

d) Update *pbest* and *gbest*

At each iteration, each particle compares its current fitness value with the fitness value of its own *pbest* solution and the fitness value of the population's *gbest* solution. The *pbest* and *gbest* values are updated if the new values are better than the old ones. If the fitness value of each particle X_i in the current generation is better than the previous *pbest* fitness value, then both the position and fitness value of *pbest* are updated to X_i. Similarly, if the fitness value of *pbest* in the current generation is better than the previous *gbest* fitness value, then both the position and fitness value of *gbest* are updated to X_i.

III. Result and Discussion

A. Parameter settings

In our experiments, the iteration was set to 1000 and the population size was set to 50. The acceleration parameters were for PSO set to c_1=c_2=2. $V_{\max}$ was equal to ($X_{\max} - X_{\min}$) and $V_{\min}$ was equal to $-$ ($X_{\max} - X_{\min}$) [9]. The results are the averages of 50 simulation runs. For each run, 10 × N iterations were carried out for each of the six data sets in every algorithm when solving an N-dimensional problem. The criterion 10 × N was adopted in many previous experiments with a great success in terms of its effectiveness [8].

Table I
Summary of the characteristics of the considered data sets

Name of data set	Number of classes	Number of features	Size of data set (size of classes in parentheses)
Vowel	6	3	871 (72, 89, 172, 151, 207, 180)
Iris	3	4	150 (50, 50, 50)
Crude Oil	3	5	56 (7, 11, 38)
CMC	3	9	1473 (629, 334, 510)
Cancer	2	9	683 (444, 239)
Wine	3	13	178 (59, 71, 48)

B. Data sets

Six experimental data sets, i.e., Vowel, Iris, Crude oil, CMC, Cancer, and Wine are used to assess the quality of the respective clustering methods. These data sets represent examples of data with low, medium and high dimensions. All data sets are available at ftp://ftp.ics.uci.edu/pub/machine-learning-databases/.

Table I summarizes the characteristics of these data sets. Given is a data set with three features that are grouped into two clusters. The number of parameters are optimized in order to find two optimal cluster center vectors that are equal to the product of the number of clusters and the number of features as $N = k \times d = 2 \times 3 = 6$. The six real-life data sets are described below:

(1) The Vowel data set (n = 871, d = 3, k = 6) consists of 871 Indian Telugu vowel sounds. It includes the three features corresponding to the first, second and third vowel frequencies, and six overlapping classes {d (72 objects), a (89 objects), i (172 objects), u (151 objects), e(207 objects), o (180 objects)}.

(2) Fisher's iris data set (n = 150, d = 4, k = 3) consists of three different species of iris flowers: iris setosa, iris virginica and iris versicolour. For each species, 50 samples were collected from four features, namely sepal length, sepal width, petal length and petal width.

(3) The Crude oil data set (n = 56, d = 5, k = 3) consists of 56 objects characterized by five features: vanadium, iron, beryllium, saturated hydrocarbons, and aromatic hydrocarbons. Three crude-oil samples were collected from three zones of sandstone (Wilhelm has 7 objects, Sub-Mulnia has 11 objects, and Upper has 38 objects).

(4) The Contraceptive Method Choice (denoted CMC with n = 1473, d = 9, k = 3) consists of a subset of the 1987 National Indonesia Contraceptive Prevalence Survey. The samples consist of married women who were either not pregnant or not sure of their pregnancy status at the time the interviews were conducted. It predicts the choice of the current contraceptive method (no contraception has 629 objects, long-term methods have 334 objects, and short-term methods have 510 objects) of a woman based on her demographic and socioeconomic characteristics.

(5) The Wisconsin breast cancer data set (n = 683, d = 9, k = 2) consists of 683 objects characterized by nine features: clump thickness, cell size uniformity, cell shape uniformity, marginal adhesion, single epithelial cell size,

bare nuclei, bland chromatin, normal nucleoli and mitoses. There are two categories in the data; malignant tumors (444 objects) and benign tumors (239 objects).

