An Efficient Differential Memetic Algorithm for Clustering Problem

Lei Jiang, Datong Xie

***Abstract*—Clustering is a popular topic in data analysis and pattern recognition research. This paper presents an efficient memetic algorithm for clustering tasks. It applies to the whole simulated annealing process rather than those popular methods, only using metropolis criterion, as the local search mechanism to refine the differential evolutionary for improving the accuracy and robustness. The results show that this algorithm performs better than several existing methods in terms of clustering accuracy and efficiency in the majority of the three synthetic and four real life data sets used in this study. Moreover, the presented algorithm is more robust, flexible and not sensitive to the initial value in the unbalanced, overlapped and noisy data sets.**

***Index Terms*—differential evolutionary; memetic algorithm; clustering; simulated annealing**

I. Introduction

CLUSTERING is an important data analysis and pattern recognition technique which is to group similar objects into classes or clusters. This means that the objects in the same cluster share a high degree of similarity while dissimilar objects are in separate clusters. It is commonly applied to artificial intelligence, bioinformatics, biology, computer vision, city planning, data mining, intrusion detection, image segmentation, information retrieval, machine learning, marketing, object recognition, pattern recognition and web service discovery [1]–[6]

However, conventional clustering approaches have some shortcomings such as sensitivity to initial value, slowness of the convergence and preset classed in large scale data set etc. Moreover, with the growth of the data, clustering problems become more and more complex. It is often quite difficult to yield satisfactory results with one strategy or algorithm [7] [8]. Furthermore, according to [9], many real-world clustering problems involve inaccurate, noisy, discrete and complex data. Thus, robust and flexible methods are needed. For this reason, evolutionary algorithms (EA) are introduced to solve clustering problems as a promising direction. There are two main strategies in clustering research. First, the evolutionary algorithm combined with other algorithms such as Kmeans and FCM searches the solution globally. Second, the evolutionary algorithm generates the candidate solution from global search and the other algorithms that refines the solution from local search. This combination of evolutionary algorithms with local search is named memetic algorithms (MAs). Generally, compared with traditional EAs, MAs are more efficient (i.e., requiring orders of magnitude fewer evaluations to find optima) and more effective (i.e., identifying higher quality solutions) with respect to clustering problems. It can be known from the work of [10] that despite the impressive success achieved by some hybrid algorithms in the clustering analysis, the problem remains an open issue.

The reasoning of our research is based on the following three aspects. Firstly, there is a lack of the hybrid algorithm of simulated annealing (SA) with the EA under the framework of MA. The current methods mainly use SA as a heuristic strategy in some process of EA, and participate in the global search to get the final solution. Secondly, the existing methods of combined SA with DE are all mainly using the partial process of simulated annealing, i.e. they only use metropolis criterion one time to improve the selection period of DE at each iteration. Finally, the isothermal change in each annealing stage which is very important in improving the quality of the original solution by selecting with metropolis criteria has not yet been taken into consideration in the existing algorithms. The main contribution of this work is that an efficient differential memetic algorithm (mSADE) which employed the simulated annealing as a local search mechanism for the clustering problem was presented. Especially, the algorithm performs better while handling the robust, unbalance and overlapped datasets and can get a more accurate result than 9 other methods.

The rest of this paper is as follows. Section 2 is shown the related work. Section 3 presents the related techniques such as simulated annealing and the proposed algorithm. Section 4 details the experiments, and section 5 indicates the results and discussion. Finally, section 6 summarizes contents of this paper.

II. Related work

One of the first studies of EA related to clustering problems was conducted by Deneubourg's [11], where a basic ant colony clustering algorithm was proposed. The particle swarm optimization was also applied to find the centroids of a user-specified number of clusters [12]. Paterlini et al. [13] reported that the efficiency of differential evolution is clear which are consistently superior in respect both to precision and robustness of the results for clustering problems.

To further improve the precision and increase robustness, many hybrid clustering approaches based on the combination of evolutionary algorithms have been introduced in recent years. For instance, Monmarche [14] proposed a new hybrid clustering algorithm, which applies the Kmeans algorithm as the deterministic and heuristic principles to combine with the stochastic and exploratory principles of an ant colony. This

Manuscript received March 28, 2017; revised August 11, 2017. This work was supported in part by the Key projects of Research Fund in Hunan Provincial Education Department,China (Grant No. 15A064), the Natural Science Foundation of Hunan Province, China (Grant No. 2016JJ2056), the Ministry of education of Humanities and Social Science project, China (Grant No. 17YJAZH032).

Lei Jiang is with the Key Laboratory of Knowledge Processing and Networked Manufacture, Hunan University of Science and Technology, Xiangtan, 411201, China e-mail: jleihn@hotmail.com.

Datong Xie is with the Department of Information Management Engineering, Fujian Commercial College.

algorithm clusters automatically without prior knowledge of the data to be classified and complex parameter settings. Kanade [15] presented a hybrid approach to determine the number of clusters automatically by using the ant systems with the classical fuzzy c-means algorithm (FCM). Saatchi [16] integrated the ACO and the Kmeans algorithm for image clustering problems. Shang [17] analyzed the effectiveness of the hybrid algorithm and proposed a novel approach which integrated ACO, Kmeans and simulated annealing for solving clustering problems. Dougan [18] proposed a new clustering method based on kernelized fuzzy c-means algorithm and a hybrid ant colony optimization for continuous domains. Huang [19] applied hybridization strategies for continuous ant colony optimization and particle swarm optimization to data clustering. Wang [20] proposed a hybrid algorithm based on gravitational search and particle swarm optimization algorithm which improved the accuracy of the solution.

Ye [21] presented a hybridization algorithm which integrated PSO and Kmeans to optimally cluster N data points into K clusters by automatically detecting the cluster centers of geometrical structure datasets. Cui [22] presented a hybrid document clustering algorithm which applied the PSO+Kmeans at four different text document datasets. Devi [23] presented a new hybrid algorithm based on particle swarm optimization and Kmeans algorithm to cluster data. Their results showed that the PSO+Kmeans algorithm can produce the most compact clustering solutions than other approaches. Niknam [24] presented an efficient hybrid evolutionary algorithm, called PSOSA, by combining particle swarm optimization and simulated annealing for optimal clustering N object into K clusters. The basic idea is to search around the global solution by SA and to increase the information exchange among particles using a mutation operator to escape local optima. Huang [19] proposed an efficient hybrid approach based on PSO and ACO for cluster analysis. Karthi [25] presented a novel neighborhood search assisted particle swarm optimization (NPSO) algorithm for data clustering problems. Garg [26] integrated particle swarm optimization (PSO) and genetic algorithm (GA) to solve constrained optimization problems.

Yarn [27] improved the differential evolution by updating the acceptance probability with the simulated annealing rule and named it SADE. Liu [28] presented a simulated annealing based differential evolution. The two methods both only used the metropolis criterion of SA to mix with DE. Das [29] presented AnDE which introduced simulated annealing as a stochastic selection mechanism. Kwedlo [30] combined differential evolution algorithm (DE) with the Kmeans procedure for clustering analysis. In this method, the Kmeans algorithm was used to refine each candidate solution obtained by mutation and crossover operators of DE. Thangavel [31] presented a novel hybrid PSOSA for biclustering of expression data. Vakil [32] proposed a differential memetic algorithm (DMA) which used a DBRS algorithm to do the local search. The results showed that the DMA had better performance in most of the cases.

III. Proposed algorithm

A. Simulated annealing

Simulated annealing (SA) is a significant algorithmic approach to avoid the local-trap problem. Kirkpatrick [33] noted that bringing a physical system into a low energy state (such as growing a crystal from a melted substance) is very similar to the process of finding an optimum solution of a combinatorial optimization problem in thermodynamics. It is well known that annealing is widely used to grow crystals, and Gidas [34] proposed the Metropolis Monte Carlo method to simulate the physical annealing process. For complex learning tasks, SA has a better chance to converge to a global optimum solution, which has been tested by Zhang [35].

