Linearly Decreasing Weight Particle Swarm Optimization with Accelerated Strategy for Data Clustering

Cheng-Hong Yang, *Member, IAENG,* Chih-Jen Hsiao and Li-Yeh Chuang

***Abstract*— Clustering technique helps effectively to simplify the complexity of a large amount of data and identify the structure between data to data. It is a common technique for statistical data analysis that is used in many fields, e.g., machine learning, data mining, pattern recognition, image analysis, and bioinformatics. The distribution of information can be in different sizes or shapes. An improved technique combines the linearly decreasing weight particle swarm optimization (LDWPSO) and an acceleration strategy is proposed in this paper. The accelerated linearly decreasing weight particle swarm optimization (ALDWPSO) searches for cluster centers in an arbitrary data set and identifies global optima effectively. ALDWPSO is tested through six experimental data sets, and its performance is compared with the performance of PSO, NM-PSO, K-PSO, K-NM-PSO, LDWPSO and K-means clustering methods. The results indicate that ALDWPSO is a robust and suitable method for solving the clustering problems.**

***Index Terms*—data clustering, linearly decreasing weight, particle swarm optimization.**

I. Introduction

There are two primary categories in machine learning, i.e., the supervised learning and unsupervised learning; they attract many researchers with the most attention in this field. Clustering analysis is a typical and very popular unsupervised learning technique. Clustering analysis is the process of grouping a set of objects into clusters so that these objects within a cluster are similar to each other, but they are dissimilar to objects in other clusters [1] [2] [3]. When using for a set of objects, clustering analysis helps to identify some inherent structures that presents in the objects. The purpose of cluster analysis is to classify the clusters into subsets. In this context, each subset and its particular problem have certain meanings. More specifically, a set of patterns usually are vectors in a multi-dimensional space that are grouped into some clusters. When the number of clusters, *K*, is known as a priori, clustering may be formulated as the distribution of *n* objects in an *N*-dimensional space among *K* groups in such a way that the objects in the same group are more similar than those in different groups [4]. This involves the minimization of some extrinsic optimization criteria. The well-known of K-means [5] algorithm is successfully applied to many of practical clustering problems that suffer from several drawbacks due to its choice of initializations. The objective function of the K-means is not convex and it may contain many local minima. In recent years, many clustering algorithms based on evolutionary computing such as genetic algorithms [6] [7] and particle swarm optimization [8] are being introduced. Genetic algorithms are typically set out with some candidate solutions to the optimization problem. The candidates evolve toward a better solution through selection, crossover and mutation. Particle swarm optimization (PSO) is a population-based algorithm [9]. It simulates the behavior of natural occurring swarm, e.g., a self-evolving system is likely similar to a school fish. It automatically searches the optimum solution in the search space, but the involved search process is not random. Depending on the nature of different problems, a fitness function decides the best way to conduct the search. The PSO algorithm is rapidly becoming more popular, also it is applied in neural network optimization [10], data clustering [11] [12], engineering design [13], etc. Although the evolutionary computation techniques are eventually allocating the desired solution, the practical use of these techniques in solving complex optimization problems are severely limited by the high computational cost and the slow convergence rate.

People are using PSO to clustering data of a multi-dimensional space and obtaining for the outstanding results. However, the rate of convergence is insufficient when it searches global optima [14]. A linearly decreasing weight particle swarm optimization (LDWPSO) in a linear decreasing inertia factor is introduced by the velocity. It is an updated equation of the original PSO [15] which is proposed in this study, in order to solve this problem. The performance of LDWPSO is significantly improved over the original PSO because LDWPSO effectively balances out the global and local search abilities from the swarm. The accelerated linearly decreasing weight particle swarm optimization (ALDWPSO) algorithm is adapted by the cluster arbitrary data for evolving the appropriate cluster centers in an attempt to optimize a given clustering metric. The results of the experimental studies on a variety of data sets are taken from several real-life situations that demonstrate ALDWPSO is superior to the K-means, PSO, LDWPSO, K-PSO, and K-NM-PSO algorithms.

The work is organized as follows. In Section 2, the PSO algorithm, linearly decreasing weight and ALDWPSO

C. H. Yang is with the Department of Electronic Engineering, National Kaohsiung University of Applied Sciences, Kaohsiung 80778, Taiwan and Department of Network Systems, Toko University, Chiayi 61363, Taiwan (corresponding author to provide phone: +886-7-3814526ext5639; fax: +886-7-3836844; e-mail: chyang@cc.kuas.edu.tw).