(6) The Wine data set (n = 178, d = 13, k = 3) consists of 178 objects characterized by 13 features: alcohol content, malic acid content, ash content, alkalinity of ash, concentration of magnesium, total phenols, flavanoids, nonflavanoid phenols, and proanthocyanins, and color intensity, hue and OD280/OD315 of diluted wines and pralines. These features were obtained by chemical analysis of wines that are produced in the same region in Italy but derived from three different cultivars. The quantities of objects in the three categories of the data sets are: class 1 (59 objects), class 2 (71 objects), and class 3 (48 objects).

C. Statistical significance test

Results from GaussPSO were compared to other methods, i.e., K-means, GA, KGA, PSO, NM-PSO, K-PSO, and K-NM-PSO, to demonstrate the data clustering capability. The quality of the respective clustering was measured by the following four criteria:

(1) The sum of the intra-cluster distances: The distances between data vectors within a cluster and the centroid of the cluster are defined in equation 7; a higher quality of clustering is represent by relatively small.

(2) Error rate: The numbers of misplaced points are divided by the total number of points, as shown in equation 10:

$$error = \left(\sum_{i=1}^{n} \begin{Bmatrix} 1, A_i \neq B_i \\ 0, A_i = B_i \end{Bmatrix} \right) \times 100 \qquad (10)$$

where n denotes the total number of points. A_i and B_i denote the data sets of which the ith point is a member before and after of clustering. In Table II an example is shown for two data points (2.5, 4.5) and (7.5, 5.5) that are outside of clusters; 1 and 2 are misplaced, and the error rate is 2/5, i.e., 40%.

Table II
ERROR RATE CALCULATIONS

i	Data point	A_i	B_i	Not misplaced (0) / misplaced (1)
1	(4.0, 5.0)	2	2	0
2	(2.5, 4.5)	1	2	1
3	(4.5, 3.5)	2	2	0
4	(7.5, 5.5)	1	2	1
5	(5.0, 6.0)	2	2	0

D. Experimental Results and Discussion

In this section, the performances of GaussPSO and the other methods from 20 runs simulations are compared by means of the best fitness values and the standard deviation among six data sets. Table III summarizes the intra-cluster distances and error rates obtained from the eight clustering algorithms from the six data sets.

The test results clearly show that PSO outperforms the GA method, independent of whether the average intra-cluster distance or best intra-cluster distance is measured. For K-PSO compared with KGA, K-PSO still shows better results than KGA. PSO offers better optimized solutions than GA with or without integration of the K-means method. For the all data sets, the averages and standard deviation of the GaussPSO is better than the ones of K-PSO and K-NM-PSO, in which K-PSO is a hybrid of the K-means and PSO algorithm, and K-NM-PSO is a hybrid of the K-means, Nelder–Mead simplex search [13] and PSO. Please note that in terms of the best distance, PSO, NM-PSO, K-PSO and K-NM-PSO all have a larger standard deviation than GaussPSO, even though they may achieve a global optimum. This means that PSO, NM-PSO, K-PSO, K-NM-PSO are weaker search tools for global optima than GaussPSO if all algorithms are executed just once. It follows that GaussPSO is more efficient in finding a globally optimum solution than the other four PSO methods. For the error rates, standard deviations of the error rates and the best solution of the error rates are implemented on 20 simulation runs. Table IV lists the number of objective function evaluations required by the eight methods after 10 × N iterations. K-means algorithm needed the least number of fewest function evaluations on all data sets, but its results are less than satisfactory, as seen in Table III. GaussPSO had the same number of function evaluations, and thus fewer than PSO, NM-PSO, K-PSO and K-NM-PSO in terms of an average.

E. Advantage of the Gauss chaotic map algorithm

The Gauss chaotic map is a very powerful tool for avoiding entrapment in local optima and does not increase the complexity. The computational complexity for GaussPSO and PSO can be derived as O(PG), where P is the population size and G is the number of generations. In equation 5, we observe that the chaotic map is only used to amend the PSO updating equation.

The standard PSO, together with each individual and the entire population, evolves towards a best fitness; the fitness function is evaluated with an objective function. Although this scheme increases the convergence capability, i.e., evolves the population toward better fitness, the convergence speed is too fast, and thus the population may get stuck in a local optimum since the swarms diversity rapidly decreases. On the other hand, an arbitrarily slow search is not feasible if we want PSO to be effective.