Let $T_1 > T_2 > \cdots > T_k > \cdots$ be a sequence of monotonically decreasing temperatures, where T_1 is reasonably large and $\lim_{k\to\infty} T_k = 0$. SA works in the following procedure.

Algorithm 1 Simulated annealing algorithm

1: Initialize: Let $k = 0$. Set initial temperature $T_k = T_0$. Set isothermal change times $L_k = L$. Generate initial state $x = x_0$ randomly.
2: **repeat**
3: Isothermal change process:
4: **for** $t = 1$ to L_k **do**
5: Select new state x_{new} from the neighborhood space of the state x_i randomly.
6: According to the state of objective function $f(x)$, calculate $E = f(x_{new}) - f(x)$;
7: Apply the Metropolis criterion to accept the new state x_{new}:
8: **if** $E \leq 0$ **then**
9: $x_{i+1} = x_{new}$,
10: **else**
11: calculate the probability
$$p_{T_k} = \exp(-\frac{\Delta E}{T_k})$$
12: **end if**
13: generate a uniform random number r on the interval $(0, 1)$.
14: **if** $p_{T_k} \geq r$ **then**
15: $x_{i+1} = x_{new}$.
16: **else**
17: $x_{i+1} = x_i$.
18: **end if**
19: **end for**
20: Cooling process: Increase k to $k+1$. Use the cooling speed α to calculate
$$T_k = \alpha T_{k-1}, k = 1, 2, \cdots, K$$
21: **until** meeting the minimum temperature.

B. DE algorithm

Paterlini [13] first employed differential evolution algorithm into clustering analysis. They use floating point (real-coded) to encode representation. Here, medoids are used to specify the allocation of objects to clusters. That is, the ith individual genome $Z_i(t)$ of the population at time-step t is built by allocating each object to the cluster corresponding to its nearest medoid according to the Euclidean distance.

For each individual genome of the current population, DE randomly samples three other individuals, i.e., $Z_j(t)$, $Z_k(t)$, and $Z_l(t)$, from the same generation (with $j \neq k \neq l$). It

then calculates the difference of $Z_k(t)$ and $Z_l(t)$, scales it by multiplication with a parameter β (usually $\in [0,1]$), and creates a candidate offspring $Z_i'(t+1)$ by adding the result to $Z_j(t)$. Herein, it is not the entire candidate offspring is created in this way, rather, some are inherited from individual $Z_i(t)$, such that

$$Z_i'(t+1) = \begin{cases} Z_j(t) + \beta(Z_k(t) - Z_l(t)), \\ \qquad\qquad if\ rand(0,1) < Cr \\ Z_i(t), \qquad otherwise. \end{cases} \quad (1)$$

Where, $Cr \in [0,1]$ is the *crossover rate* of the algorithm. If the fitness of the new offspring is better, it replaces its parent in the next generation; otherwise, the parent is retained in the population, i.e.,

$$Z_i(t+1) = \begin{cases} Z_i'(t+1), & if\ f(Z_i'(t+1)) < f(Z_i(t)) \\ Z_i(t); & otherwise. \end{cases} \quad (2)$$

$f(\cdot)$ is the fitness function which is the smaller the better. The process is repeated for a fixed number of iterations.

C. Differential memetic algorithm with simulated anealling

The mSADE algorithm herein proposed combines the simulated annealing and integrates it within a DE framework. The mSADE is a simple memetic algorithm which is a very efficient possibility for clustering analysis. In order to get a better solution, we use two strategies to improve the clustering algorithm, of which one is to improve the chromosome regeneration (i.e. we increase population diversity in the early stage and enhance their exploitation ability later) and the other is to use the memetic strategy done the local search to refine the solution.

1) Chromosome initialize: Chromosome initialize: Let Z be an $n-$data set $Z = \{z_1, z_2, \cdots, z_n\}$ that is grouped into c clusters. In this proposed method, we select c samples as centroids without replacement from the data set Z randomly to build chromosome $chr^{(t)}$.

$$chr = \boxed{m_1}\boxed{m_2}\boxed{\cdots}\boxed{m_c}$$

2) Chromosome representation: The DE we used herein is significantly differently from the original one. In building candidate offspring (namely, chromosome representation), it uses a random factor to replace the constant parameter β of the equation 3, shown as follows:

$$\beta = 0.5 * (1 + rand(0,1)) \quad (3)$$

Where $rand(0,1) \in [0,1]$ is a uniformly distributed random number. This stochastic parameter amplifies the difference in the progresses and thus helps retain population diversity. Secondly, it is well known that a big crossover rate can be helped the exploration abilities in the whole search space at the beginning and a small crossover rate can enhance the exploitation abilities in the region of the suspected global optimum lies carefully. Therefore the Cr may be defined in the following equation:

$$Cr = (Cr_{max} - Cr_{min}) \cdot \frac{2}{1 + e^{iter-1}} \quad (4)$$

Where Cr_{max} and Cr_{min} are the maximum and minimum values of crossover rate Cr; $iter$ is the current generation number.

3) Local search: According to the definition of [36] and [37], "An MA is an EA that includes one or more local search phases within its evolutionary cycle". As such, we employ the SA to do the local search after the crossover phase of DE. The strategy implemented by SA consists of exploring the solution space starting from an arbitrary selected solution and generating a new one after the search. When a new solution is generated, its fitness is evaluated to accept or reject according to an accepted rule. That is to say, the next solution of the problem is generated by the better one of the DE and SA but not DE itself.

Algorithm 2 The mSADE algorithm

1: Initialize: Set the current iterations $t = 0$, initial temperature T_0 and the maximum number of iterations t_{max}, Population number P.
2: Build chromosome $chr_i^{(t)}, i = 0, \cdots, P-1$,randomly.
3: **repeat**
4: **for** $\{i = 0; i < P; i++\}$ **do**
5: for each sample, calculate its distance with the every centroids, m_i, and finds which class it will belong.
6: Generate the feasible solution $Solution_i^{(t)}$.
7: calculate $fitness(i)$.
8: SA apply the $Solution_i^{(t)}$ as the initial solution to do local optimal and get new $Solution_i^{(t')}$.
9: calculate $fitness(i)'$.
10: **if** $fitness(i)'$ is better than $fitness(i)$ **then**
11: Get the $chr_i^{t'}$ from $Solution_i^{(t')}$.
12: Use $chr_i^{t'}$ replace chr_i^t
13: **end if**
14: **end for**
15: Sample three chromosome from populaton,randomly.
16: Get the crossing rates Cr by using equation 4.
17: Uses the equation 3 and 1 to do the crossover.
18: Get the new chromosome $chr^{(t+1)}$.
19: t++
20: **until** $t = t_{max}$.

IV. Experiments

In this section, we compare the results obtained by the mSADE with: a) three evolutionary methods, PSO, SA, and ACDE (an improved DE clustering method) [38]. b) a classical approach, Kmeans, which is implemented in Weka 3 of [39]. c) four hybrid algorithms which combined the evolutionary algorithm and SA, one is now being widely used PSOSA, for example, kathpal [40] use PSOSA for achieving partitioning optimization in various network applications. Basu et al. [41] use PSOSA for the multidimensional function optimization problem. Other methods are algorithms of hybrid DE with SA. Namely, AnDE, SADE and SADEA. d) the latest efficient memetic algorithm, DMA. In these experiments, three artificial data sets and four real-world data sets are used to evaluate its performance in the noisy, unbalanced and overlapped circumstance. Meanwhile, the results gained from ACDE, PSO, SA, Kmeans, SADE, AnDE, SADEA, PSOSA and DMA under the assessment by CS and DB validity indices are compared. Then, the accuracy rate has been evaluated between the mSADE and the other

performing algorithms. Finally, the box plots are applied to conduct an inspection.

A. Descriptions datasets

1) Synthetic datasets: Here, in order to observe the ability of the proposed algorithm, three synthetic clustered datasets with a variety of structures were generated. The three datasets are referred to as D_1-D_3.

$D_1 \in \mathcal{R}^2$ is a noisy dataset, which consists of 150 samples and distributes in 3 clusters. There is no overlap between each cluster and the number of the cluster is 50. All the samples are generated from uniform distribution.

