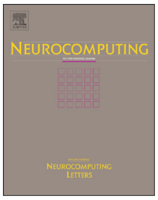
Neurocomputing 158 (2015) 144–154

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



A dynamic shuffled differential evolution algorithm for data clustering

Wan-li Xiang [a,b,*], Ning Zhu [a], Shou-feng Ma [a], Xue-lei Meng [b], Mei-qing An [b]

[a] *Institute of Systems Engineering, Tianjin University, Tianjin 300072, PR China*
[b] *School of Traffic & Transportation, Lanzhou Jiaotong University, Lanzhou, Gansu 730070, PR China*

ARTICLE INFO

Article history:
Received 21 June 2014
Received in revised form
5 December 2014
Accepted 30 January 2015
Communicated by Swagatam Das
Available online 8 February 2015

Keywords:
Differential evolution
Shuffled scheme
Information exchange
Novel initial technique
Data clustering

ABSTRACT

In order to further improve the convergence performance of data clustering algorithms, a dynamic shuffled differential evolution algorithm, DSDE for short, is presented in this paper. In DSDE, mutation strategy DE/best/1 is employed, which can take advantage of the direction guidance information of best individual so as to speed up the corresponding algorithm. Meanwhile, inspired by shuffled frog leaping algorithm, a sorting scheme and a randomly shuffled scheme are used to divide a total population into two subpopulations during the evolving process. In this way, mutation strategy DE/best/1 is actually used in two subpopulations, respectively, which can effectively exchange information between two subpopulations and balance the exploitation ability of DE/best/1/bin. In addition, most popular data clustering algorithms suffer from the choice of initial clustering centers, which may cause a premature convergence. Here a novel initial technique, called the random multi-step sampling, is integrated into DSDE to overcome the shortcoming. Then an experiment tested on 11 well-known datasets has been carried out, and the related results demonstrate that DSDE significantly outperforms DE/rand/1/bin and DE/best/1/bin. Next, another comparison among DSDE and other four well-known data clustering algorithms is conducted. The related results also show that DSDE is superior to other four approaches including particle swarm optimization with age-group topology (PSOAG) in terms of objective function value, i.e., the sum of intra-cluster distance. In a word, all the experimental results confirm that the proposed algorithm DSDE can be considered as an excellent tool for data clustering.

1. Introduction

Clustering is an unsupervised learning method that divides a dataset into groups of similar objects by minimizing the similarity between objects in different clusters and maximizing the similarity between objects within the same cluster. When used on a set of objects, it is helpful for identifying some inherent structures of the set of objects. Thus, clustering analysis can play a key role in many research fields including data mining, machine learning, bioinformatics and so on. Especially, it can work well without any prior information, which is also the main difference to supervised/ semisupervised classification. During the past five decades, many research works have been proposed for data clustering [6]. That is, the clustering problem has arisen many ways to cluster a dataset. These approaches mainly focus on complex network approach [1,27,29], K-means [4] and its enhanced variants [5,12,35,43], swarm intelligent algorithms [2,3,7,9,11,15-25,28,31,32,35-37,44], and others [8,10,13,14,26,30,33,34]. Among them, K-means algorithm is one of the most popular algorithms, and it is an unsupervised data clustering algorithm, which tries to divide an entire dataset (i.e., X) into K clusters (i.e., $C_1, C_2, \ldots, C_K$) through randomly choosing K data points as initial cluster centers. However, K-means algorithm is sensitive to the selection of initial points and may fail to cluster large scale datasets [12,36,43].

In order to overcome the shortcomings of K-means algorithm, many researchers have focused on swarm intelligence algorithms, which can perform parallel search in a complex search space so as to better avoid local minima trap. For example, Chuang et al. [2] proposed a chaotic particle swarm optimization (CPSO) for data clustering by replacing the random parameters of PSO with chaotic variables of the logistic map. Inspired by the crossover operator of genetic algorithm, Yan et al. [22] proposed a hybrid artificial bee colony algorithm (HABC) by introducing arithmetic crossover operation in original artificial bee colony algorithm. Hatamlou et al. [44] proposed a combined algorithm based on K-means algorithm and gravitational search algorithm (GSA). The related comparisons show that it is superior to both K-means algorithm and GSA. More recently, Jiang et al. [11] proposed a novel age-based particle swarm optimization, which is called the PSOAG. In PSOAG, a novel technique of keeping population diversity is introduced. Meantime, its superiority is obvious when compared with ant colony optimization algorithm (ACO), particle swarm

[*] Corresponding author.
E-mail address: xiangwl@tju.edu.cn (W.-l. Xiang).

http://dx.doi.org/10.1016/j.neucom.2015.01.058
0925-2312/© 2015 Elsevier B.V. All rights reserved.

optimization (PSO), artificial bee colony algorithm (ABC) and differential evolution (DE) algorithm in terms of fitness function values and the clustering accuracy. In summary, the performances achieved by these modified variants based on various evolutionary computing techniques are better than before.

However, there still exists the shortcoming of premature convergence in those modified evolutionary algorithms to some extent. To be specific, the exploitation and exploration abilities of existing evolutionary algorithms have to be further balanced. Especially, the population diversity is necessary to be better kept during the evolution process, and a better mechanism of information sharing among different individuals are also needed to be designed. In order to further improve the clustering effectiveness of swarm based intelligent algorithms, a dynamic shuffled differential evolution, named DSDE, is proposed in view of the faster convergence speed of DE [41,42]. When compared with DE/rand/1/bin, DE/best/1/bin, ACO, ABC, PSO, and the recent PSOAG, the superiority of DSDE is obvious.

The rest of the paper is organized as follows. In Section 2, the problem formulation over data clustering is described. In Section 3, the traditional differential evolution algorithm is briefly described. Subsequently, a dynamic shuffled differential evolution, called the DSDE, is presented in detail in Section 4. Nextly, comprehensive experiments are conducted in Section 5 to validate the performance of DSDE. Finally, a conclusion is drawn in Section 6.

2. Problem formulation

For the global optimization based clustering problem [3,22,50], a clustering criterion, i.e., a degree of similarity, is needed, with which a global assignment can be found by minimizing the objective function of the sum of intra-cluster distance. As far as the similarity measurement is concerned, there are various similarity measurements such as Euclidian distance, Manhattan distance and Minkowski distance within some clustering researches. In general, the Euclidian distance is usually considered as the similarity measurement. In our study, the measurement of Euclidean distance is also used to calculate the distance of any two objects (o_i and o_j) within the cluster. Generally, it can be formulated as follows [22,36]:

$$D(o_i, o_j) = \| o_i - o_j \| = \sqrt{\sum_{m=1}^{d} (o_{im} - o_{jm})^2} \tag{1}$$

where d is the number of attributes of object, and o_i and o_j are two different objects within the cluster, respectively.

Based on the above similarity measurement, a partition clustering problem can be converted into a global optimization problem, which can be described as follows [3,11,21]:

$$\underset{Z,W}{\text{Minimize}} \quad f(Z,W) = \sum_{k=1}^{K} \sum_{i=1}^{n} w_{ik} D(x_i, z_k) \tag{2}$$

$$\text{s.t.} \begin{cases} \sum_{k=1}^{K} w_{ik} = 1, \quad i = 1,2,\ldots,n; & \text{(a)} \\ w_{ik} \in \{0,1\}, \quad \forall i \in \{1,2,\ldots,n\},\ k \in \{1,2,\ldots,K\} & \text{(b).} \end{cases} \tag{3}$$

where n is the number of all the samples, and K is the number of given clusters, and x_i represents the coordinates of the ith object in current samples, and $w_{ik} = 1$ means that the ith object is grouped into the kth cluster, or else the ith object does not belong to the kth cluster, and $D(x_i, z_k)$ denotes the distance between the ith object x_i and the center of kth cluster z_k. It should be noted that $W = \{w_{ik} | i = 1,2,\ldots,n, k = 1,2,\ldots,K\}$, and $Z = \{z_k | k = 1,2,\ldots,K\}$.

