Abstract—In recent years, clustering is still a popular analysis tool for data statistics. The data structure identifying from the large-scale data has become a very important issue in the data mining problem. In this paper, an improved particle swarm optimization based on Gauss chaotic map for clustering is proposed. Gauss chaotic map 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. It provides the significant chaos distribution to balance the exploration and exploitation capability for search process. This easy and fast function generates a random seed processes, and further improve the performance of PSO due to their unpredictability. In the experiments, the eight different clustering algorithms were extensively compared on six test data. The results indicate that the performance of our proposed method is significantly better than the performance of other algorithms for data clustering problem.

Index Terms—Data Clustering, Particle Swarm Optimization.

I. INTRODUCTION

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

K-means algorithm is a popular clustering technique and it was successfully applied to many of practical clustering problems [2]. However, the K-means is not convex and it may contain many local minima since it suffers from several drawbacks due to its choice of initializations. Recent advancements in clustering algorithm introduce the evolutionary computing such as genetic algorithms [3] and particle swarm optimization [4, 5]. Genetic algorithms typically start with some candidate solutions to the optimization problem and these candidates evolve towards a better solution through selection, crossover and mutation. The concept of PSO was designed to simulate social behavior which major property is information exchange and in practical applications. Many studies used PSO to cluster data within multi-dimensional space and obtained the outstanding results. However, the rate of convergence is insufficient when it searches global optima. Fan et al., [6] proposed to combine Nelder–Mead simplex search method with PSO, the rationale behind it being that such a hybrid approach will enjoy the merits of both PSO and Nelder–Mead simplex search method. Kao et al., explore the applicability of the hybrid K-means algorithm, Nelder-Mead simplex search method, and particle swarm optimization (K–NM–PSO) to clustering data vectors [7].

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 clustering of high-dimensional. We combined the Gauss chaotic Map and particle swarm optimization, named GaussPSO. Results of the conducted experimental trials on a variety of data sets taken from several real-life situations demonstrate that proposed GaussPSO is superior to the K-means, PSO, NM-PSO, K-PSO, and K-NM-PSO algorithms [7].

II. METHOD

A. Particle Swarm Optimization (PSO)

The original PSO method [8] is a population-based optimization technique, where a population is called a swarm. Every particle in swarm is analogous to an individual “fish” in a school, and it can be seemed a 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 pbesti is introduced as the best previously visited position of the ith particle; it is denoted as pbesti = (x1, x2, …, xD). The best is the global best position of all individual pbesti values; it is denoted as the g = (g1, g2, …, gD). The position of the ith particle is represented by xj = (xj1, xj2, …, xjD), x ∈ [Xmin, Xmax]D and its velocity is represented as vj = (v1, v2, …, vD), v ∈ [Vmin, Vmax]D. The position and velocity of the ith particle are updated by pbest, and gbest in the each generation. The update equations can be formulated as:

\[ v_{id}^{new} = w \times v_{id}^{old} + c_1 \times r_1 \times (pbest_{id} - x_{id}^{old}) + c_2 \times r_2 \times (gbest_{id} - x_{id}^{old}) \]  \( (1) \)
\[
x_{id}^{new} = x_{id}^{old} + v_{id}^{new}
\]  \hspace{1cm} (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 one 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; \( x_{id}^{new} \) is a 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. For high-performance problem, a suitable tradeoff 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 [9]. In general, the inertia weight decreases linearly from 0.9 to 0.4 throughout the search process [10]. The respective equation can be written as:

\[
w_{LW} = (w_{max} - w_{min}) \times \frac{\text{Iteration}_{max} - \text{Iteration}}{\text{Iteration}_{max}} + w_{min} \hspace{1cm} (3)
\]

where \( w_{max} \) is 0.9, \( w_{min} \) is 0.4 and \( \text{Iteration}_{max} \) is the maximum number of allowed iterations.

B. Gauss chaotic Map Particle Swarm Optimization (GaussPSO)

Gauss chaotic map is similar to the quadratic transformation in the sense that it allows a complete analysis of its 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 can be satisfied by an analogy to the shift transformation corresponding to the quadratic iterator. This shift transformation can be satisfied by the anticipated number of clusters. 

\[
\text{The velocity update equation for GaussPSO can thus be formulated as:}
\]

\[
v_{id}^{new} = w \times v_{id}^{old} + c_1 \times Gr \times (pbest_{id} - x_{id}^{old}) + c_2 \times Gr \times (gbest_{id} - x_{id}^{old}) \hspace{1cm} (5)
\]