$D_2 \in \mathcal{R}^2$ is an unbalanced and overlapped dataset, which consists of 230 samples and allocates in 4 clusters. The number of each cluster is 60,40,30 and 100, respectively. All the samples are built from Gaussian distribution and each cluster is with different covariance matrices and means.

$D_3 \in \mathcal{R}^3$ is a relatively balanced and partially overlapped dataset, which consists of 250 samples and allocated in 5 clusters. Its first and second dimension are generated from a uniform distribution and the third dimension is built from a Gaussian distribution. The number of each cluster is 42,48,43,47 and 50.

All above mentioned datasets are shown in Fig 1.

2) Real world datasets: We employ 4 UCI repositories of machine learning databases: *Iris, Wine, Breast Cancer* and *Vehicle* which are introduced in the following paper.

Iris: This dataset consists of 150 samples with three classes: setosa, versicolor, and verginica. Each class has 50 objects with four attributes: sepal length, sepal width, petal length, and petal width.

Wine: The database contains 13 features of chemical analysis about wines. 178 samples are divided in 3 different cultivars. The number of each cultivar is 59,71 and 48.

Wisconsin Breast Cancer: The dataset contains 9 attributes and 699 samples. But, 16 samples have missing data, we use the rest 683 objects in our experiments. The objective is to classify each sample into benign (444 objects) or malignant tumors (239 objects) clearly.

Vehicle: The data were classified into 4 overlapping classes (212 objects, 218 objects, 217 objects and 199 objects). This database consists of 946 samples and 18 relevant features. 846 samples are used in our experiments, excluding the samples containing missing data.

B. Parameters settings

We adopted the best possible parameter settings which are recommended in literatures, and then an optimal set of parameters is chosen after considering many possibilities. The parameters are set in Table I.

C. Validity indices and fitness function

Since clustering is an unsupervised process where the data available are unlabeled, it is difficult to determine which clustering result is the best. Generally, data in a multi-dimensional space are coherent in the same group and are separated into different groups. Thus, the traditional approach to evalute the clustering quality is validity indices which take care of the clustering partitioning cohesion and separation.

TABLE I: The parameters of methods

Method	parameters
mSADE	$F = 0.65, C_r = 0.35, Pop\_size = 30, L_k = 5, P_0 = 0.5, P_{ls} = 0.95, T_0 = 100$
ACDE	$CR_{max} = 1.0, CR_{min} = 0.5, F = 0.65, C_r = 0.35, Pop\_size = 30$
SA	$\alpha = 0.98, T_0 = 50, L_k = 100, P_{ls} = 0.95$
PSO	$C_1, C_2 = 1.49, w = 0.72, Pop\_size = 10$
PSOSA	$C_1, C_2 = 1.49, w = 0.72, Pop\_size = 10, \alpha = 0.995, T_0 = 50, L_k = 5, P_0 = 0.5, P_{ls} = 0.95$
AnDE	$CR_{max} = 1.0, CR_{min} = 0.5, F = 0.8, C_r = 0.9, Pop\_size = 30, T_0 = 1000000$
SADE	$\alpha = 0.3, F = 0.80, C_r = 0.60, Pop\_size = 30$
SADEA	$\alpha = 0.3, F = 0.50, C_r = 0.30, Pop\_size = 30, T_0 = 1000000$
DMA	$\delta = 0.5, \alpha = 0.8, Pop\_size = 100$

From Pakhira [42] and Rizman [43], the widely used validity indices are DI, DB, CS, PBM, etc. Here, the two validity index used in the paper are introduced as follows:

1) DB index: This index is proposed by [44] which measure the relation between within-cluster scatter and inter-cluster separation. Lower DB index means a better clustering partition. It can be computed as the follows, respectively:

$$S_{i,q} = \left(\frac{1}{|C_i|} \sum_{x \in C_i} \{||x - z_i||_2^q\} \right)^{1/q} \tag{5}$$

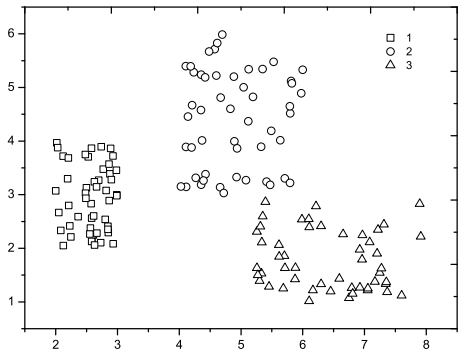
$$d_{ij,t} = \left(\sum_{s=1}^{p} |z_{is} - z_{js}|^t \right)^{1/t} = ||z_i - z_j||_t \tag{6}$$

$$\begin{cases} R_{i,qt} = \max\limits_{j, j \neq i} \left(\dfrac{S_{i,q} + S_{j,q}}{d_{ij,t}} \right) \\ DB(K) = \frac{1}{K} \sum\limits_{i=1}^{K} R_{i,qt} \end{cases} \tag{7}$$

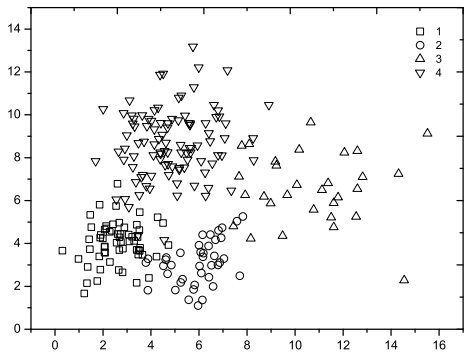
2) CS index: This measure is presented by Chou [45] which is a function of the ratio of the sum of within cluster scatter to between-cluster separation. According to Chou et al., it is more efficient in dealing with clusters of different densities and/or sizes than the other indices. And the smallest CS index indicates an optimal valid partition. Its defined as follows:

$$\begin{aligned} CS(K) &= \frac{\frac{1}{K} \sum\limits_{i=1}^{K} \left[\frac{1}{N_i} \sum\limits_{\vec{X}_i \in C_i} \max\limits_{\vec{X}_q \in C_i} \left\{ d(\vec{X}_i, \vec{X}_q) \right\} \right]}{\frac{1}{K} \sum\limits_{i=1}^{K} \left[\min\limits_{j \in K, j \neq i} \{ d(\vec{m}_i, \vec{m}_j) \} \right]} \\ &= \frac{\sum\limits_{i=1}^{K} \left[\frac{1}{N_i} \sum\limits_{\vec{X}_i \in C_i} \max\limits_{\vec{X}_q \in C_i} \left\{ d(\vec{X}_i, \vec{X}_q) \right\} \right]}{\sum\limits_{i=1}^{K} \left[\min\limits_{j \in K, j \neq i} \{ d(\vec{m}_i, \vec{m}_j) \} \right]} \end{aligned} \tag{8}$$

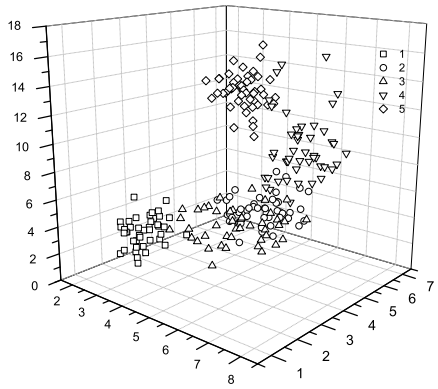
The validity index of CS is widely used as fitness functions for an evolutionary computational approach based clustering from the research of Xu [46]. Therefore, the CS index is introduced as the fitness function in this paper.



(a) D_1



(b) D_2



(c) D_3

Fig. 1: D_1 data set with 3 clusters; D_2 data set with 4 clusters; D_3 data set with 5 clusters. Each cluster is marked with a digital.