TABLE III
COMPARISON OF INTRA-CLUSTER DISTANCES AND ERROR RATES FOR GAUSSPSO, K-MEANS, GA, KGA, PSO, NM-PSO, K-PSO, AND K-NM-PSO

Data set	Criteria	Method							
		K-means	GA	KGA	PSO	NM-PSO	K-PSO	K-NM-PSO	GaussPSO
Vowel	Average	159242.87	390088.24	149368.45	168477.00	151983.91	149375.70	149141.40	**149015.50**
	STD	916	N/A	N/A	3715.73	4386.43	155.56	120.38	120.67
	Best	149422.26	383484.15	149356.01	163882.00	149240.02	149206.10	149005.00	148967.20
	Error rates (%)	44.26	N/A	N/A	44.65	41.96	42.24	41.94	42.10
	STD	2.15	N/A	N/A	2.55	0.98	0.95	0.95	1.59
	Best	42.02	N/A	N/A	41.45	40.07	40.64	40.64	39.84
Iris	Average	106.05	135.40	97.10	103.51	100.72	96.76	96.67	**96.66**
	STD	14.11	N/A	N/A	9.69	5.82	0.07	0.008	6.551E-4
	Best	97.33	124.13	97.10	96.66	96.66	96.66	96.66	96.66
	Error rates (%)	17.80	N/A	N/A	12.53	11.13	10.20	10.07	10.00
	STD	10.72	N/A	N/A	5.38	3.02	0.32	0.21	0.00
	Best	10.67	N/A	N/A	10.00	8.00	10.00	10.00	10.00
Crude Oil	Average	287.36	308.16	278.97	285.51	277.59	277.77	277.29	**277.23**
	STD	25.41	N/A	N/A	10.31	0.37	0.33	0.095	3.465E-2
	Best	279.20	297.05	278.97	279.07	277.19	277.45	277.15	277.21
	Error rates (%)	24.46	N/A	N/A	24.64	24.29	24.29	23.93	26.43
	STD	1.21	N/A	N/A	1.73	0.75	0.92	0.72	0.71
	Best	23.21	N/A	N/A	23.21	23.21	23.21	23.21	25
CMC	Average	5693.60	N/A	N/A	5734.20	5563.40	5532.90	5532.70	**5532.18**
	STD	473.14	N/A	N/A	289.00	30.27	0.09	0.23	4.055E-5
	Best	5542.20	N/A	N/A	5538.50	5537.30	5532.88	5532.40	5532.18
	Error rates (%)	54.49	N/A	N/A	54.41	54.47	54.38	54.38	54.38
	STD	0.04	N/A	N/A	0.13	0.06	0.00	0.054	0.00
	Best	54.45	N/A	N/A	54.24	54.38	54.38	54.31	54.38
Cancer	Average	2988.30	N/A	N/A	3334.60	2977.70	2965.80	2964.70	**2964.39**
	STD	0.46	N/A	N/A	357.66	13.73	1.63	0.15	8.21E-6
	Best	2987	N/A	N/A	2976.30	2965.59	2964.50	2964.50	2964.39
	Error rates (%)	4.08	N/A	N/A	5.11	4.28	3.66	3.66	3.51
	STD	0.46	N/A	N/A	1.32	1.10	0.00	0.00	0.00
	Best	3.95	N/A	N/A	3.66	3.66	3.66	3.66	3.51
Wine	Average	18061.00	N/A	N/A	16311.00	16303.00	16294.00	16293.00	**16292.68**
	STD	793.21	N/A	N/A	22.98	4.28	1.70	0.46	0.66
	Best	16555.68	N/A	N/A	16294.00	16292.00	16292.00	16292.00	16292.18
	Error rates (%)	31.12	N/A	N/A	28.71	28.48	28.48	28.37	28.31
	STD	0.71	N/A	N/A	0.27	0.27	0.40	0.27	0.28
	Best	29.78	N/A	N/A	28.09	28.09	28.09	28.09	28.09

The results of K-means, GA, KGA, PSO, NM-PSO, K-PSO, K-NM-PSO can be found in [8].