What is more, the value of w_{ik} in Eq. (3) depends on a partition criterion of samples. Namely, given a sample set $X = \{x_1, x_2, \ldots, x_n\}$, determine a partition of all objects which satisfies the following equations [3]:

$$\begin{cases} \sum_{i=1}^{K} C_i = X; & & \text{(a)} \\ C_i \cap C_j = \phi, & \forall i,j \in \{1,2,\ldots,K\} \wedge i \neq j; & \text{(b)} \\ C_i \neq \phi, & \forall i \in \{1,2,\ldots,K\} & \text{(c).} \end{cases} \tag{4}$$

where $C_i (i = 1,2,\ldots,K)$ is the objects set of ith cluster, and its members can be determined by the following [3]:

$$\begin{cases} C_i = \{x_k | \ \|x_k - z_i\| \leq \|x_k - z_p\|, x_k \in X\}, & p \neq i, p = 1,2,\ldots,K; & \text{(a)} \\ z_i = \dfrac{1}{|C_i|} \sum_{x_k \in C_i} x_k, & i = 1,2,\ldots,K & \text{(b).} \end{cases} \tag{5}$$

where $\| \cdot \|$ represents the Euclidean distance of any two objects in the sample set, and the mean of all objects within cluster i, z_i, denotes a new center of cluster i, which is often used in well-known clustering technique k-means.

3. Differential evolution algorithm

Differential evolution algorithm, first proposed by Storn and Price, is a simple yet powerful meta-heuristic algorithm for global optimization over continuous search space [41]. Since its invention in 1997, the DE algorithm has attracted many researchers to study the creative algorithm. For example, Liu and Lampinen proposed a fuzzy adaptive differential evolution algorithm by using a fuzzy controller to adapt the control parameters [47]. Teo proposed a parameterless differential evolution algorithm based on self-adaption [48]. Rahnamayan et al. proposed an opposition based differential evolution algorithm (ODE) [42], in which a novel opposition based learning technique is proposed. Gong et al. proposed a hybrid algorithm DE/BBO based on differential evolution and biogeography-based optimization [49].

Like other population-based intelligent algorithms, the first phase is to randomly generate an initial population $X (X = \{x_i | x_i = (x_{i1}, x_{i2}, \ldots, x_{iD}), i = 1,2,\ldots,N_p\})$ in DE. Subsequently, DE enters a loop composed of mutation, crossover, and selection operations.

3.1. Mutation

At this phase, a mutant vector v_i is generated by the following equation:

$$v_i = x_a + F \cdot (x_b - x_c) \tag{6}$$

where $i = 1,2,\ldots,N_p$, and $a, b, c \in \{1,2,\ldots,N_p\}$ are random integer number and $a \neq b \neq c \neq i$, and scale factor F is a real and constant number within $[0,2]$, which is employed to control the amplification of differential deviation $(x_b - x_c)$ [41].

3.2. Crossover

At the second phase, DE usually utilizes a binomial crossover operation to produce a trial vector $u_i = (u_{i1}, u_{i2}, \ldots, u_{iD})$ according to the following equation:

$$u_{ij} = \begin{cases} v_{ij} & \text{if } \text{rand}[0,1]_j < Cr \vee j == j_{rand}, \\ x_{ij} & \text{otherwise ,} \end{cases} \tag{7}$$

where $i = 1,2,\ldots,N_p$, and $j = 1,2,\ldots,D$, and $\text{rand}[0,1]_j$ is a real random number within [0,1], and $j_{rand} \in \{1,2,\ldots,D\}$ is a randomly generated integer, which is used to make sure that the trial vector u_i gets at least one component from the mutant vector v_i. In addition, the key parameter *Cr* is a predefined constant within the range of [0,1], which is used to control the fraction of component values copied from the mutant vector v_i.

3.3. Selection

At the last phase, each trial vector u_i is evaluated. Then a better individual between $\{x_i, u_i\}$ is retained according to their fitness values. Regarding a minimization problem, the chosen vector $x_i^\star$ is given according to the following equation:

$$x_i^\star = \begin{cases} u_i & \text{if } f(u_i) < f(x_i), \\ x_i & \text{otherwise}. \end{cases} \tag{8}$$

where $i = 1, 2, \ldots, N_p$, and $f(\cdot)$ denotes the objective function value of a solution, and the retained solution $x_i^\star$ represents a parent vector for substituting the target vector x_i in the next generation.

In view of the flexibility of the aforementioned mutation scheme and crossover mode, DE can be extended to a variety of versions. The notation DE/*x*/*y*/*z* was introduced by R Storn and K Price [41] in order to classify the different variants, where "*x*" specifies the vector to be mutated, "*y*" is the number of difference vector used, and "*z*" denotes the type of crossover scheme (bin: binomial; exp: exponential). For more details about the notation, please refer to the literature [41]. It should be noted that the notation "DE/*x*/*y*" is used to denote mutation strategy itself in this paper. In this way, it can be distinguished from the complete notation of the DE algorithm.

4. A dynamic shuffled differential evolution algorithm

4.1. Solution representation

For the clustering optimization problem, an individual is represented by a coordinate of centroid, which is encoded by a D-dimensional vector. Meanwhile, it depends on the number of clusters and the number of attributes of objects in a dataset. That is, the dimension size of optimization parameters $D = K \cdot d$, where the first d-dimensional vector denotes the coordinate of first cluster center, the *k*th D-dimensional vector is the coordinate of *k*th cluster center, and so on.

4.2. Novel initial technique

In order to improve the solution quality of initial population and better keep the population diversity, a novel initial technique, called the random multi-step sampling method, is proposed to initial the first population. In the initial technique, there need *K* steps to obtain an initial coordinate x_i $(x_i = (\underbrace{x_{i,1}, x_{i,2}, \ldots, x_{i,d}}, \underbrace{x_{i,d+1}, x_{i,d+2}, \ldots, x_{i,2d}}, \ldots, \underbrace{x_{i,(K-1)d+1}, x_{i,(K-1)d+2}, \ldots, x_{i,Kd}}))$ for each dataset with *K* clusters. In addition, a mean strategy based on median-weighted is used to collect samples from a dataset during every initial step. The specific implementation is given in Algorithm 1.

Algorithm 1. Initialization based on random multi-step sampling.

```
for i=1 to $N_p$ do
   // $K$ denotes the number of clusters
for $k=1$ to $K$ do
        Generate three integers $a, b, c \in [1..m] \wedge a \neq b \neq c$
   randomly
        // $m$ represents the number of samples
        Calculate the median value $me$ of samples $x_a, x_b, x_c$
        Compute the arithmetic mean $\mu$ of samples $x_a, x_b, x_c$
   with their median
        value $me$
        Take $\mu$ as the initial center of the $k$th cluster
   end for
end for
```

As shown in Algorithm 1, an arithmetic mean μ is usually used to represent the central tendency of a group of data $\{x_1, x_2, \ldots, x_n\}$, and $\mu = \sum_{i=1}^{n} x_i / n$, which is less influenced than both the median and the mode on the aspect of statistical advantage. Thus, the initial technique can make initial population possess better diversity through a number of random samplings. But it is easier to be influenced by some outliers [45]. To overcome the shortcoming, in the initial strategy, three samples (x_a, x_b, x_c) randomly chosen together with their median are all used to compute the arithmetic mean, which is consider as an initial cluster center. In this way, the median of $\{x_a, x_b, x_c\}$ receives heavier weights, which is helpful to reduce the effect of some outliers on the arithmetic mean.