V. Results and discussion

By fairly comparing the performance of the mSADE algorithm with other state-of-the-art clustering approaches, each of them runs for a very long time over every synthetic and real world data set, until the number of FEs (fitness function evaluations) is equaled to 10^5. And each competitor clustering algorithm is run for 30 times, independently. Then, the CS and DB validity indices are used to evaluate the quality of these solutions. The value of mean and standard deviation are shown in Table II-III. At the same time, since the partition of these synthetic and real world datasets used in this paper is known in advance, the accuracy rate of the solutions gained with the clustering algorithms is also used to evaluate. Table IV illustrates the corresponding mean values and standard deviations of accuracy rate over 30 runs. Based on 30 solutions obtained from each algorithm with each dataset, the distribution of the results of all the evaluations of the final solutions is displayed. Then, we apply the box plots (Figure 2 to 4) to evaluate the consistency of these results.

A. Overall evaluation

In order to make an overall evaluation, all the results are ranked based on their evaluations given by the CS, DB and accuracy rate. The scores of each algorithm are calculated as follows:

$$score(A) = \sum_{i=1}^{D_{num}} rank_i(A) \tag{9}$$

Where the D_{num} is the number of the datasets. Since the score is the judgment from one or some aspect of the algorithms, the three validity methods are integrated in order to obtain a more comprehensive evaluation. Consequently, the comprehensive evaluation score of these algorithms is given by the follow equation:

$$score(A^{ce}) = \sum_{i=1}^{V_{num}} \sum_{i=1}^{D_{num}} rank_i(A) \tag{10}$$

Where the V_{num} is the number of the evaluating method. The lower the score is, the better the algorithm is.

As a result, the scores of three synthetic and four real datasets given in Tables V reveal that the method based upon the framework of MA, mSADE is very precise. It provides the optimum value and small standard deviation in comparison to those achieved by the other algorithms.

B. Discussion on the comparison of algorithms

Furthermore, the comparison is conducted for a more detailed evaluation in four levels: a) The first level is to compare with several typical evolution clustering algorithms

TABLE II: The results of solution evaluation with CS validity index (mean and standard deviation over 30 independent runs)

	Statistic	mSADE	ACDE	SA	PSO	Kmeans	PSOSA	AnDE	SADE	SADEA	DMA
	mean	9.979	20.408	57.716	43.381	10.385	45.260	17.437	12.355	10.003	10.120
D_1	Std dev	0.058	5.207	3.445	7.440	0	7.346	4.921	1.780	0.125	0.281
	rank	1	7	10	8	4	9	6	5	2	3
	mean	7.800	11.134	63.958	37.137	8.470	38.239	13.613	10.081	8.125	8.912
D_2	Std dev	0.073	1.138	4.070	7.814	0	5.096	2.519	1.146	0.470	0.461
	rank	1	6	10	8	3	9	7	5	2	4
	mean	4.099	6.694	38.517	22.055	4.082	22.925	6.789	5.639	4.295	4.487
D_3	Std dev	0.055	1.185	2.657	3.732	0	5.205	1.342	0.729	0.182	0.216
	rank	2	6	10	8	1	9	7	5	3	4
	mean	7.681	8.325	58.863	46.991	7.643	49.831	11.419	8.929	7.671	7.740
iris	Std dev	0.092	0.046	5.544	10.37	0	10.35	3.356	1.049	0.096	0.116
	rank	3	5	10	8	1	9	7	6	2	4
	mean	9.798	18.927	179.46	106.58	11.343	169.92	15.329	11.296	9.935	10.185
wine	Std dev	0.155	6.310	70.29	64.81	0	105.6	4.094	0.813	0.346	0.268
	rank	1	7	10	8	5	9	6	4	2	3
	mean	186.49	209.16	499.84	1214.2	71.866	1579.5	345.57	71.863	51.949	57.864
breast	Std dev	0.210	28.68	61.25	809.2	0	888.1	200.59	9.743	2.103	4.064
	rank	5	6	8	9	4	10	7	3	1	2
	mean	17.827	22.490	1070.8	288.24	33.714	274.21	32.590	23.202	17.803	19.136
viche	Std dev	0.053	2.699	328.1	143.1	0	130.5	7.290	4.501	0.022	0.714
	rank	2	4	10	9	7	8	6	5	1	3
	score	15	41	68	58	25	63	46	33	13	23

TABLE III: The results of solution evaluation with DB validity index (mean and standard deviation over 30 independent runs)

	Statistic	mSADE	ACDE	SA	PSO	Kmeans	PSOSA	AnDE	SADE	SADEA	DMA
	mean	2.749	1.847	69.330	25.406	2.643	31.134	4.562	3.231	2.891	2.811
D_1	Std dev	0.367	0.197	98.98	28.26	0	34.06	1.919	0.566	0.382	0.397
	rank	3	1	10	8	2	9	7	6	5	4
	mean	3.124	0.999	73.161	20.772	3.442	23.173	5.171	3.488	2.594	3.144
D_2	Std dev	1.015	0.171	36.27	9.380	0	14.66	1.456	1.264	1.198	1.007
	rank	3	1	10	8	5	9	7	6	2	4
	mean	2.325	1.831	62.318	18.463	2.773	21.770	3.694	3.445	2.342	2.670
D_3	Std dev	0.499	0.241	19.67	8.942	0	14.70	1.408	1.031	0.809	0.642
	rank	2	1	10	8	5	9	7	6	3	4
	mean	2.067	1.217	51.749	33.289	2.395	34.152	3.636	2.748	2.847	2.696
iris	Std dev	0.635	0.030	37.37	27.49	0	33.82	1.720	1.016	0.749	0.820
	rank	2	1	10	8	3	9	7	5	6	4
	mean	2.388	4.836	103.83	37.231	3.089	75.645	3.533	2.991	2.258	2.375
wine	Std dev	0.687	3.355	195.8	27.74	0	82.23	0.900	0.618	0.788	0.752
	rank	3	7	10	8	5	9	6	4	1	2
	mean	9.784	9.284	37.231	68.497	10.627	90.068	15.971	10.053	6.142	6.978
breast	Std dev	0.010	0.642	27.74	58.21	0	57.36	8.953	3.206	2.086	2.120
	rank	4	3	8	9	6	10	7	5	1	2
	mean	4.713	1.399	2.696	79.308	4.751	89.049	6.225	5.079	4.283	4.176
viche	Std dev	1.782	0.072	0.820	51.99	0	71.53	4.484	1.757	1.715	1.767
	rank	5	1	2	9	6	10	8	7	4	3
	score	22	15	60	58	32	65	49	39	22	23

TABLE V: The comprehensive evaluation score of the all algorithms

	mSADE	ACDE	SA	PSO	Kmeans	PSOSA	AnDE	SADE	SADEA	DMA
score	51	92	196	174	83	191	138	105	53	72

including the PSO, ACDE and SA. b) The second level is the classical method called Kmeans. c) The third level is the hybrid algorithms including PSOSA, AnDE, SADE and SADEA. d) The fourth level is the latest efficient memetic algorithm, DMA.

When comparing with the typical evolution clustering algorithms, it can be seen that the mSADE is obviously superior to the other methods under the CS evaluation in Table II. Also, Table III shows that the PSO and SA are inferior to the mSADE, but the ACDE has outstanding performance in DB evaluation. More importantly, in the view of accuracy rate of partition, the mSADE is obviously better than these evolutionary clustering algorithms in the seven datasets. Furthermore, as shown in Figure 2 to Figure 4, the mSADE is very cohesive over 30 independent runs. Based on the above depiction, the hybrid structure of DE and SA which is under the MA frame is more efficient than a single method. The single method is still not as good as the hybrid method although it does employ some tactics. For instance, the ACDE decreases the crossover rate accompanying time

TABLE IV: The accuracy rate (mean and standard deviation over 30 independent runs)