TABLE IV
NUMBER OF FUNCTION EVALUATIONS OF EACH CLUSTERING ALGORITHM

Data set	Method					
	K-means	PSO	NM-PSO	K-PSO	K-NM-PSO	GaussPSO
Vowel	180	16,290	10,501	15,133	**9,291**	9,774
Iris	120	7,260	4,836	6,906	4,556	**4,356**
Crude Oil	150	11,325	7,394	10,807	7,057	**6,795**
CMC	270	36,585	23,027	34,843	**21,597**	21,951
Cancer	180	16,290	10,485	15,756	10,149	**9,774**
Wine	390	73,245	47,309	74,305	46,459	**45,747**
Average	215	26,833	17,259	26,292	16,519	**16,400**

The results of K-means, PSO, NM-PSO, K-PSO, K-NM-PSO can be found in [8].

The Gauss chaotic map is a non-linear system with ergodicity, stochastic and regularity properties; it is very sensitive to its initial conditions and parameters. Consequently, the efficiency of GaussPSO is better than the one of standard PSO because of the chaotic property, i.e., small variations in an initial variable will result in huge differences in the solutions after a few iterations. Since chaotic maps are frequently used chaotic behavior maps and chaotic sequences can be quickly generated and easily stored, there is no need for storage of long sequences [12, 14]. All the evidence gathered in the simulations illustrates that GaussPSO converges to global optima with fewer function evaluations and a smaller error rate than the other algorithms, which naturally leads to the conclusion that GaussPSO is a viable and robust technique for data clustering.

IV. CONCLUSION

In this study, we propose GaussPSO to solve data clustering problems. Six public recognizable UCI data sets are used to investigate the performance in experiments. The minimum intra-cluster distances as a metric search robustly clusters data centers in *N*-dimensional Euclidean space. Experimental results show that GaussPSO reaches a minimal error rate and possessed the fastest convergence, and yields the highest stability of results.

Acknowledgment

This work is partly supported by the National Science Council in Taiwan under grant NSC99-2221-E-151-056-.

REFERENCES

[1] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *Acm Computing Surveys,* vol. 31, pp. 264-323, Sep 1999.

[2] S. Z. Selim and M. A. Ismail, "K-Means-Type Algorithms: A Generalized Convergence Theorem and Characterization of Local Optimality," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 6, pp. 81-87, 1984.

[3] K. J. Kim and H. Ahn, "A recommender system using GA K-means clustering in an online shopping market," *Expert Systems with Applications,* vol. 34, pp. 1200-1209, Feb 2008.

[4] S. Paterlini and T. Krink, "Differential evolution and particle swarm optimisation in partitional clustering," *Computational Statistics & Data Analysis,* vol. 50, pp. 1220-1247, Mar 1 2006.

[5] S. Rana, S. Jasola, and R. Kumar, "A review on particle swarm optimization algorithms and their applications to data clustering," *Artificial Intelligence Review,* vol. 35, pp. 211-222, Mar 2011.

[6] Li-Yeh Chuang, Yu-Da Lin, and C.-H. Yang, "An Improved Particle Swarm Optimization for Data Clustering," in *Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2012, IMECS 2012* Hong Kong, 2012, pp. 440-445.

[7] S. K. S. Fan, Y. C. Liang, and E. Zahara, "Hybrid simplex search and particle swarm optimization for the global optimization of multimodal functions," *Engineering Optimization,* vol. 36, pp. 401-418, Aug 2004.

[8] Y. T. Kao, E. Zahara, and I. W. Kao, "A hybridized approach to data clustering," *Expert Systems with Applications,* vol. 34, pp. 1754-1762, 2008.

[9] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *IEEE International Joint Conference on Neural Network*. vol. 4, 1995, pp. 1942-1948.

[10] Y. Shi and R. C. Eberhart, "Fuzzy adaptive particle swarm optimization," in *In Proceedings of the 2001 congress on evolutionary computation*, 2001, pp. 101-106.

[11] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in *In Proceedings of Congress on Evolutionary Computation*, Washington, D.C., 2002, pp. 1945-1949.

[12] Heinz-Otto Peitgen, Hartmut Jürgens, and D. Saupe, *Chaos and fractals: new frontiers of science* New York: Springer, 2004.

[13] J. A. Nelder and R. Mead, "simplex method for function minimization," *Computer Journal,* vol. 7, pp. 308-313, 1965.

[14] H. J. Gao, Y. S. Zhang, S. Y. Liang, and D. Q. Li, "A new chaotic algorithm for image encryption," *Chaos Solitons & Fractals,* vol. 29, pp. 393-399, Jul 2006.