C. J. Hsiao is with the Electronic Engineering Department, National Kaohsiung University of Applied Sciences, Kaoshiung 80778, Taiwan (e-mail: 1097305142@cc.kuas.edu.tw).

L. Y. Chuang is with the Chemical Engineering Department, I-Shou University, Kaoshiung 84001, Taiwan (e-mail: chuang@isu.edu.tw).

clustering are described. In Section 3, experimental results and a discussion are provided. Finally, the conclusion is offered in Section 4.

II. METHODS

A. Particle Swarm Optimization

The PSO algorithm, developed by Kennedy and Eberhart in 1995 [9], is an evolution-based computation method that performs robust and efficient optimization. In the original PSO, each particle is analogous to an individual fish in a school of fish. It is a population-based optimization technique, where a population is called a swarm. A swarm consists of N particles that are moving around in a D-dimensional search space.

The position of the i_{th} particle is represented by $x_i = (x_{i1}, x_{i2}, \ldots, x_{i\text{D}})$. The velocity for the i_{th} particle is written as $v_i = (v_{i1}, v_{i2}, \ldots, v_{i\text{D}})$. The positions and velocities of these particles are confined within $[X_{\min}, X_{\max}]^D$ and $[V_{\min}, V_{\max}]^D$, respectively. Each particle in the swarm evolves simultaneously by making good use of its own memory. In addition to that, knowledge from the neighboring particles must be considered to find the best solution.

The best encountered position of the i_{th} particle, denoted as $pbest_i$, is represented by $p_i = (p_{i1}, p_{i2},..., p_{i\text{D}})$. The best value of the all individual $pbest_i$ values are denoted as the global best position $g = (g_1, g_2, \ldots, g_{\text{D}})$ called *gbest*. The PSO process is initialized with a population of random particles, and the algorithm then executes a search for optimal solutions through continuous updating generations. At each time of the generation, the position and velocity of the i_{th} particle are updated by $pbest_i$ and *gbest* of the population. The updated equations are formulated as follow:

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

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

where r_1 and r_2 are the random numbers between (0, 1), and c_1 and c_2 are the acceleration which constants the control of how far a particle moves in a single generation. Velocities v_{id}^{new} and v_{id}^{old} denote the velocities of the new and old particles respectively. x_{id}^{old} is the current particle position, and x_{id}^{new} is the new updated particle position. Eberhart *et al.* [16] [17] suggested values of $c_1 = c_2 = 2$. The inertia weight w controls the previous velocity of particle. It is defined in Eq. (3).

$$w = 0.5 + \frac{rand}{2.0} \quad (3)$$

In Eq. (3), *rand* is a random generated number between zero to one.

The pseudo-code of the PSO process is shown as Fig. 1.

PSO pseudo-code

```
01: begin
02:   Randomly initialize 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 the number of particles
06:       Find pbest
07:       Find gbest
08:       for d = 1 to the number of dimension of particle
09:         update the position of particles by Eq. 1 and Eq. 2
10:       next d
11:     next n
12:     update the inertia weight value with Eq. 3
13:   next generation until the stopping criterion is met
14: end
```

Fig. 1. Pseudo-code of PSO

B. Linearly Decreasing Weight Particle Swarm Optimization

Shi and Eberhart proposed a linearly decreasing weight particle swarm optimization (LDWPSO) of which, a linearly decreasing inertia factor was introduced into the velocity of the updated equation from the original PSO [15]. The performance of LDWPSO is significantly improved over the original PSO because LDWPSO balances out the global and local search abilities of the swarm effectively. In LDWPSO, w_{LDW} is the inertia weight which linearly decreases from 0.9 to 0.4 through the search process [15]. The equation for the linearly decreased weight is written as:

$$w_{LDW} = \left(w_{\max} - w_{\min}\right) \times \frac{Iteration_{\max} - Iteration_i}{Iteration_{\max}} + w_{\min} \quad (4)$$

In Eq. (4), w_{max} is 0.9, w_{min} is 0.4 and $Iteration_{max}$ is the maximum number of the allowed iterations. The velocity of the updated equation for LDWPSO is formulated as:

$$v_{id}^{new} = w_{LDW} \times v_{id}^{old} + c_1 \times r_1 \times \left(pbest_{id} - x_{id}^{old}\right) + c_2 \times r_2 \times \left(gbest_d - x_{id}^{old}\right) \quad (5)$$

The pseudo-code of the LDWPSO process is shown as Fig. 2.