4.3. Constraint handling

During the evolutionary process, some individual variables may exceed the bound constraints, i.e., bounds of attributes of sample in a dataset, which are denoted by $[y_j^{min}, y_j^{max}]$ $(j = 1, 2, \ldots, d)$, where *d* is the number of properties of an object in a dataset. For the purpose of conveniently performing a comparison between an individual and bound constraints, x^{min} and x^{max} are used to denote the lower bound and upper bound of an individual, respectively. Next, they are described as follows:

$$x^{min} = [\underbrace{y_1^{min}, y_2^{min}, y_d^{min}}_{1}, \ldots, \underbrace{y_1^{min}, y_2^{min}, y_d^{min}}_{K}] \tag{9}$$

$$x^{max} = [\underbrace{y_1^{max}, y_2^{max}, y_d^{max}}_{1}, \ldots, \underbrace{y_1^{max}, y_2^{max}, y_d^{max}}_{K}] \tag{10}$$

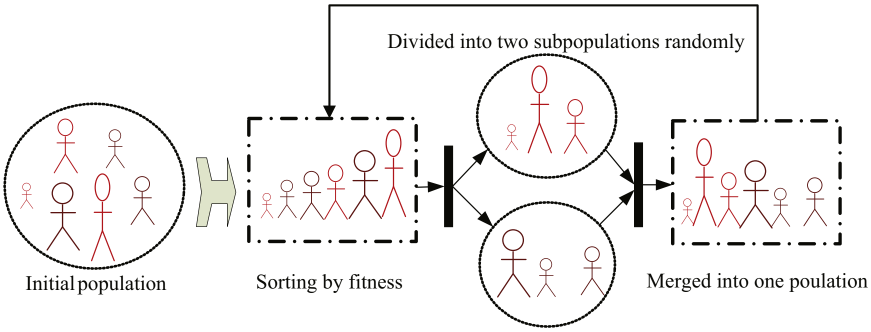

Divided into two subpopulations randomly
Initial population
Sorting by fitness
Merged into one poulation

Fig. 1. The idea behind DSDE.

When a component x_j of an individual x violates related constraints, the following repairing rule is employed:

$$x_j = \begin{cases} x_j^{min} + \text{rand}(0,1) \cdot (x_j^{max} - x_j^{min}) & \text{if } x_j < x_j^{min}, \\ x_j^{max} - \text{rand}(0,1) \cdot (x_j^{max} - x_j^{min}) & \text{if } x_j > x_j^{max}. \end{cases} \quad (11)$$

where $j = 1, 2, \ldots, K \cdot d$, and rand(0, 1) is used to generate a random number within [0,1].

4.4. The proposed approach

In DSDE, mutation strategy DE/best/1 is employed because it can speed up the convergence speed of differential evolution under the guidance of best individual (base vector). At the same time, a new produced individual is immediately compared with the target vector in a current population, and the better individual could be retained, which can further faster guide the corresponding population evolution. However, all these may cause DSDE premature convergence. To this end, the total population is divided into two subpopulations according to the uniform random distribution, which may make at least one subpopulation evolve under the guidance of non-global best individual at the beginning of search. In order to further improve the population diversity, inspired by shuffled frog leap algorithm (SFLA) [38,39], two subpopulations are merged into a total population after each subpopulation completes an evolution process of a generation. Then the merged population are sorted by the fitness values. Next, the sorted population is uniformly randomly divided into two subpopulations again. The process is called a shuffled process, which is used to improve the population diversity and to be helpful to exchange/share information between two subpopulations. The idea behind the process is given in Fig. 1.

Based on the aforementioned analysis, the detailed description of proposed algorithm DSDE is presented in Algorithm 2.

Table 1
The summary of test datasets used in experiments.

Datasets	K	d	Number of data objects	Description
Art1	4	2	600(150,150,150,150)	Artificial data
Art2	5	3	250(50,50,50,50,50)	Artificial data
Iris	3	4	150(50,50,50)	Fisher's iris data
Wine	3	13	178(59,71,48)	Wine quality data
Glass	6	9	214(70,76,17,13,9,29)	Glass identification data
Cancer	2	9	683(444,239)	Wisconsin breast cancer
CMC	3	9	1473(629,334,510)	Contraceptive method choice
Vowel	6	3	871(72,89,172,151,207,180)	Indian Telugu vowel
Crude Oil	3	5	56(7,11,38)	Crude Oil quality data
Thyroid	3	5	215(150,35,30)	Thyroid gland data
bupaLD	2	6	345(200,145)	Liver disorders

Algorithm 2. The DSDE algorithm.

```
Set $N_p = n_1 + n_2$ //$n_1$ and $n_2$ represent the size of the first sub-
    population and the second sub-population, respectively
Initialize the population $X$ of $N_p$ individuals using Algorithm 1
Compute the fitness for each individual
Set $FEs = N_p$ //$FEs$ represents the iterative variable
while $FEs \le maxFEs$ do
    Sort the population $X$
    Subdivide the population $X$ into two sub-populations $X'$
    and $X''$ randomly
```

Table 2
Comparison of objective values among DE/rand/1/bin, DE/best/1/bin and DSDE on the 11 datasets.

Data sets	Methods	Best	Worst	Median	Mean	Std.	Sig.
Art1	DE/rand/1/bin	562.30	651.53	596.91	597.54	23.3496	[a]
	DE/best/1/bin	535.17	535.17	535.17	535.17	0	[b]
	DSDE	535.17	535.17	535.17	535.17	0	
Art2	DE/rand/1/bin	2152.85	2869.66	2674.61	2619.72	188.5628	[a]
	DE/best/1/bin	1731.92	2367.46	1733.37	1784.89	147.2191	[a]
	DSDE	1731.88	1731.88	1731.88	1731.88	0	
Iris	DE/rand/1/bin	97.54	107.42	102.01	101.29	2.6852	[a]
	DE/best/1/bin	96.65	97.31	96.65	96.70	0.1485	[a]
	DSDE	96.65	96.65	96.65	96.65	0	
Wine	DE/rand/1/bin	16,306.87	16,401.15	16,355.28	16,351.61	27.8355	[a]
	DE/best/1/bin	16,294.63	16,313.26	16,300.18	16,302.01	4.9672	[a]
	DSDE	16,292.18	16,292.66	16,292.18	16,292.39	0.2450	
Glass	DE/rand/1/bin	293.72	347.33	313.51	315.37	15.3437	[a]
	DE/best/1/bin	246.99	290.18	265.24	266.53	12.4545	[a]
	DSDE	210.05	215.64	213.00	212.73	1.6796	
Cancer	DE/rand/1/bin	2994.45	3241.09	3048.93	3062.58	56.5500	[a]
	DE/best/1/bin	2964.61	3005.07	2969.03	2974.10	10.9925	[a]
	DSDE	2964.38	2964.38	2964.38	2964.38	0	
CMC	DE/rand/1/bin	5590.99	5915.62	5798.70	5790.81	76.0309	[a]
	DE/best/1/bin	5547.54	5702.15	5609.31	5612.61	45.3701	[a]
	DSDE	5532.18	5532.18	5532.18	5532.18	0	
Vowel	DE/rand/1/bin	172,623.36	186,531.43	179,915.92	179,792.69	344.5600	[a]
	DE/best/1/bin	149,011.17	157,749.67	150,795.00	151,080.71	227.8969	[a]
	DSDE	148,967.24	150,138.92	149,062.41	149,193.97	373.4470	
Crude Oil	DE/rand/1/bin	278.44	287.32	280.32	282.14	3.4532	[a]
	DE/best/1/bin	277.21	278.05	277.42	277.51	0.2712	[a]
	DSDE	277.21	277.30	277.21	277.22	0.0329	
Thyroid	DE/rand/1/bin	1924.74	2173.05	2034.52	2043.18	75.4963	[a]
	DE/best/1/bin	1868.60	1986.21	1920.82	1919.87	29.9104	[a]
	DSDE	1866.46	1895.99	1868.29	1874.00	11.7699	
bupaLD	DE/rand/1/bin	9852.60	9864.31	9856.38	9857.12	2.8990	[a]
	DE/best/1/bin	9851.71	9907.75	9852.48	9857.02	12.6783	[a]
	DSDE	9851.71	9852.07	9851.71	9851.73	0.0782	