	Statistic	mSADE	ACDE	SA	PSO	Kmeans	PSOSA	AnDE	SADE	SADEA	DMA
	mean	0.987	0.556	0.414	0.550	0.953	0.530	0.765	0.892	0.985	0.974
D_1	Std dev	0	0.294	0.026	0.076	0	0.065	0.129	0.071	0.009	0.026
	rank	1	7	10	8	5	9	6	4	2	3
	mean	0.909	0.559	0.353	0.545	0.943	0.494	0.748	0.814	0.918	0.877
D_2	Std dev	0.020	0.204	0.040	0.122	0	0.068	0.103	0.103	0.021	0.050
	rank	3	7	10	8	1	9	6	5	2	4
	mean	0.879	0.762	0.261	0.461	0.900	0.459	0.712	0.767	0.864	0.839
D_3	Std dev	0.145	0.148	0.014	0.047	0	0.077	0.083	0.063	0.028	0.031
	rank	2	6	10	8	1	9	7	5	3	4
	mean	0.880	0.989	0.419	0.513	0.887	0.500	0.812	0.895	0.877	0.878
iris	Std dev	0	0.007	0.027	0.058	0	0.058	0.110	0.048	0.021	0.028
	rank	4	1	10	8	3	9	7	2	6	5
	mean	0.917	0.733	0.431	0.509	0.725	0.464	0.737	0.716	0.881	0.866
wine	Std dev	0.039	0.131	0.024	0.070	0	0.050	0.087	0.067	0.085	0.057
	rank	1	5	10	8	6	9	4	7	2	3
	mean	0.970	0.843	0.760	0.655	0.965	0.621	0.789	0.943	0.972	0.956
breast	Std dev	0.001	0.159	0.024	0.103	0	0.113	0.152	0.035	0.025	0.024
	rank	2	6	8	9	3	10	7	5	1	4
	mean	0.589	0.538	0.288	0.347	0.423	0.360	0.503	0.523	0.588	0.561
viche	Std dev	0.010	0.029	0.010	0.027	0	0.041	0.069	0.046	0.013	0.033
	rank	1	4	10	9	7	8	6	5	2	3
	score	14	36	68	58	26	63	43	33	18	26

linearly to promote the algorithm. Its accuracy rate, however, is superior to the mSADE in the iris dataset only.

Although great progress has been made in clustering, the classical method Kmeans has strong practicality to solve many problems. Consequently, we make a comparison between the Kmeans and mSADE. In Table II, for the seven datasets, the mSADE has the advantage on the CS validity index assessment in 4 of them. Moreover, as shown in Table III, the mSADE is superior to the Kmeans in 6 datasets by the evaluation of the DB validity index. As for the accuracy rate, the mSADE is in an obviously dominant position at the D_1, $wine$ and $viche$ can be observed. And in the case of $Wisconsin\ Breast\ cancer$, the mSADE is also a little superior to the Kmeans. Moreover, the difference is extremely significant statistically from Table III. For Kmeans, on the contrary, it takes advantage of the accuracy rate in D_2, D_3. In $iris$, the Kmeans is slightly better than the mSADE but the difference is also extremely significant statistically. Herein, results show that the Kmeans has good performance mainly for synthetic datasets. The possible reason is that these synthetic datasets are more convexly designed than real datasets. In other words, the mSADE is more suitable for noncomvexity datasets.

After comparing the methods which are not under the MA frame to hybrid of SA and evolutionary algorithm with the mSADE, it shows that the mSADE is overall better than the PSOSA, AnDE and SADE under the CS and DB validity indices and the accuracy rate evaluation. For the SADEA algorithm, the mSADE is better in the integral evaluation from Table 4. It is a slight advantage than the mSADE under the CS assessment (Table II) and has the same score in the judgement of the DB (Table III). However, regarding the most important thing of clustering partition - the accuracy rate, the mSADE is obviously better than the SADEA as shown in Table IV. In summary, the mSADE performs better than the others. One possible reason of this may be that on the whole solution space $\mathcal{S}$ each method has a search trajectory space $\mathcal{S}_i$ determined by the approach itself. The PSOSA, AnDE, SADE and SADEA have only used the SA as a procedure of PSO or DE. And its search trajectory is still in the $\mathcal{S}_{PSO}$ or $\mathcal{S}_{DE}$. Whereas, the mSADE algorithm applies the SA as an independent step to do local search after the procedure of DE in each iteration. The search trajectory space of mSADE $\mathcal{S}_{mSADE}$ has a greater chance than that of DE and PSO in this case. That is to say, $\mathcal{S}_{mSADE}$ maybe contains part of $\mathcal{S}_{SA}$. Namely, $\mathcal{S}_{mSADE} \geq \mathcal{S}_{DE}$ and $\mathcal{S}_{mSADE} \geq \mathcal{S}_{PSO}$.

The fact that mSADE is superior to the latest efficient memetic algorithm DMA is also noted. Results in Table II and Table III reveal the fact that the mSADE performs better in six datasets. In contrast, the DMA has done better only in one dataset in terms of the CS and DB evaluation. The results of mSADE are very good from Table IV. The accuracy rate of the DMA is close to that of the mSADE in the case of $iris$ only, meaning in the other datasets, the DMA is inferior to the mSADE. Consequently, it can be concluded that the memetic mechanism of mSADE is more efficient than the DMA. A possible reason for this is that the DMA gives its local neighbors solution still by the DEs operators, whereas the mSADE employs the simulated annealing method to create the local search solution. As a result, the search trajectory space of mSADE $\mathcal{S}_{mSADE}$ will be larger than the DMA's.

C. Discussion on the performance on various datasets

Due to further study on the performance of mSADE, we made an analysis on various datasets such as noise, balanced\unbalanced datasets and overlapped\partially-overlapped datasets.

For the balanced datasets D_1 and $iris$, the mSADE has the best assessment in CS and the second best in DB evaluation. Meanwhile, for the accuracy rate evaluation, the mSADE is superior to the other algorithms in D_1, but not good enough for $iris$. The fact that the D_1 has the feature noise

is noticed. This case coincides with the fact that memetic methods perform well in noisy circumstances.

We also noticed that the D_2, $wine$ and $WisconsinBreastCancer$ are unbalanced and partially overlapped datasets. In these datasets, as shown in Table I-III, the performance of mSADE judged by CS, DB and the accuracy rate is good enough too.

At the same time, the mSADE is performing well in the relatively unbalanced and partially overlapped datasets, for instance, D_3 and $viche$. Something to note is that as the $viche$ is a high dimensional dataset which has 18 features, it maybe has some noisy data with some attributes.

To summarize, the performance of the mSADE algorithm in the noisy, unbalanced\ relatively unbalanced and partially overlapped is good. On the other hand, the robustness of mSADE is better than the others. Results shown in Figure 2-4 reveal that the mSADE is more cohesive than the typical evolutionary algorithms, the hybrid of SA with evolutionary methods and the latest efficient memetic algorithm employed when comparing in the terms of the CS, DB validity indices and the accuracy rate evaluation. It is shown that the mSADE is less sensitive to initial value than the algorithms used in comparison. Furthermore, performing well in the various types of datasets, the mSADE is flexible enough to use in many clustering problems.

VI. Conclusion

In this article, an efficient differential memetic algorithm for clustering is proposed. This approach employs simulated annealing as a local search mechanism to improve the performance of the clustering. Then an extensive comparison of several clustering algorithms has been done for both artificial and real-life datasets. Results show that the performance of mSADE is better than that of other algorithms when they are judged by the CS, DB and accuracy rate. Furthermore, capabilities of these methods in the aspect of various datasets are analyzed. As shown by the results of the computational tests, the proposed algorithm has better evaluation by validity indices and better accuracy than that of other methods in the balanced\unbalanced\relatively-balanced, non-overlapped\partially-overlapped, noisy circumstance. Based on the experiments on three synthetic and four real life datasets, the mSADE is more robust, flexible and insensitive to the initial value than other algorithms.

Despite the experimental evaluation presented in this article, the assessment judged by CS and DB is sometimes not efficient enough, which means an extensive research to find a more efficient clustering validity index is an important task down the road.

References

[1] F. Petitjean, A. Ketterlin, and P. Gançarski, "A global averaging method for dynamic time warping, with applications to clustering," *Pattern Recognition*, vol. 44, no. 3, pp. 678–693, 2011.

[2] S. S. Kamath and V. Ananthanarayana, "A bio-inspired, incremental clustering algorithm for semantics-based web service discovery," *International Journal of Reasoning-based Intelligent Systems*, vol. 7, no. 3-4, pp. 261–275, 2015.