LDWPSO pseudo-code

```
01: begin
02:   Randomly initialize 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 the number of particles
06:       Find pbest
07:       Find gbest
08:       for d = 1 to the number of dimensions of particle
09:         update the position of particles by Eq. 5 and Eq. 2
10:       next d
11:     next n
12:     update the inertia weight value with Eq. 4
13:   next generation until stopping criterion is met
14: end
```

Fig. 2. Pseudo-code of LDWPSO

C. ALDWPSO Clustering

Previous applications that are used as PSO to solve practical clustering problems show some promising results, however, the slow convergence rate and the lack of global search ability of PSO still requires to be improved. We proposed a combination of a linearly decreasing weight and an acceleration strategy to improve the performance of PSO. ALDWPSO consists of four major processes; they are the encoding, the initialization of the particle, the acceleration strategy, the velocity and position update, and the fitness evaluation.

The ALDWPSO procedure for data clustering is described as below:

Step1). Initial population and encoding**:** 3*N* particle is randomly generated. All particles in the solution space are randomly generated with an individual position and velocity. Initialize population of particle with a random position x ($x \in \{x_1, x_2, \ldots, x_n\}$) and the velocity for the i_{th} particle is written as v ($v \in \{v_1, v_2, \ldots, v_n\}$), where n is the number of particle. Every particle contains a center position for each cluster. *N* is computed as follow:

$$N = K \times d \tag{6}$$

where *d* is the data set dimension and *K* is the anticipated number of clusters.

A possible encoding of a particle for a two-dimensional problem with three clusters is illustrated in Fig. 3. The three cluster centers are represented by this particle that are (1.5, 2.7), (3, 4.5), and (-5, -6).

Step2). Acceleration strategy: In the initialization stage, a portion of the population is used to accelerate the convergence rate of the particles. A portion of particles are being set after several experiments. The distances between data vectors within a cluster and the center of the cluster are defined in Eq. (7). The acceleration strategy recalculates the cluster center vectors by using Eq. (8). It produces the mean centers. The mean centers replace the original centers. The new position of the particle is given by:

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

$$z_j = \frac{1}{n_j} \sum_{\forall x_p \in c_j} x_p \tag{8}$$

where z_j denotes the center vector of cluster j, x_p denotes the p^{th} data vector, the subscript of d is the number of features for each center vector, n_j is the number of data vectors in cluster j and C_j is the subset of data vectors that form cluster j.

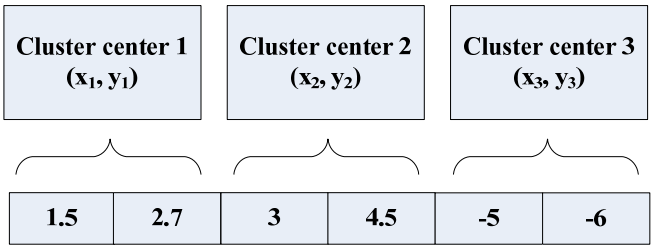

Fig. 3. Encoding of a single particle in PSO

The acceleration strategy is used for a portion of the population in the initialized stage. Although this procedure helps the convergence speed of PSO in solving clustering problems, however, the inappropriate setting of the ratio can lead to undesired local optima situation. Thus, the influence is caused by different ratios on clustering performance that will need to analyze further. This section presents the result of experiments that are conducted with different settings and sizes of particles for applying the acceleration strategy (*P* represents the ratio of population size). In the experiments, six UCI datasets are used to validate the effects of different *P* values and the values are set as 1/10, 1/9, … 1, the results are used to decide what value can produce the optimal solutions. Fig. 4 shows fitness values gathered from experiments with different settings of ratios that conduct on the six UCI datasets by the PSO algorithm that is equipped with the accelerated strategy. For each result, the value is the average over 20 independent simulation runs.

It is shown in Figs. 2; the performance behaviors of *P* values on each datasets are quite similar. The fitness values usually get better when the *P* values increase. However, in most situations, *P* values in the range of 1/4 to 1/2 often result in optimal fitness. When the *P* value leaves the optimal region and approaches to 1, the fitness values get worse as shown in the figure. These results tell us that *P* values in the rage of 1/4 and 1/2 make the acceleration strategy of a promising complement to the PSO algorithm. By analyzing the experimental results, all of the best fitness values are obtained when *P* values are set at 1/3 except for the Vowel dataset, where it is achieved when *P* is 1/2, thus we decided to set the *P* values as 1/3.