[a] DSDE is better than its competitor by the Wilcoxon's rank sum test at $\alpha = 0.05$.
[b] There is no significant difference between DSDE and its competitor.

```
    // $X' = \{x'_1, x'_2, \ldots, x'_{n_1}\}$, $X'' = \{x''_1, x''_2, \ldots, x''_{n_2}\}$
    //The evolving process of subpopulation $X'$
    for i=1 to $n_1$ do
        Choose the best individual in the subpopulation $X'$, and its index refers to as pbest
        Produce two integers $a, b \in [1..n_1]$ randomly and $a \neq b \neq i$
        Generate noise vector $v$ by using $v = x'_{pbest} + F \cdot (x'_a - x'_b)$
        Perform constraint handling operation
        Generate trial vector $u$ by using binary crossover operation
        Compute the objective $f(u)$ of temporary individual $u$
        Set $FEs = FEs + 1$
        if $f(u) < f(x'_i)$ then
            Replace the target vector $x'_i$ with trial vector $u$ immediately
        end if
    end for
    //The evolving process of subpopulation $X''$
    for i=1 to $n_2$ do
        Choose the best individual in the subpopulation $X''$, and its index refers to as pbest
        Produce two integers $a, b \in [1..n_2]$ randomly and $a \neq b \neq i$
        Generate noise vector $v$ by using $v = x''_{pbest} + F \cdot (x''_a - x''_b)$
        Perform constraint handling operation
        Generate trial vector $u$ by using binary crossover operation
        Calculate the objective $f(u)$ of temporary individual $u$
        Set $FEs = FEs + 1$
        if $f(u) < f(x''_i)$ then
            Replace the target vector $x''_i$ with trial vector $u$ immediately
        end if
```

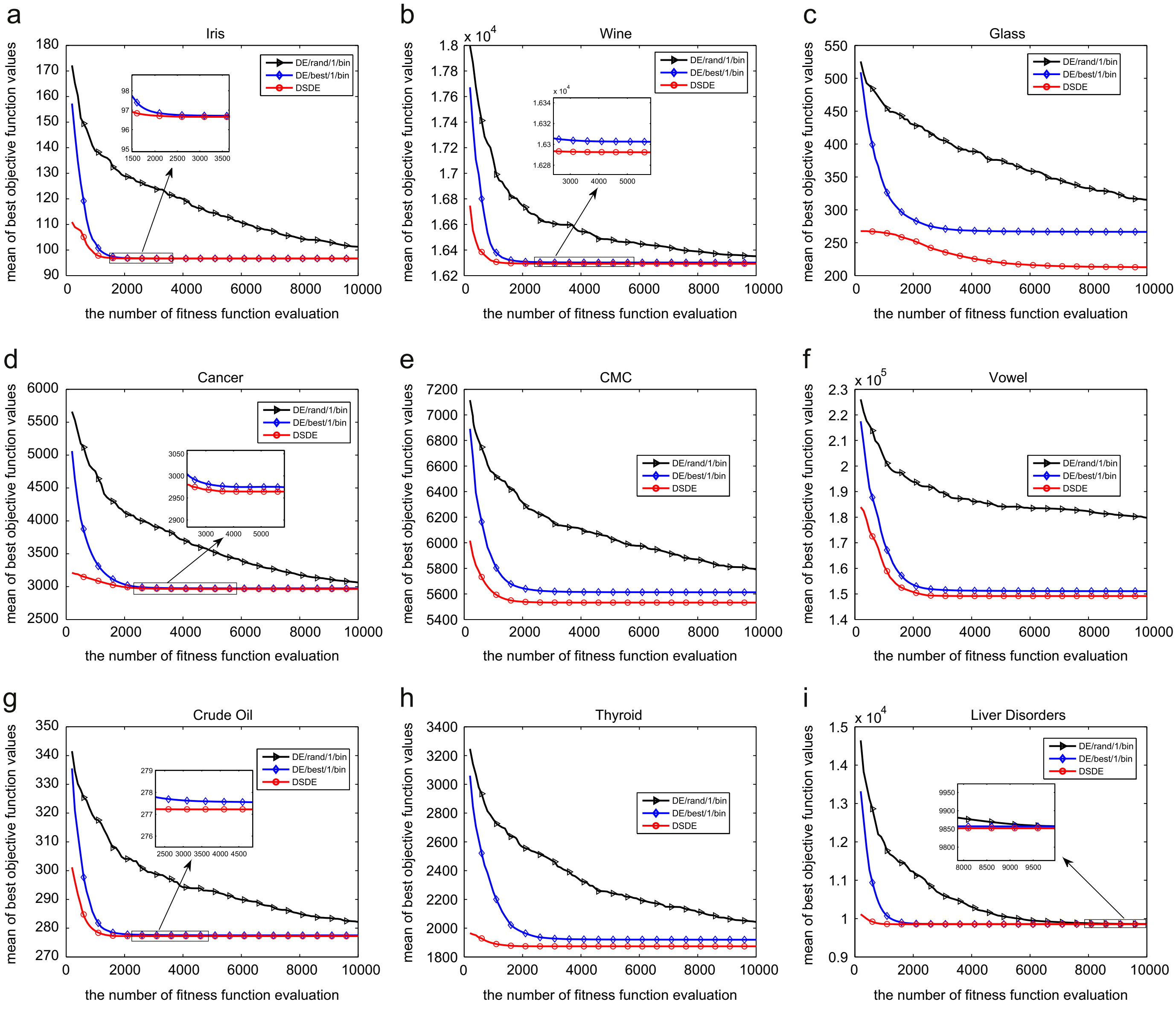


Fig. 2. Convergence performance of DE/rand/1/bin, DE/best/1/bin and DSDE on the nine benchmark datasets. (a) Iris, (b) 1 Wine, (c) Glass, (d) Cancer, (e) CMC, (f) Vowel, (g) Crude Oil, (h) Thyroid, (i) bupaLD.

```
        end for
    Merge the two sub-populations $X'$ and $X''$ into a population $X$
    Record the best solution achieved so far in the current population $X$
end while
```

Last but not least, it should be noticed that Eq. (2) is considered as the objective function of DSDE. In addition, the objective function values are also directly used to compare the superiority of two individuals in DSDE.