[3] Y. Maleh and A. Ezzati, "Lightweight intrusion detection scheme for wireless sensor networks," *IAENG International Journal of Computer Science*, vol. 42, no. 4, pp. 347–354, 2015.

[4] J. Hou, W. Liu, E. Xu, and H. Cui, "Towards parameter-independent data clustering and image segmentation," *Pattern Recognition*, vol. 60, pp. 25–36, 2016.

[5] A. Awasthi, T. Adetiloye, and T. G. Crainic, "Collaboration partner selection for city logistics planning under municipal freight regulations," *Applied Mathematical Modelling*, vol. 40, no. 1, pp. 510–525, 2016.

[6] Q. Duan, Y. L. Yang, and Y. Li, "Rough k-modes clustering algorithm based on entropy." *IAENG International Journal of Computer Science*, vol. 44, no. 1, pp. 13–18, 2017.

[7] V. Singh, L. Mukherjee, J. Peng, and J. Xu, "Ensemble clustering using semidefinite programming with applications," *Machine learning*, vol. 79, no. 1, pp. 177–200, 2010.

[8] G. Krishnasamy, A. J. Kulkarni, and R. Paramesran, "A hybrid approach for data clustering based on modified cohort intelligence and k-means," *Expert Systems with Applications*, vol. 41, no. 13, pp. 6009–6016, 2014.

[9] M. Abul Hasan and S. Ramakrishnan, "A survey: hybrid evolutionary algorithms for cluster analysis," *Artificial Intelligence Review*, vol. 36, no. 3, pp. 179–204, 2011.

[10] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science*, vol. 344, no. 6191, pp. 1492–1496, 2014.

[11] J. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, and L. Chrétien, "The dynamics of collective sorting robot-like ants and ant-like robots," in *Proceedings of the first international conference on simulation of adaptive behavior on From animals to animats*, 1991, pp. 356–363.

[12] M. Omran, A. Salman, and A. P. Engelbrecht, "Image classification using particle swarm optimization," in *Proceedings of the 4th Asia-Pacific conference on simulated evolution and learning*, vol. 1. Singapore, 2002, pp. 18–22.

[13] S. Paterlini and T. Krink, "High performance clustering with differential evolution," in *Proceedings of the 2004 Congress on Evolutionary Computation, CEC2004*, vol. 2. IEEE, 2004, pp. 2004–2011.

[14] N. Monmarché, M. Slimane, and G. Venturini, "On improving clustering in numerical databases with artificial ants," *Advances in Artificial Life*, pp. 626–635, 1999.

[15] P. Kanade and L. Hall, "Fuzzy ants as a clustering concept," in *Fuzzy Information Processing Society, 2003. NAFIPS 2003. 22nd International Conference of the North American*. IEEE, 2003, pp. 227–232.

[16] S. Saatchi and C. Hung, "Hybridization of the ant colony optimization with the k-means algorithm for clustering," *Image Analysis*, pp. 283–293, 2005.

[17] G. Shang, Z. Zaiyue, Z. Xiaoru, and C. Cungen, "A new hybrid ant colony algorithm for clustering problem," in *Education Technology and Training, 2008. and 2008 International Workshop on Geoscience and Remote Sensing. ETT and GRS 2008. International Workshop on*, vol. 1. IEEE, 2008, pp. 645–648.

[18] B. Doğan and M. Korürek, "A new ecg beat clustering method based on kernelized fuzzy c-means and hybrid ant colony optimization for continuous domains," *Applied soft computing*, vol. 12, no. 11, pp. 3442–3451, 2012.

[19] C.-L. Huang, W.-C. Huang, H.-Y. Chang, Y.-C. Yeh, and C.-Y. Tsai, "Hybridization strategies for continuous ant colony optimization and particle swarm optimization applied to data clustering," *Applied Soft Computing*, vol. 13, no. 9, pp. 3864–3872, 2013.

[20] J. Wang and J. Song, "A hybrid algorithm based on gravitational search and particle swarm optimization algorithm to solve function optimization problems." *Engineering Letters*, vol. 25, no. 1, pp. 22–29, 2017.

[21] F. Ye and C. Chen, "Alternative kpso-clustering algorithm," *Tamkang Journal of Science and Engineering*, vol. 8, no. 2, pp. 165–174, 2005.

[22] X. Cui and T. Potok, "Document clustering analysis based on hybrid pso+ k-means algorithm," *Journal of Computer Sciences (Special Issue)*, vol. 27, pp. 1549–3636, 2006.

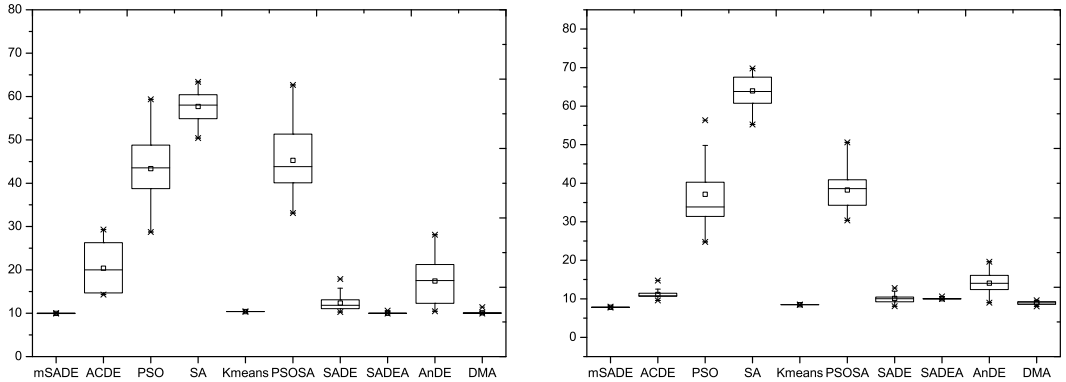
[23] G. M. Devi, M. Seetha, and K. Sunitha, "A novel hybrid clustering techniques based on k-means, pso and dynamic optimization," *International Journal of Computer Applications*, vol. 119, no. 20, pp. 20–25, 2015.

[24] T. Niknam, B. Amiri, J. Olamaei, and A. Arefi, "An efficient hybrid evolutionary optimization algorithm based on pso and sa for clustering," *Journal of Zhejiang University-Science A*, vol. 10, no. 4, pp. 512–519, 2009.

[25] R. Karthi, C. Rajendran, and K. Rameshkumar, "Neighborhood search assisted particle swarm optimization (npso) algorithm for partitional data clustering problems," in *Proceedings of the First International Conference on Advances in Computing and Communications, ACC 2011*. Springer, 2011, pp. 552–561.

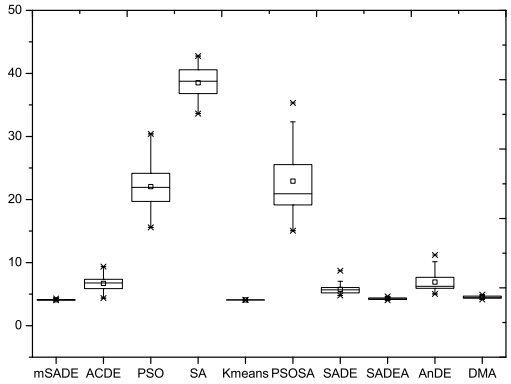
[26] H. Garg, "A hybrid pso-ga algorithm for constrained optimization problems," *Applied Mathematics and Computation*, vol. 274, no. C, pp. 292–305, 2016.