Step3). Group the data vectors for every particle: The data vectors are grouped into *K* clusters on the basis of the Euclidean distance as the similar measurement. Each particle maintains a matrix $x_i = (C_1, C_2, \ldots, C_j, .., C_K)$ where C_j represents the *i*th cluster centroid vector and *K* is the number of clusters. For each data vector, it is assigned to the cluster with the shortest distance. The distance is calculated as the length between the data vector and the centroid vector of the respective cluster, the calculation is described in Eq. (7).

Step4). Fitness evaluation: The fitness value of each particle is computed by the following fitness function.

$$\text{fitness} = \sum \left\| X_j - Z_i \right\|, \ i = 1, ..., K, \ j = 1, ..., n \tag{9}$$

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

Step5). 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.

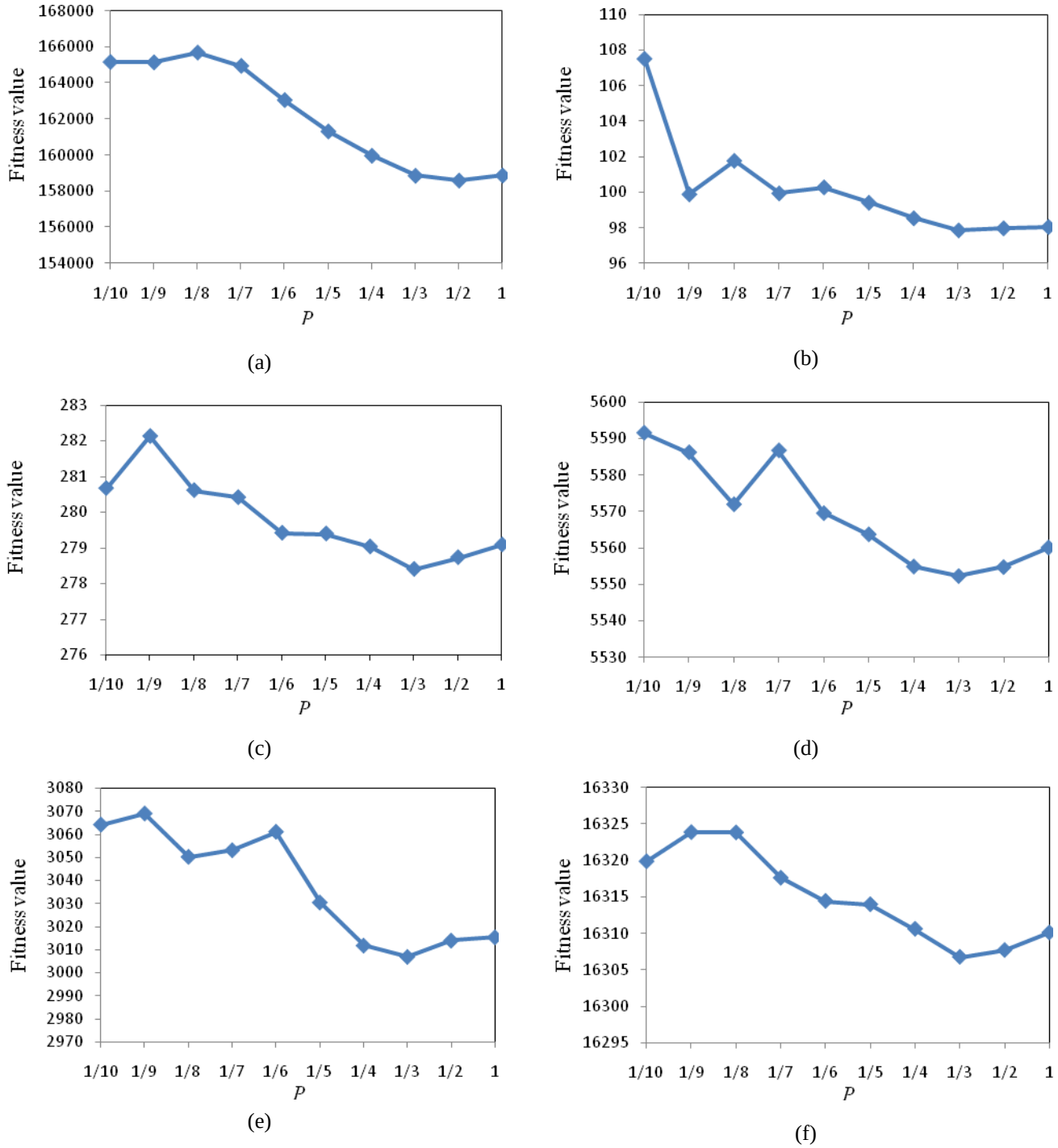

Fig. 4. Fitness values of clustering results are produced by PSO algorithm with acceleration strategy over different *P* values ranging from 1/10 to 1 for the data sets as (a) Vowel, (b) Iris, (c) Crude oil, (d) CMC, (e) Cancer, and (f) Wine.