5. Experimental study and discussion

5.1. Benchmark data sets and parameter settings

In order to validate the performance of proposed algorithm DSDE, we construct a testbed composed of 11 well-known datasets, which are widely used in a variety of researches [7,11,17,22,35,36,40]. These benchmark datasets are listed briefly in Table 1, where the number of clusters of each dataset is denoted by **K**, and **d** specifies the number of attributes of each dataset. In addition, all datasets except Art1 & Art2 are real life datasets. Art1 and Art2 are randomly generated according to a bivariate normal distribution and an uniform distribution, respectively [36]. In specific, Classes in the artificial dataset Art1 are distributed according to $N_2(\mu, \Sigma)$, where $\mu = (\mu_i, \mu_i)'$ is the mean vector, and $\Sigma = \begin{bmatrix} 0.50 & 0.05 \\ 0.05 & 0.50 \end{bmatrix}$ is the covariance matrix. In reality, $\mu_1 = -3$, $\mu_2 = 0$, $\mu_3 = 3$, and $\mu_4 = 6$ are used to produce four independent clusters, respectively. The artificial dataset Art2 contains 250 objects with three features and five clusters, and every feature of these clusters is distributed according to five independent uniform distributions with ranges of [85,100], [70,85], [55,70], [40,55] and [25,40], respectively.

For a fair comparison, the maximum number of fitness function evaluations (*maxFEs*) is set to 1e4 as recommended in Jiang et al. [11] in all experiments. In addition, other parameters for DE/rand/1/bin, DE/best/1/bin and DSDE are set as follows:

Table 4
The best clustering center on the data sets Cancer and bupaLD.

Data sets	Center1	Center2
Cancer	7.117120827982673	2.889329284178662
	6.640969453546074	1.127768891100870
	6.625393415447806	1.200659417770133
	5.614330131509711	1.164149874300891
	5.240748395467229	1.993402097654656
	8.100901186763572	1.121191011120431
	6.078219125594872	2.005409693809643
	6.021873351600457	1.101323727152610
	2.325717925036817	1.031605044353600
bupaLD	90.943297479406027	89.895305474079862
	74.865755013951301	66.422217791442719
	48.894039158789127	23.922375125224157
	33.725767934385857	21.305797612097820
	90.828560716481320	21.882825135197557
	5.660648641397779	2.806378448273524

Table 3
Comparison of clustering accuracy among DE/rand/1/bin, DE/best/1/bin and DSDE on the 11 datasets.

Datasets	Methods	Best%	Worst%	Median%	Mean%	Std.	Sig.
Art1	DE/rand/1/bin	100.00	99.00	99.66	99.60	0.0023	[a]
	DE/best/1/bin	99.83	99.66	99.83	99.78	7.00e – 004	[b]
	DSDE	99.83	99.66	99.83	**99.79**	**7.00e – 004**	
Art2	DE/rand/1/bin	100.00	77.60	95.20	92.38	0.0805	[a]
	DE/best/1/bin	100.00	80.00	100.00	98.96	0.0446	[b]
	DSDE	100.00	100.00	100.00	**100.00**	**0**	
Iris	DE/rand/1/bin	92.00	88.66	90.00	89.90	0.0084	[b]
	DE/best/1/bin	92.66	89.33	90.00	90.00	0.0068	[b]
	DSDE	90.00	90.00	90.00	**90.00**	**0**	
Wine	DE/rand/1/bin	72.47	70.78	71.34	71.40	0.0060	[b]
	DE/best/1/bin	71.91	70.78	71.34	71.26	0.0045	[a]
	DSDE	71.91	71.34	71.91	**71.65**	**0.0028**	
Glass	DE/rand/1/bin	51.40	45.32	49.53	49.50	**0.0138**	[a]
	DE/best/1/bin	54.67	48.13	50.46	50.51	0.0148	[a]
	DSDE	56.07	48.13	54.43	**53.48**	0.0283	
Cancer	DE/rand/1/bin	96.63	95.31	95.90	96.00	0.0041	[a]
	DE/best/1/bin	96.63	95.90	96.48	96.37	0.0021	[b]
	DSDE	96.486	96.486	96.486	**96.486**	**0**	
CMC	DE/rand/1/bin	38.83	35.77	38.32	38.20	0.0067	[b]
	DE/best/1/bin	38.49	37.13	37.84	37.82	0.0043	[a]
	DSDE	38.49	38.49	38.49	**38.49**	**0**	
Vowel	DE/rand/1/bin	62.91	40.52	51.83	52.00	0.0679	[b]
	DE/best/1/bin	60.16	49.48	56.31	55.31	0.0334	[b]
	DSDE	57.97	53.50	55.62	**56.21**	**0.0122**	
Crude Oil	DE/rand/1/bin	69.64	62.50	67.85	67.05	0.0242	[b]
	DE/best/1/bin	69.64	64.28	67.85	67.50	0.0148	[b]
	DSDE	67.85	66.07	67.85	**67.58**	**0.0065**	
Thyroid	DE/rand/1/bin	83.25	46.04	66.04	**65.97**	0.1186	[b]
	DE/best/1/bin	82.32	44.18	63.72	61.07	0.1099	[b]
	DSDE	65.58	58.14	62.79	62.14	**0.0257**	
bupaLD	DE/rand/1/bin	51.01	50.43	50.58	50.62	0.0021	[c]
	DE/best/1/bin	51.01	50.43	51.01	**50.79**	0.0028	[c]
	DSDE	51.01	50.43	50.43	50.46	**0.0013**	

[a] DSDE is better than its competitor by the Wilcoxon's rank sum test at α=0.05.

[b] There is no significant difference between DSDE and its competitor.

[c] DSDE is worse than its competitor.

1. For both DE/rand/1/bin and DE/best/1/bin, population size is set to 100 [42,49,51], scale factor *F* takes 0.5 [41,42,51], and crossover rate *Cr* is set to 0.9 [41,42,47-49,51].
2. For DSDE, there are two sub-populations, and each subpopulation size is set to 50. More over, mutation strategy DE/best/1 is used during the evolution process of each subpopulation. Accordingly, parameter settings of $F=0.5$ and $Cr=0.9$ are also used here.

5.2. Comparison among DE/rand/1/bin, DE/best/1/bin and DSDE

In order to evaluate the performance of DSDE, it is first compared with both DE/rand/1/bin and DE/best/1/bin. In the experiments, all approaches were run 20 times independently for each of the datasets. The results are presented in Table 2 in terms of the best, worst, median, mean and standard deviation (Std.) of the solutions obtained by each algorithm. In the eighth column of Table 2, the statistical significance level of the difference between the corresponding algorithm and the DSDE algorithm is also reported. That is, Wilcoxon's rank sum test [46] at a 0.05 significance level was conducted on the experiments. And then convergence curves of DE/rand/1/bin, DE/best/1/bin and DSDE are shown in Fig. 2 in order to show the convergence speed of DSDE more clearly. Last, the best clustering centers achieved by DSDE are also reported in Tables 4–6 in order to show the feasibility of results given in Table 2. The clustering centers can be used to validate the results in Table 2 by assigning any dataset to its corresponding clustering centers and then reaching a corresponding value given in Table 2.

Table 5
The best clustering center on the data sets Iris, Wine, CMC, crude Oil and Thyroid.