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**

01: Begin
02: Initial particle swarm
03: While (number of iterations, or the stopping criterion is not met)
04: Evaluate fitness of particle swarm
05: For \( n = 1 \) to number of particles
06: Find \( pbest \)
07: Find \( gbest \)
08: For \( d = 1 \) to number of dimension of particle
09: Update the position of particles by equations 5 and 2
10: Next \( d \)
11: Next \( n \)
12: Update the inertia weight value by equation 3
13: Update the value of \( Gr \) by equation 4
14: Next generation until stopping criterion
15: End

---

C. The application of the PSO algorithm

a) Initial particle swarm

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

\[
N = K \times d \hspace{1cm} (6)
\]

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 this particle \( X_i \) are randomly generated as \( X_1 = (2.5, 2.7, 4.5, 5.0, 1.2, 2.2) \) and the particle dimension is \( N = 6 \), i.e., \( K = 3, d = 2 \) and the population size is 18.

---

**Fig. 1. Encoding of particles in PSO**

---

The 3×6 particles are randomly generated with an individual position and velocity in the solution space. The generated position for the 4th particle is defined as \( x_4 (x_4 \in [x_{41}, x_{42}, \ldots, x_{4d}]) \) and the velocity is defined as \( v_4 (v_4 \in [v_{41}, v_{42}, \ldots, v_{4d}]) \), where \( n \) is the number of particle. Every particle is composed of \( K \) center positions for each cluster, where \( K \) is the anticipated number of clusters. \( N \) is computed as follow:

\[
N = K \times d \hspace{1cm} (6)
\]

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 this particle \( X_i \) are randomly generated as \( X_1 = (2.5, 2.7, 4.5, 5.0, 1.2, 2.2) \) and the particle dimension is \( N = 6 \), i.e., \( K = 3, d = 2 \) and the population size is 18.
b) Grouping the data vectors for every particle

The all data set are grouped into K clusters according to the data vectors on the basis of the Euclidean distance as the similar measurement. A matrix \( x_j = (C_{ij}, C_{j1}, ..., C_{jk}) \), where \( C_j \) represents the \( j \)th cluster centroid vector and \( K \) is the number of clusters, is calculated 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. For each data vector, it is assigned to the cluster with the shortest distance.

\[
D(x_p, z_j) = \sqrt{\sum_{i=1}^{d} (x_{pi} - z_{pj})^2} \quad (7)
\]

\[
z_j = \frac{1}{n_j} \sum_{i \in x_j} x_{pi} \quad (8)
\]

c) Fitness evaluation of each particle

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 distance has a profound impact on the error rate.

\[
\text{fitness} = \sum_{i=1}^{K} \sum_{j=1}^{n} |x_{pi} - z_{ij}| \quad (9)
\]

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

d) Update pbest and gbest

In each of the iteration, each particle will compare 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 of the position and fitness value of pbest will be updated as \( X_i \). Similarly, if the fitness value of gbest in the current generation is better than previous gbest fitness value, then both of the position and fitness value of gbest will be updated as \( X_i \).

III. RESULT AND DISCUSSION

A. Parameter settings

In an experiment, the iteration was set to 1000 and the population size was set to 50. The acceleration parameters were for PSO were set to \( c_1 = c_2 = 2 \). \( V_{\text{max}} \) was equal to \((X_{\text{max}} - X_{\text{min}})\) and \( V_{\text{min}} \) was equal to \( -(X_{\text{max}} - X_{\text{min}}) \) [8]. 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 [7].

B. Data sets

Six experimental data sets, i.e., Vowel, Iris, Crude oil, CMC, Cancer, and Wine are used to test the qualities 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 the 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 the three different species of iris flowers: iris setosa, iris virginica and iris versicolor. For each species, 50 samples were collected from the four features that are 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 the 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 the married women who were either not pregnant or not sure of their pregnancy 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. Test for statistical significance

Results from GaussPSO was compared with the other methods, i.e., K-means, GA, KGA, PSO, NM-PSO, K-PSO, and K-NM-PSO, to demonstrate the capability of data clustering. 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, and a higher quality of clustering represents that the sum is relatively small.

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

\[
error = \left( \sum_{i=1}^{n} \left| A_i \neq B_i \right| \right) \times 100 
\]

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

D. Experimental Results and Discussion

In this section, the performances of GaussPSO, and other proposed 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 are clearly shown that the PSO outperforms the GA method, independent of whether the average intra-cluster distance or best intra-cluster distance is measured. For K-PSO compare with KGA, K-PSO still leads KGA, however, 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 for 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 [12] 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 are more efficient in finding the global 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 from the 20 simulation runs. Table IV lists the number of objective function evaluations required by the seven methods after \( 10 \times N \) iterations. K-means algorithm has newest function evaluations on all data sets, but its results are less than satisfactory, as seen in Table III. GaussPSO is the same function evaluations, and they are less 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, besides it 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 can observe that the chaotic map is only used to amend the PSO updating equation.