Step6). Velocity and position update: The particles are moving through each iteration in the search space. The parameters w_{LDW} are modified by the linearly decreasing weight based on Eq. **(4)**. The particles' velocities and positions are dynamically updated by Eq. (5) and Eq. (2).

Step7). Repeat steps 3-6 until the termination condition is met.

The flowchart of ALDWPSO is shown in Fig. 5.

III. Experimental Results and Discussions

A. Data sets

To validate our method, six experimental data sets, i.e., Vowel, Iris, Crude oil, CMC, Cancer, and Wine are used. 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/.

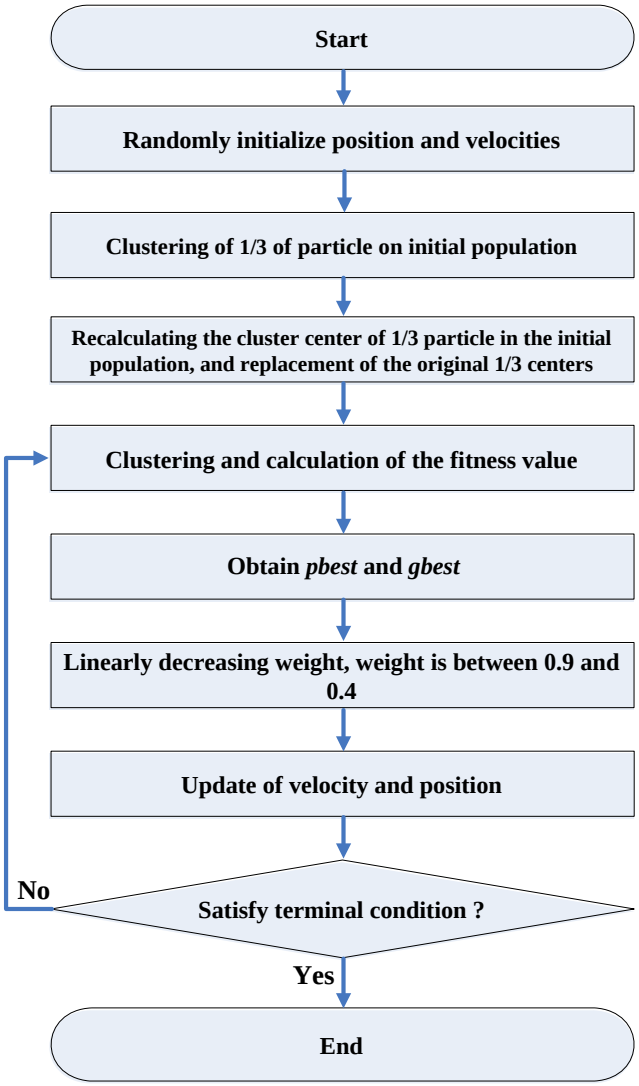

Fig. 5. Flowchart of the ALDWPSO clustering algorithm

Table 1 summarizes the characteristics of these data sets. Given 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 is 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. The data set has 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 from four features in each of sepal length, sepal width, petal length, and petal width were collected.

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

(4) Contraceptive Method Choice (denoted CMC with n = 1473, d = 9, k = 3). This data set is a subset of the 1987 National Indonesia Contraceptive Prevalence Survey. The samples consist of the married women who either not pregnant or not sure of their pregnancy at the time of interviews were conducted. The problem is to predict 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 uniform ity, marginal adhesion, single epithelial cell size, bare nuclei, bland chromatin, normal nucleoli, and mitoses. There are two categories in the data malignant (444 objects) and benign (239 objects) tumors.

(6) The Wine data set (n = 178, d = 13, k = 3) consists of 178 objects characterized by 13 features, namely alcohol, malic acid, ash content, alcalinity of ash, concentration of magnesium, total phenols, flavanoids, nonflavanoid phenols, proanthocyanins, color intensity, hue, and OD280/OD315 of diluted wines and pralines. The results 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).