Data sets	Center1	Center2	Center3
Iris	5.012138686242361	6.733346741783071	5.934328024253308
	3.403101545488279	3.067850110992885	2.797799201326991
	1.471639046743056	5.630075113481303	4.417893164608254
	0.235406803776499	2.106798306955245	1.417266800307920
Wine	12.525522842313734	13.741668297437199	12.809976721739549
	2.322188155690992	1.862786379061949	2.542091132462463
	2.331462864620108	2.433160576176573	2.382322145893643
	21.325332639408678	16.921377235895356	19.506739284662256
	92.531514587541551	105.2804479073058	98.940794820320576
	2.036446041634795	2.860096411366974	2.063455772497936
	1.779209260688906	3.064716833165437	1.493369791970438
	0.409123596461618	0.293859784234457	0.427509368664548
	1.439367350194149	2.016455525973532	1.418263313904469
	4.355255798069449	5.698798165869451	5.780481715801642
	0.950659733613235	1.077965919011581	0.888392622707915
	2.461840514497851	3.024449089895384	2.222833830448662
	463.5998692309975	1137.272816317636	686.9668919771109
CMC	24.416794308274717	33.495271621709925	43.636950404337945
	3.042766122008858	3.133610566824130	3.004432394841012
	3.042766122008858	3.133610566824130	3.004432394841012
	3.512958106290536	3.554263665123835	3.454610437118063
	1.791987466310359	3.649066988852982	4.584795342459174
	0.928195507675319	0.789971345224440	0.795493483772608
	0.793959532247462	0.696752112246018	0.763680557832968
	2.302600209819029	2.098409830316567	1.822493028365769
	2.972493029384379	3.285850583844533	3.432925909755034
	0.035918936449945	0.059050743756431	0.089729012319805
Crude Oil	7.952803882991137	5.031403530240487	4.141865809296858
	17.818144669475657	32.108613664046359	46.117594289171116
	0.320997533123584	0.441579794762027	0.131798767444252
	4.372846935730347	5.762483664705491	6.731073351909815
	6.506980467396309	5.585603295576902	8.333174462132616
Thyroid	84.729560668885085	117.8168622827352	105.1608439377734
	18.781624089893459	9.395263961275477	9.182735419555405
	4.796517058239326	1.822031207306441	1.685517120514144
	0.972447388755393	1.995958984595695	1.233791319891045
	−0.021251732599964	3.708135800679412	1.881579623154738

Table 6
The best clustering center on the data sets Glass and Vowel.

Data sets	Center1	Center2	Center3	Center4	Center5	Center6
Glass	1.52627	1.52279	1.51516	1.52028	1.51978	1.51773
	13.01213	13.32149	12.85517	13.09923	13.80574	14.64658
	0.00188	3.58811	3.46384	0.23673	3.53328	0.06954
	3.02671	1.41896	1.32193	1.42930	0.94668	2.21689
	70.64063	72.66511	73.01082	72.68251	71.84998	73.25901
	6.19624	0.57449	0.58887	0.31357	0.16642	0.04435
	6.94893	8.20457	8.55687	11.96939	9.51825	8.70311
	0.00170	0.00361	0.00234	0.04167	0.04035	1.00826
	0.00938	0.03690	0.06990	0.06161	0.05065	0.01173
Vowel	357.26095	407.89375	375.45287	623.71776	439.24162	506.98782
	2291.43650	1018.05281	2149.40168	1309.59345	987.68666	1839.66010
	2977.39342	2317.82599	2678.44338	2333.45457	2665.47405	2556.19804

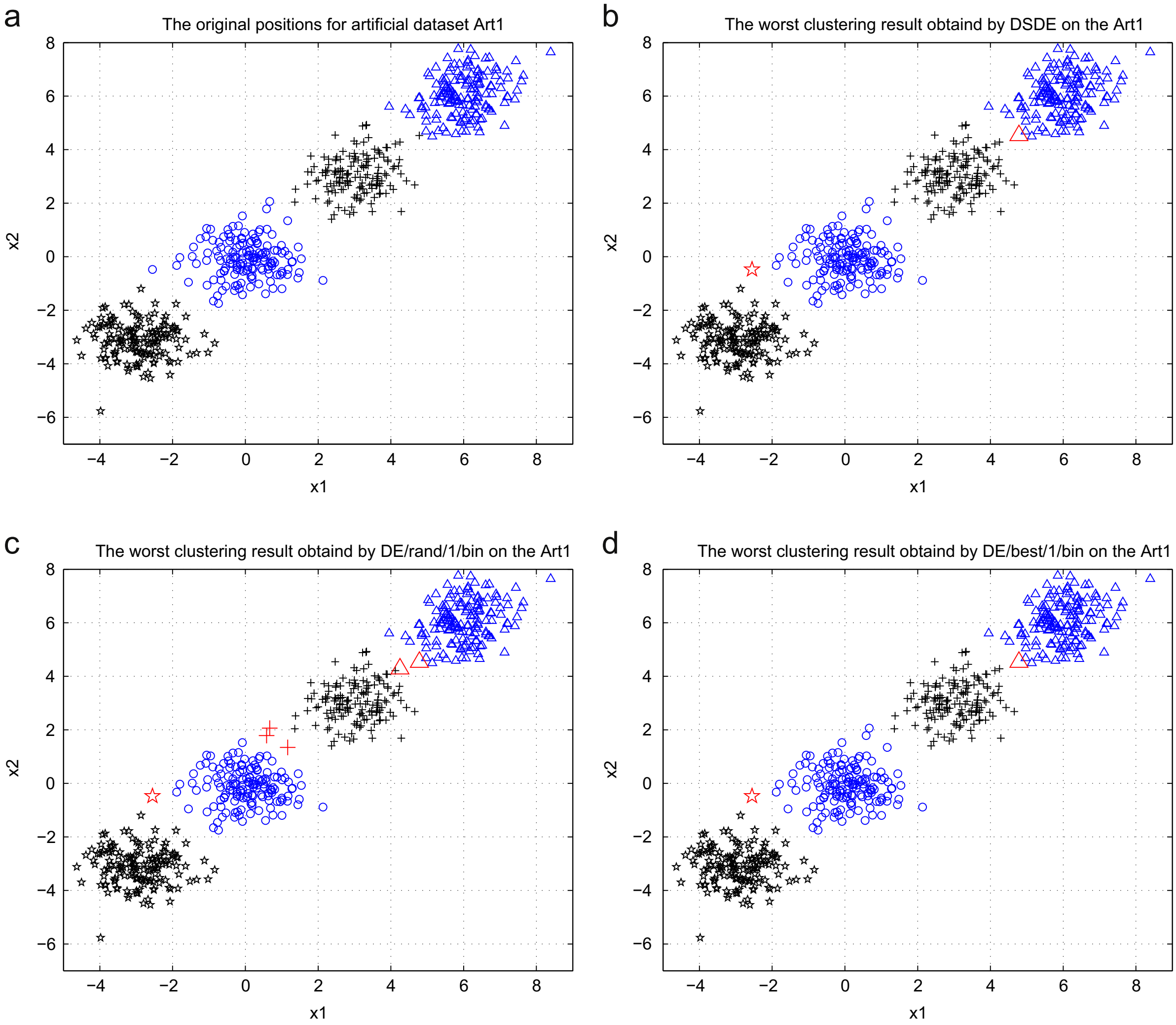


Fig. 3. The worst clustering results obtained by DE/rand/1/bin, DE/best/1/bin and DSDE on the artificial dataset Art1.