[27] J. Yan, Q. Ling, and D. Sun, "A differential evolution with simulated annealing updating method," in *Proceedings of the International*

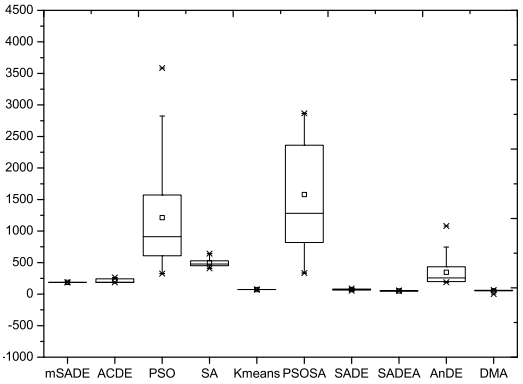


(a) D1

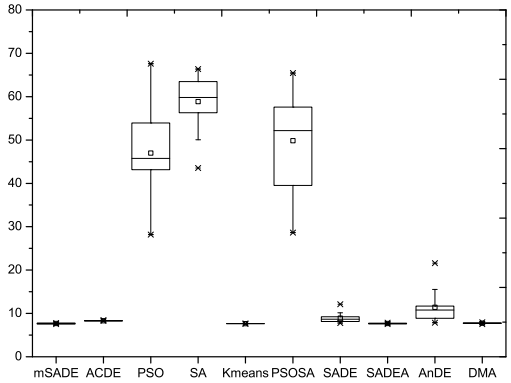
(b) D2



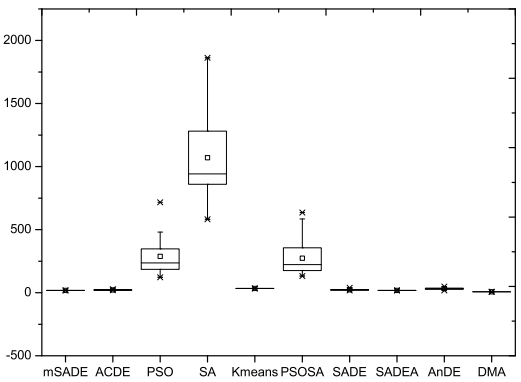
(c) D3



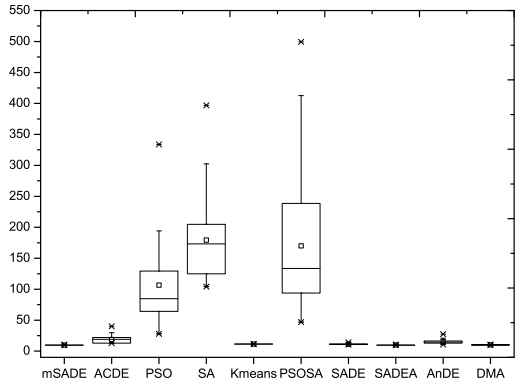
(d) Breast cancer



(e) Iris

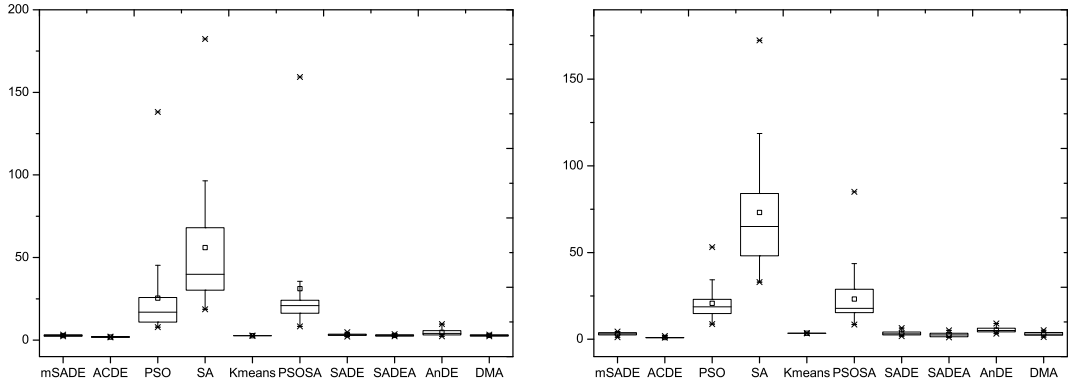


(f) Viche



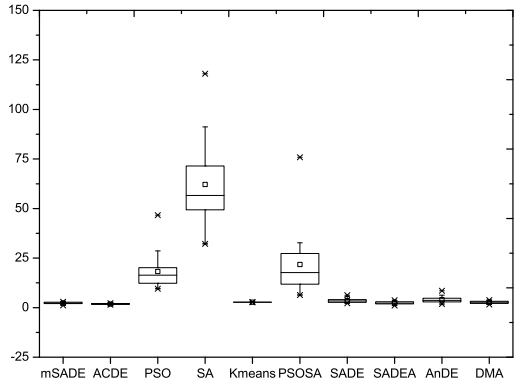
(g) Wine

Fig. 2: Results of ten clustering algorithms for three synthetic and four real datasets. The $y-$axis gives the CS index evaluation of the cluster partition. The boxplots show the distribution of results of every 30 runs.