B. Results and discussion

Results from ALDWPSO were compared with those from other popular methods, i.e., K-means, PSO, NM-PSO, K-PSO, K-NM-PSO and LDWPSO, to demonstrate its capability of data clustering. The quality of the respective clustering was also compared, where the quality was measured by the following two criteria:

1. The sum of the intra-cluster distances, i.e. the distances between data vectors within a cluster and the centroid of the cluster, are defined in Eq. (7). A higher quality of clustering is indicated if the sum is relatively small.

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

$$\text{error rate} = \left(\sum_{i=1}^{n} (if \quad (A_i = B_i) \quad then \quad 0 \quad else \quad 1) \div n \right) \times 100 \qquad (10)$$

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 2 an example is shown by the two data points (2, 6) and (1, 7) are out of clusters, 1 and 2 are misplaced and the error rate is 2/5, i.e., 40%.

The reported results are the averages of 20 runs of simulation, the details are given below. The algorithms are implemented through Java. For each run, $10 \times N$ iterations are carried out by each of the six data sets in every algorithm when solving an N-dimensional problem. The criterion $10 \times$

N is adopted in many previous experiments with a great success in terms of its effectiveness [14].

Table 3 summarizes the intra-cluster distances obtain from the seven clustering algorithms from the above data sets. The reported values are the averages of the sums from intra-cluster distances over 20 runs simulations with standard deviations that are given in parentheses to indicate the range of values that the algorithms span and the best fitness solution are from the 20 simulations. For all of the experimental data sets, ALDWPSO turns out to be the most outstanding method than the other fives. In addition, according to ALDWPSO result, the average values and the best solution show the least differentiation and a smaller standard deviation. In terms of the best distance for PSO, NM-PSO, K-PSO, K-NM-PSO and LDWPSO all have a larger standard deviation than the ALDWPSO does, even though they may achieve a global optimum. This means that the PSO, NM-PSO, K-PSO, K-NM-PSO and LDWPSO are weaker for the search tools of global optima than ALDWPSO if all methods are executed just once. ALDWPSO is more effective to find the global optimum solutions than the other five methods.

Table 1. Characteristics and the used of 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)

Table 2. Error rate calculations

I	Data point	A_i	B_i	Not misplaced (0)/Misplaced (1)
1	(2, 6)	2	1	1
2	(6, 3)	2	2	0
3	(1, 7)	2	1	1
4	(5, 4)	1	1	0
5	(8, 7)	1	1	0
Number of misplaced point：2				

Table 3. Comparison of intra-cluster distances for the seven clustering algorithms

Data set	Criteria	K-means	PSO	NM-PSO	K-PSO	K-NM-PSO	LDWPSO	ALDWPSO
Vowel	Average	159242.87	168477.00	151983.91	149375.70	149141.40	152392.6	**148985.50**
	(Std)	(916)	(3715.73)	(4386.43)	(155.56)	(120.38)	(4935.466)	(30.67)
	Best	149422.26	163882.00	149240.02	149206.10	149005.00	149041.3	148967.20
Iris	Average	106.05	103.51	100.72	96.76	96.67	96.67	**96.66**
	(Std)	(14.11)	(9.69)	(5.82)	(0.07)	(0.008)	(0.03)	(0.0009)
	Best	97.33	96.66	96.66	96.66	96.66	96.66	96.66
Crude Oil	Average	287.36	285.51	277.59	277.77	277.29	**277.24**	**277.24**
	(Std)	(25.41)	(10.31)	(0.37)	(0.33)	(0.095)	(0.043)	(0.039)
	Best	279.20	279.07	277.19	277.45	277.15	277.21	277.21
CMC	Average	5693.60	5734.20	5563.40	5532.90	5532.70	**5532.18**	**5532.18**
	(Std)	(473.14)	(289.00)	(30.27)	(0.09)	(0.23)	(9.5E-05)	(2E-06)
	Best	5542.20	5538.50	5537.30	5532.88	5532.40	5532.18	5532.18
Cancer	Average	2988.30	3334.60	2977.70	2965.80	2964.70	**2964.39**	**2964.39**
	(Std)	(0.46)	(357.66)	(13.73)	(1.63)	(0.15)	(0.0001)	(6.8E-06)
	Best	2987	2976.30	2965.59	2964.50	2964.50	2964.39	2964.39
Wine	Average	18061.00	16311.00	16303.00	16294.00	16293.00	16292.68	**16292.38**
	(Std)	(793.21)	(22.98)	(4.28)	(1.70)	(0.46)	(0.64)	(0.24)
	Best	16555.68	16294.00	16292.00	16292.00	16292.00	16292.19	16292.19

Legend : Clustering results over 20 runs of the seven different algorithms for six data sets; the best average values are indicated in bold type. The results of K-means, PSO, NM-PSO, K-PSO, K-NM-PSO can be found in [14].