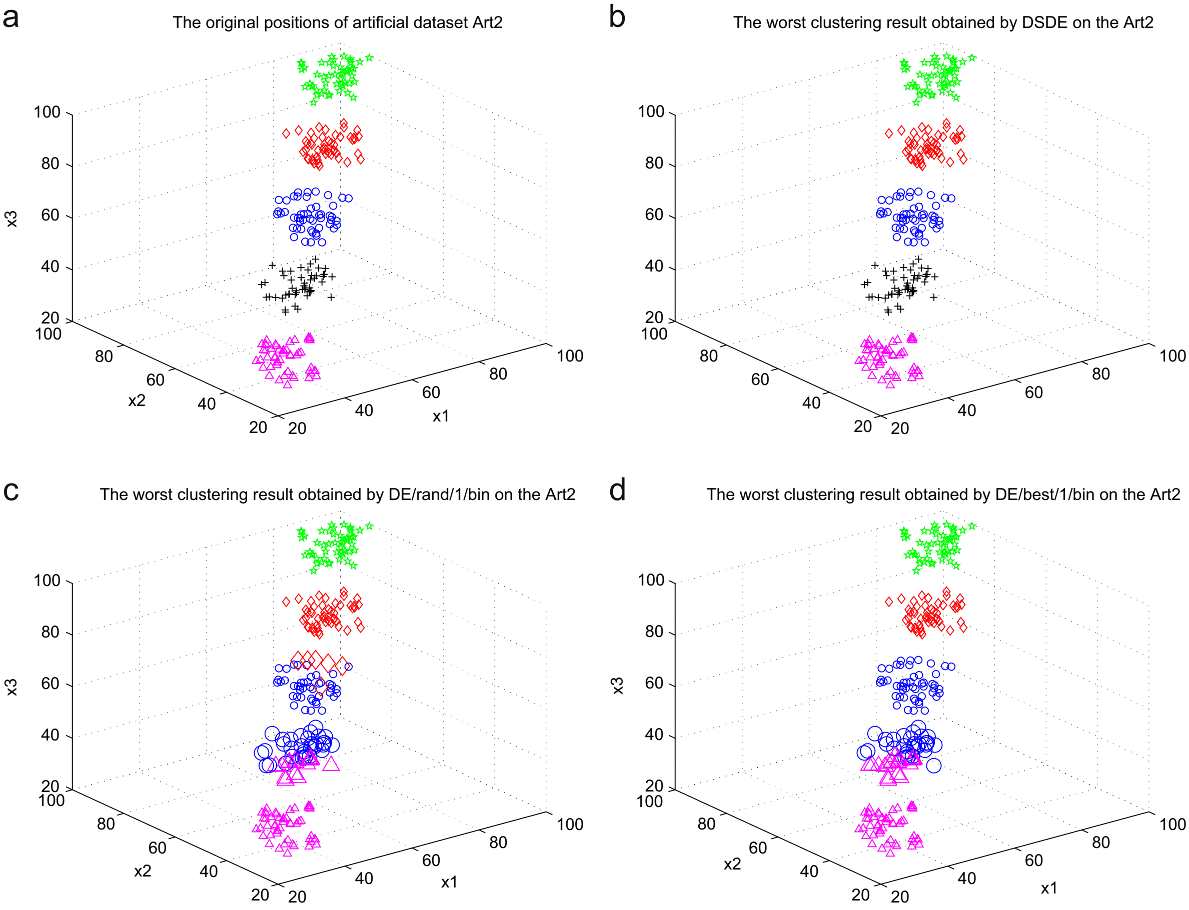
From the results given in Table 2, we can see that the results obtained by DSDE are significantly better than both DE/rand/1/bin and DE/best/1/bin for all the test datasets except for Art1. In particular, the mean values obtained by DSDE are better than those obtained by other two algorithms on almost all the test datasets. Meantime, the standard deviation values achieved by DSDE are also smaller than those found by other two algorithms, which suggests that DSDE is more effective and robust than DE/rand/1/bin and DE/best/1/bin on all the datasets. In a word, DE/best/1/bin is better than DE/rand/1/bin on all the datasets. However, DSDE is much better than DE/best/1/bin on almost all the datasets. All these superiorities of DSDE can also be seen from Fig. 2 and the eighth column of Table 2. Namely, the improved performance of our presented algorithm DSDE is very clear for data clustering.

In addition, the best, worst, median, mean and standard deviation of clustering accuracies obtained by the three algorithms for the 11 datasets are summarized in Table 3. We also report the statistical significance level of the difference of the clustering accuracy obtained by the corresponding algorithm and the proposed algorithm in the eighth column of Table 3. It can be seen that the DSDE algorithm had the best average clustering accuracy on nine test datasets, i.e., Art1, Art2, Iris, Wine, Glass, Cancer, CMC, Vowel and Crude Oil datasets. On the Thyroid dataset, the performance order of the algorithms is DE/rand/1/bin $>$ DSDE $>$ DE/best/1/bin in terms of the mean and standard deviation of clustering accuracies. On the bupaLD dataset, the mean clustering accuracy found by DSDE is slightly inferior to those found by other two algorithms. It should be noted that the best clustering accuracy obtained by DSDE is the same as that found by DE/best/1/bin on the bupaLD dataset. According to the eighth column of Table 3, i.e, the statistical significance test results, it can be found that DSDE is better than or at least equal to the other two algorithms on almost all the test datasets. Namely, the DSDE is worse than DE/rand/1/bin and DE/best/1/bin only on the dataset bupaLD. On the whole, the performance of DSDE is more better and stable than those of other two algorithms.

In order to further show the effectiveness of the DSDE algorithm, we visually illustrate the clustering results obtained by the three algorithms (DE/rand/1/bin, DE/best/1/bin and DSDE) on the two artificial datasets Art1 and Art2. The related results are given in Figs. 3 and 4, respectively.

The visual clustering results together with the original positions of data set Art1 are shown in Fig. 3. Specifically, the dataset Art1 is illustrated in Fig. 3(a), and the clustering results obtained by DSDE are illustrated in Fig. 3(b), and the clustering results searched by DE/rand/1/bin are illustrated in Fig. 3(c), and the clustering results found by DE/best/1/bin are illustrated in Fig. 3(d). It can be observed that two objects are misplaced by DSDE and DE/best/1/bin. However, six objects are misplaced by DE/rand/1/bin. Therefore, DSDE is better than or equal to the other two competitors on the artificial dataset Art1.

For artificial dataset Art2, it is illustrated in Fig. 4(a). The clustering results obtained by DSDE are illustrated in Fig. 4(b),


a The original positions of artificial dataset Art2

b The worst clustering result obtained by DSDE on the Art2

c The worst clustering result obtained by DE/rand/1/bin on the Art2

d The worst clustering result obtained by DE/best/1/bin on the Art2

Fig. 4. The worst clustering results obtained by DE/rand/1/bin, DE/best/1/bin and DSDE on the artificial dataset Art2.

Table 7
Comparison between DSDE and other population-based algorithms over 20 independent runs on the seven data sets.

Data sets	Indexes	ACO	ABC	PSO	PSOAG	DSDE
Iris	Mean	100.67	101.00	104.45	96.97	**96.65**
	Std.	1.58	1.43	4.77	0.35	**0**
Wine	Mean	16,300.71	16,506.75	16,303.16	16,296.30	**16,292.39**
	Std.	10.86	131.42	4.82	1.69	**0.24**
Glass	Mean	226.41	297.10	324.31	244.99	**212.73**
	Std.	4.64	9.15	12.63	10.41	**1.67**
Cancer	Mean	3376.20	3102.63	4024.79	2984.24	**2964.38**
	Std.	42.60	68.00	270.67	17.63	**0**
CMC	Mean	6151.35	5649.94	5750.07	5559.98	**5532.18**
	Std.	63.22	54.01	59.56	31.97	**0**
Vowel	Mean	170,849.03	160,347.29	154,017.66	149,734.40	**149,193.97**
	Std.	2055.93	3275.35	3722.14	988.20	**373. 44**
Thyroid	Mean	1950.37	2111.39	2369.90	1902.77	**1874.00**
	Std.	15.16	94.36	85.11	16.69	**11.76**

Bold entities mean the best results.

which is the same as shown in Fig. 4(a). This means that the DSDE algorithm exactly clusters the dataset Art2. The clustering results obtained by DE/rand/1/bin are illustrated in Fig. 4(c), where the bigger symbols represent those misplaced points. Thus, the clustering performance of DE/rand/1/bin is very bad on the dataset Art2 and its clustering accuracy is 77.60%. From Fig. 4(d), it can be seen that the clustering effect of DE/best/1/bin is slightly better than that of DE/rand/1/bin, which can also be found according to the worst clustering accuracy of Table 3. In general, the DSDE is superior to both DE/rand/1/bin and DE/best/1/bin on the Art2.