The standard PSO, together with each individual and the whole population, evolves towards best fitness, in which the fitness function is evaluated with the objective function. Although this scheme has the property to increase the convergence capability, i.e., to evolve the population toward better fitness, but the convergence speed is too fast, the population may get stuck in a local optimum, since the swarms diversity rapidly decreases. On the other hand, it cannot be searched arbitrarily slowly if we want PSO to be effective.

Gauss chaotic map is a non-linear system with ergodicity, stochastic and regularity properties, and is very sensitive to its initial conditions and parameters. Consequently, the efficiency of GaussPSO is better than the standard PSO because of the chaotic property, i.e., small variation in an initial variable will result in huge difference in the solutions after some iteration. Since chaotic maps are frequently used chaotic behavior maps and the chaotic sequences can be quickly generated and easily stored, there is no need for storage of long sequences [11, 13].

Summary 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

The novel method GaussPSO is introduced to solve the data clustering problems. This study used the six public recognizable UCI data sets to investigate the performance through our experiments. We use minimum intra-cluster distances as a metric to search robustly data cluster centers in N-dimensional Euclidean space. The experimental results demonstrate that our proposed clustering algorithm reaches a minimal error rate and are possessed of the fastest convergence and the highest stabilities of results.

Acknowledgment

This work is partly supported by the National Science Council in Taiwan under grant NSC97-2622-E-151-008-CC2.

REFERENCES

The results of K-means, GA, KGA, PSO, NM-PSO, K-PSO, K-NM-PSO can be found in [7].

### 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**

<table>
<thead>
<tr>
<th>Data set</th>
<th>Criteria</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>K-means</td>
</tr>
<tr>
<td>Vowel</td>
<td>Average</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Std</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Best</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Error rates (%)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Std</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Best</td>
<td></td>
</tr>
<tr>
<td>Iris</td>
<td>Average</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Std</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Best</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Error rates (%)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Std</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Best</td>
<td></td>
</tr>
<tr>
<td>Crude Oil</td>
<td>Average</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Std</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Best</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Error rates (%)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Std</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Best</td>
<td></td>
</tr>
<tr>
<td>CMC</td>
<td>Average</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Std</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Best</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Error rates (%)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Std</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Best</td>
<td></td>
</tr>
<tr>
<td>Cancer</td>
<td>Average</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Std</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Best</td>
<td></td>
</tr>
<tr>
<td>Wine</td>
<td>Average</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Std</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Best</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Error rates (%)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Std</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Best</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Std</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Best</td>
<td></td>
</tr>
</tbody>
</table>

The results of K-means, GA, KGA, PSO, NM-PSO, K-PSO, K-NM-PSO can be found in [7].

### TABLE IV

**NUMBER OF FUNCTION EVALUATIONS OF EACH CLUSTERING ALGORITHM**

<table>
<thead>
<tr>
<th>Data set</th>
<th>K-means</th>
<th>PSO</th>
<th>NM-PSO</th>
<th>K-PSO</th>
<th>K-NM-PSO</th>
<th>GaussPSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vowel</td>
<td>180</td>
<td>16290</td>
<td>10501</td>
<td>15133</td>
<td>9291</td>
<td>9774</td>
</tr>
<tr>
<td>Iris</td>
<td>120</td>
<td>7260</td>
<td>4836</td>
<td>6906</td>
<td>4556</td>
<td>4356</td>
</tr>
<tr>
<td>Crude Oil</td>
<td>150</td>
<td>11325</td>
<td>7394</td>
<td>10807</td>
<td>7057</td>
<td>6795</td>
</tr>
<tr>
<td>CMC</td>
<td>270</td>
<td>36585</td>
<td>23027</td>
<td>34843</td>
<td>21597</td>
<td>21951</td>
</tr>
<tr>
<td>Cancer</td>
<td>180</td>
<td>16290</td>
<td>10485</td>
<td>15756</td>
<td>10449</td>
<td>9774</td>
</tr>
<tr>
<td>Wine</td>
<td>390</td>
<td>73245</td>
<td>47309</td>
<td>74305</td>
<td>46459</td>
<td>45747</td>
</tr>
<tr>
<td>Average</td>
<td>215</td>
<td>26833</td>
<td>17259</td>
<td>26292</td>
<td>16519</td>
<td>16400</td>
</tr>
</tbody>
</table>

The results of K-means, PSO, NM-PSO, K-PSO, K-NM-PSO can be found in [7].