Table 4. Comparison of error rates for the seven clustering algorithms

Data set	Criteria	K-means (%)	PSO (%)	NM-PSO (%)	K-PSO (%)	K-NM-PSO (%)	LDWPSO (%)	ALDWPSO (%)
Vowel	Average	44.26	44.65	41.96	42.24	41.94	42.25	**41.83**
	(Std)	(2.15)	(2.55)	(0.98)	(0.95)	(0.95)	(1.47)	(0.32)
	Best	42.02	41.45	40.07	40.64	40.64	40.18	40.87
Iris	Average	17.80	12.53	11.13	10.20	10.07	10.13	**10.00**
	(Std)	(10.72)	(5.38)	(3.02)	(0.32)	(0.21)	(0.27)	(0.00)
	Best	10.67	10.00	8.00	10.00	10.00	10.00	10.00
Crude Oil	Average	24.46	24.64	24.29	24.29	**23.93**	26.52	26.34
	(Std)	(1.21)	(1.73)	(0.75)	(0.92)	(0.72)	(0.66)	(0.79)
	Best	23.21	23.21	23.21	23.21	23.21	25.00	25.00
CMC	Average	54.49	54.41	54.47	**54.38**	**54.38**	**54.38**	**54.38**
	(Std)	(0.04)	(0.13)	(0.06)	(0.00)	(0.054)	(0.00)	(0.00)
	Best	54.45	54.24	54.38	54.38	54.31	54.38	54.38
Cancer	Average	4.08	5.11	4.28	3.66	3.66	**3.51**	**3.51**
	(Std)	(0.46)	(1.32)	(1.10)	(0.00)	(0.00)	(9.1E-16)	(9.1E-16)
	Best	3.95	3.66	3.66	3.66	3.66	3.51	3.51
Wine	Average	31.12	28.71	28.48	28.48	28.37	28.51	**28.31**
	(Std)	(0.71)	(0.27)	(0.27)	(0.40)	(0.27)	(0.40)	(0.28)
	Best	29.78	28.09	28.09	28.09	28.09	28.09	28.08

Legend : Clustering results over 20 runs of the seven different algorithms for six data sets; the best average values are indicated in bold type. The results of K-means, PSO, NM-PSO, K-PSO, K-NM-PSO can be found in [14].

Table 5. The number of function evaluations for each clustering algorithm

Data set	K-means	PSO	NM-PSO	K-PSO	K-NM-PSO	LDWPSO	ALDWPSO
Vowel	180	16,290	10,501	15,133	9,291	16,290	9,720
Iris	120	7,260	4,836	6,906	4,556	7,260	4,320
Crude Oil	150	11,325	7,394	10,807	7,057	11,325	6,750
CMC	270	36,585	23,027	34,843	21,597	36,585	21,870
Cancer	180	16,290	10,485	15,756	10,149	16,290	9,720
Wine	390	73,245	47,309	74,305	46,459	73,245	45,630
Average	**215**	**26,833**	**17,259**	**26,292**	**16,519**	**26,833**	**16,335**

Legend : The best average values are indicated in bold type. The results for K-means, PSO, NM-PSO, K-PSO, K-NM-PSO can be found in [14].

Table 4 shows the mean error rates, standard deviations, and the best solution of the error rates from the 20 simulations. For all the real life data sets except Crude Oil, ALDWPSO are exhibited significantly smaller mean as the standard deviation when comparing to K-means, PSO, NM-PSO, K-PSO, K-NM-PSO and LDWPSO. Again, ALDWPSO is superior to the other five methods with respect to the intra-cluster distance. However, it does not compare with the other methods for the Vowel, Iris, Crude Oil, and CMC data sets in terms of the best error rate as there has no absolute correlation between the intra-cluster distance and the error rate [14].