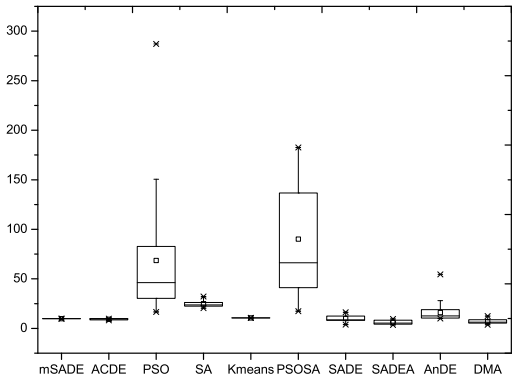


(a) D1

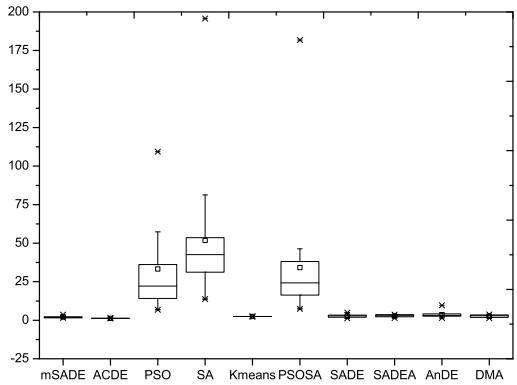
(b) D2



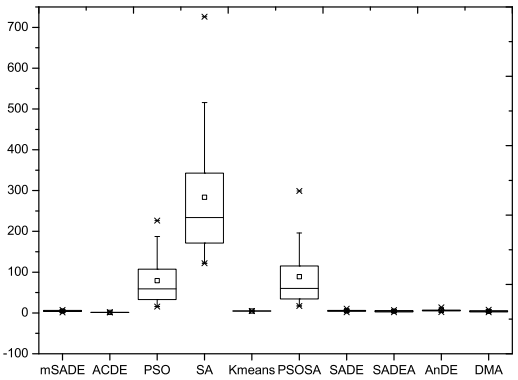
(c) D3



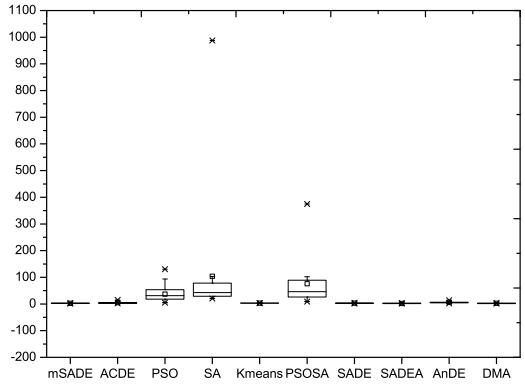
(d) Breast cancer



(e) Iris

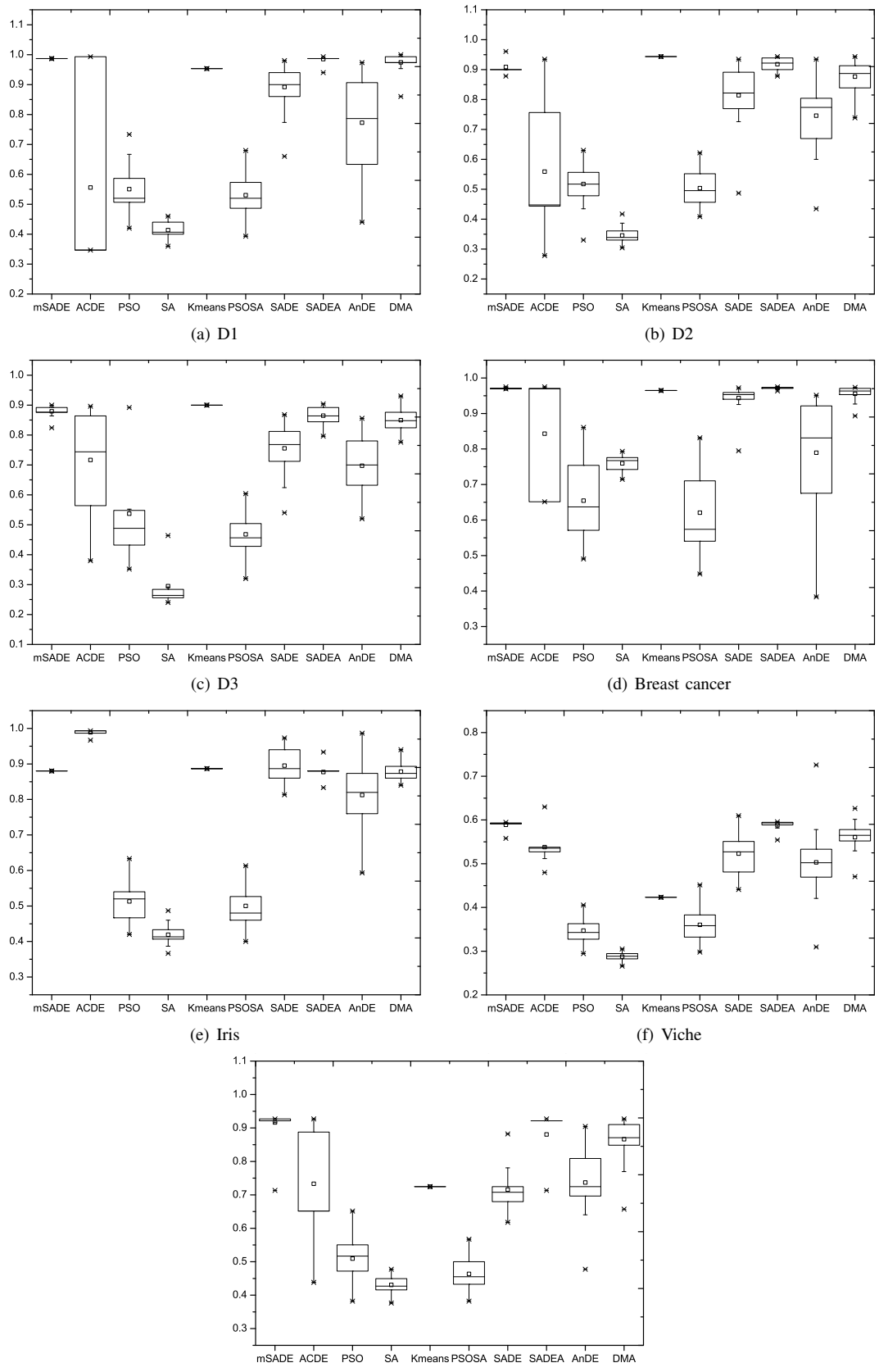


(f) Viche



(g) Wine

Fig. 3: Results of ten clustering algorithms for three synthetic and four real datasets. The $y-$axis gives the DB index evaluation of the cluster partition. The boxplots show the distribution of results of every 30 runs.



(a) D1 (b) D2

(c) D3 (d) Breast cancer

(e) Iris (f) Viche

(g) Wine

Fig. 4: Results of ten clustering algorithms for three synthetic and four real datasets. The $y-$axis gives the accuracy rate of the cluster partition. The boxplots show the distribution of results of every 30 runs.

Conference on Machine Learning and Cybernetics. IEEE, 2006, pp. 2103–2106.
[28] K. Liu, X. Du, and L. Kang, "Differential evolution algorithm based on simulated annealing," in *Advances in Computation and Intelligence*. Springer, 2007, pp. 120–126.
[29] S. Das, A. Konar, and U. K. Chakraborty, "Annealed differential evolution," in *Evolutionary Computation. CEC 2007. IEEE Congress on*, pp. 1926–1933.
[30] W. Kwedlo, "A clustering method combining differential evolution with the k-means algorithm," *Pattern Recognition Letters*, vol. 32, no. 12, pp. 1613–1621, 2011.
[31] K. Thangavel, J. Bagyamani, and R. Rathipriya, "Novel hybrid pso-sa model for biclustering of expression data," *Procedia Engineering*, vol. 30, pp. 1048–1055, 2012.
[32] M. Vakil-Baghmisheh and M. Ahandani, "A differential memetic algorithm," *Artificial Intelligence Review*, vol. 30, no. 1, pp. 129–146, 2014.
[33] S. Kirkpatrick, M. Vecchi *et al.*, "Optimization by simmulated annealing," *science*, vol. 220, no. 4598, pp. 671–680, 1983.
[34] B. Gidas, "Metropolis-type monte carlo simulation algorithms and simulated annealing," *Topics in Contemporary Probability and its Applications, CRC Press*, pp. 159–232, 1995.
[35] H. Zhang, S. Zhang, and W. Bu, "A clustering routing protocol for energy balance of wireless sensor network based on simulated annealing and genetic algorithm," *International Journal of Hybrid Information Technology*, vol. 7, no. 2, pp. 71–82, 2014.
[36] N. Rrasnogor, "Studies on the theory and design space of memetic algorithms," Ph.D. dissertation, University of the West of England, Bristol, 2002.
[37] N. Krasnogor and J. Smith, "A tutorial for competent memetic algorithms: model, taxonomy, and design issues," *Evolutionary Computation, IEEE Transactions on*, vol. 9, no. 5, pp. 474–488, 2005.
[38] S. Das, A. Abraham, and A. Konar, "Automatic clustering using an improved differential evolution algorithm," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, vol. 38, no. 1, pp. 218–237, 2008.
[39] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.
[40] S. Kathpal, R. Vohra, J. Singh, and R. Sawhney, "Hybrid pso–sa algorithm for achieving partitioning optimization in various network applications," *Procedia Engineering*, vol. 38, pp. 1728–1734, 2012.
[41] M. Basu, P. Deb, and G. Garai, "Hybrid of particle swarm optimization and simulated annealing for multidimensional function optimization," *International Journal of Information Technology*, vol. 20, no. 1, pp. 112–120, 2014.
[42] M. Pakhira, S. Bandyopadhyay, and U. Maulik, "Validity index for crisp and fuzzy clusters," *Pattern recognition*, vol. 37, no. 3, pp. 487–501, 2004.
[43] K. Rizman Žalik, "Cluster validity index for estimation of fuzzy clusters of different sizes and densities," *Pattern Recognition*, vol. 43, no. 10, pp. 3374–3390, 2010.
[44] D. Davies and D. Bouldin, "A cluster separation measure," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-1, no. 2, pp. 224–227, 1979.
[45] C. Chou, M. Su, and E. Lai, "A new cluster validity measure and its application to image compression," *Pattern Analysis & Applications*, vol. 7, no. 2, pp. 205–220, 2004.
[46] R. Xu, J. Xu, and D. C. Wunsch, "A comparison study of validity indices on swarm-intelligence-based clustering," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, vol. 42, no. 4, pp. 1243–1256, 2012.

L ei Jiang was born in Xinning, Hunan Province, China in 1974. He received his M.S. and Ph.D. degrees in Computer Science from Southwest Petroleum University in 2005 and Wuhan University in 2012, respectively. He has been a associate professor of Hunan University of Science and Technology, Xiangtan, China, since 2005. His research interests include pattern recognition, evolutionary computation, machine learning.

Datong Xie was born in Shuangfeng, Hunan Province, China in 1977. He received his M.S. and Ph.D. degrees in Computer Science from China University of Geosciences in 2007 and Wuhan University in 2013, respectively. He has been a associate professor at Department of Information Management Engineering, Fujian Commercial College, Fuzhou, China, since 2008. His research interests include evolutionary computation, computer algorithm and software engineering.