Table 5 lists out the numbers of objective function of evaluations that require for the five methods after $10 \times N$ iterations. The population sizes of PSO, K-PSO and LDWPSO are $5N$. The population sizes of NM-PSO and K-NM-PSO are $3N+1$ and ALDWPSO is $3N$. The population size of ALDWPSO is smaller than the population of the other algorithms. This results in a lower computational cost of ALDWPSO. An average of all data sets, the K-means algorithm needs the fewest function of evaluation. However, its solution is less than satisfactory as the clustering process tends to be easily trapped in local optima (Table 3 and 4). ALDWPSO and LDWPSO need fewer functions of evaluations than PSO, NM-PSO, K-PSO, and K-NM-PSO; they produce better outcomes than the other methods. K-NM-PSO is a hybrid technique that combines the K-means algorithm, Nelder-Mead simplex search [18] and PSO. In a direct comparison of the performance for ALDWPSO is proved to be better than the performance of K-NM-PSO. The indicated simulation result produce by ALDWPSO is a useful and robust tool for data clustering problems. ALDWPSO needs less functional evaluation to converge to the global optima and it still maintains a smaller error rate.

IV. Conclusions

This paper, a novel method is called ALDWPSO is to solve the data clustering problems. We used the six public recognizable UCI data sets to investigate the performance through our experiments. ALDWPSO uses minimum intra-cluster distances as a metric to search robustly data cluster centers in a *N*-dimensional Euclidean space. Under the same metric, PSO, NM-PSO, K-PSO, and K-NM-PSOSO need more iterations to achieve a global optimum. The K-means algorithm is possible to get stuck in a local optimum depending on its choice of the initial cluster centers. The experimental results indicate that ALDWPSO reaches a minimal error rate that is faster than the other methods. Therefore, ALDWPSO reduces the computational cost. In the future, we will employ ALDWPSO to other clustering problem in bioinformatics. In addition, we will propose a hybrid algorithm through our method and different algorithm in order to enhance the clustering performance. This paper is from IMECS 2010 conference [19].

References

[1] J. Han, M. Kamber, and A. K. H. Tung, "Spatial clustering methods in data mining: A survey," In H. J. Miller & J. Han (Eds.). *Geographic data mining and knowledge discover*, London: Taylor & Francis, 2001.

[2] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys*, 1999, vol. 31, pp.264–323.

[3] L. Rokach, and O. Maimon, "Clustering methods in Maimon, O. and Rokach, L. (Eds.): Data Mining and Knowledge Discovery Handbook," Springer, New York. 2005, pp. 321–352.

[4] M. R. Anderberg, "Cluster Analysis for Application," *Academic Press*, New York, 1973.

[5] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," *In: Proceedings of the Fifth Berkeley Symp. Math. Stat. Prob.*, vol.1, 1967, pp. 281-297.

[6] C. A. Murthy, and N. Chowdhury, "In search of optimal cluaters using genetic algorithms," *Pattern Recognition Letters*. 1996, vol. 17, pp. 825–832.

[7] S. Bandyopadhyay, and U. Maulik, "An evolutionary technique based on K-means algorithm for optimal clustering in RN," *Information Science*, 2002, vol. 146, pp. 221–237.

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

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

[10] R. C. Eberhart, Y. Shi, "Evolving Artificial Neural Networks." *in Proceeding of International Conference on Neural Networks and Brain*, Beijing, P.R.C. 1998, pp. 5-13.

[11] D. W. Merwe, A. P. Engelbrecht, "Data Clustering Using Particle Swarm Optimization," *in Proceedings of IEEE Congress on Evolutionary Computation (CEC)*, Canberra, Australia. 2003, pp. 215-220.

[12] M. G. Omran, A. P. Engelbrecht, and A. Salman, "Particle Swarm Optimization Method for Image Clustering," *International Journal on Pattern Recognition and Artificial Intelligence 19*, 2005, pp. 297-321.

[13] Q. He, L. Wang, "An effective co-evolutionary particle swarm optimization for constrained engineering design problems," *Engineering Applications of Artificial Intelligence 20*, 2007, pp. 89-99.

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

[15] Y. Shi, and R.C. Eberhart, "Empirical study of particle swarm optimization," *Proc. of Congress on Evolutionary Computation*, Washington, DC, 6-9 July 1999, pp. 1945-1949.

[16] R. C. Eberhart, and Y. Shi, "Tracking and optimizing dynamic systems with particle swarms," *In Proceedings of the Congress on Evolutionary Computation*. 2001, pp. 94-97.

[17] X. Hu, and R. C. Eberhart. "Tracking dynamic systems with PSO: where's the cheese?," *In Proceedings of the Workshop on Particle Swarm Optimization*, 2001.

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

[19] C.H. Yang, C. J. Hsiao and L. Y. Chuang, Accelerated Linearly Decreasing Weight Particle Swarm Optimization for Data Clustering, Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2010, IMECS 2010, 17-19 March, 2010, Hong Kong, pp. 25-30.