According to the analyses mentioned above, it can be concluded that the performance of DSDE is significantly enhanced relative to the canonical differential evolution.

5.3. Comparison between DSDE and other population based algorithms

In order to further testify the performance of DSDE, it is compared with other four well known algorithms, such as ACO, ABC, PSO, and PSOAG. Among them, PSOAG is a more recent algorithm, which was

Table 8
Comparison of clustering accuracy of the five algorithms

Data sets	Indexes	ACO	ABC	PSO	PSOAG	DSDE
Iris	Mean%	72.17	90.63	89.73	**91.03**	90.00
	Std.	3.55	1.63	2.28	1.27	**0**
Wine	Mean%	61.18	70.90	71.21	70.98	**71.65**
	Std.	3.13	0.92	0.36	0.33	**0.0028**
Glass	Mean%	37.90	48.57	49.18	51.26	**53.48**
	Std.	2.97	1.78	1.30	1.75	**0.0283**
Cancer	Mean%	78.23	95.51	94.36	96.31	**96.486**
	Std.	1.13	0.55	1.61	0.20	**0**
CMC	Mean%	36.96	**40.10**	39.80	39.87	38.49
	Std.	0.77	0.70	0.53	0.30	**0**
Vowel	Mean%	36.50	53.59	54.05	51.75	**56.21**
	Std.	1.78	4.75	4.01	4.25	**0.0122**
Thyroid	Mean%	51.93	60.67	62.93	**74.37**	62.14
	Std.	2.37	10.55	3.63	10.95	**0.0257**

Bold entities mean the best results.

Table 9
Comparison of objective values between DSDE-II and DSDE on the dataset Glass

Dataset	Index	DSDE-II	DSDE
Glass	Best	210.22	210.05
	Worst	222.19	215.64
	Median	212.95	213.00
	Mean	213.20	212.73
	Std.	2.96	1.6796

proposed by Jiang et al. [11] in 2013. For a reliable and fair comparison, the parameter *maxFEs* is set to 1e4, which is the same as in Jiang et al. [11]. The related comparison results are presented in Tables 7 and 8, where all results reported except for those obtained by DSDE are directly gained from Jiang et al. [11].

From Table 7, it can be observed that PSOAG is much better than ACO, ABC and PSO on almost all the cases. However, DSDE is obviously better than other four algorithms including PSOAG in all cases, which further verifies that our modifications to the original DE take effect. In other words, DSDE is an effective algorithm and it can be considered as a very good alternative for data clustering.

From Table 8, it can be found that the DSDE algorithm had the best mean accuracy on four of the seven datasets. The PSOAG algorithm obtained the best accuracy on two of the seven datasets. The ABC algorithm provided the higher accuracy on only one of seven datasets, i.e., the dataset CMC. The PSO algorithm also outperformed the DSDE algorithm on datasets CMC and Thyroid. Yet the ACO algorithm is inferior to the DSDE algorithm on all the seven datasets. Although DSDE outperformed the other four algorithms on the seven datasets in terms of objective function values as shown in Table 7, it failed to gain the same advantage on the seven datasets in terms of the clustering accuracy of Table 8. This is because that there is no absolute correlation between the objective function and the clustering accuracy [11]. This interesting phenomenon is also happened and reported in the literature [11]. Therefore, finding a good objective function is a hard and good way to improve the clustering accuracy of evolutionary algorithms.

5.4. *The effect of the sorting operation*

In order to further testify the effectiveness of the sorting operation of shuffled scheme, another comparison is done. For convenience, the DSDE without sorting operation is refer to as DSDE-II. For the sake of brevity, we take the sample dataset “Glass” as an example due to the complexity of the Glass dataset. The rest of parameter settings are also the same as those mentioned before. The comparison results are listed in Table 9.

From Table 9, it can be seen that the best result obtained by DSDE is slightly better than that found by DSDE-II on the dataset Glass. Especially, the worst result obtained by DSDE is obviously better than that found by DSDE-II. What is more, the mean result and its standard deviation value obtained by DSDE are also superior to those of DSDE-II. Here, only the median result obtained by DSDE is a little inferior to that found by DSDE-II. In a word, DSDE is better than DSDE-II. Namely, the sorting operation introduced into DSDE makes sense.

6. Conclusion

In this paper, we develop an enhanced differential evolution algorithm named as DSDE for data clustering. In DSDE, a novel initial technique which partly reduces the randomness in contrast with the *k*-means algorithm, a shuffled scheme for a total population, together with a double subpopulations scheme are presented and integrated. At the same time, mutation strategy DE/best/1 used in two subpopulations can effectively take use of the guidance information of the best individual. Double populations mechanism is employed to improve the population diversity and to avoid the premature convergence caused by the mutation strategy DE/best/1. At the same time, the sorting scheme and shuffled scheme inspired by SFLA further enhance the effect of improving the population diversity. Overall, the exploration ability and exploitation ability of DSDE are well balanced in the work. Then, experimental results tested on nine usually used datasets show that DSDE outperforms both DE/rand/1/bin and DE/best/1/bin in terms of their objective values. Subsequently, DSDE is compared with other four well-known algorithm including the recent algorithm PSOAG on seven datasets once more. At last, the related results also demonstrate its superiority over ACO, ABC, PSO and PSOAG. All these show that DSDE is a competitive approach for data clustering. Furthermore, we will focus on the theoretical analysis of the proposed algorithm complexity and convergence performance in the future work.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant no. 61263027) and the Youth Science Foundation of Lanzhou Jiaotong University (Grant no. 2012029) and the Science and Technology Foundation of Lanzhou Jiaotong University (Grant no. ZC2014010), and the New Teacher Project of Research Fund for the Doctoral Program of Higher Education of China (20126204120002).

References

[1] F.d.A. Guilherme, L.d.F. Costa, F.A. Rodrigues, A complex networks approach for data clustering, Phys. A: Stat. Mech. Appl. 391(23) (2012) 6174–6183.
[2] L.-Y. Chuang, C.-J. Hsiao, C.-H. Yang, Chaotic particle swarm optimization for data clustering, Expert Syst. Appl. 38 (12) (2011) 14555–14563.
[3] M. Wan, C. Wang, L. Li, et al., Chaotic ant swarm approach for data clustering, Appl. Soft Comput. 12 (8) (2012) 2387–2393.
[4] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, 1967, pp. 281–297.
[5] T. Kanungo, D.M. Mount, N.S. Netanyahu, et al., An efficient k-means clustering algorithm: analysis and implementation, IEEE Trans. Pattern Anal. Mach. Intell. 24 (7) (2002) 881–892.
[6] A.K. Jain, Data clustering: 50 years beyond K-means, Pattern Recognit. Lett. 31 (8) (2010) 651–666.
[7] B. Amiri, M. Fathian, A. Maroosi, Application of shuffled frog-leaping algorithm on clustering, Int. J. Adv. Manuf. Technol. 45 (12) (2009) 199–209.
[8] J.H. Ward, Hierarchical grouping to optimize an objective function, J. Am. Stat. Assoc. 58 (301) (1963) 236–244.
[9] B. Amiri, Application of teaching-learning-based optimization algorithm on cluster analysis, J. Basic Appl. Sci. Res. 2 (11) (2012) 11795–11802.
[10] S.Z. Selim, K. Alsultan, A simulated annealing algorithm for the clustering problem, Pattern Recognit. 24 (10) (1991) 1003–1008.

[11] B. Jiang, N. Wang, L. Wang, Particle swarm optimization with age-group topology for multimodal functions and data clustering, Commun. Nonlinear Sci. Numer. Simul. 18 (11) (2013) 3134–3145.

[12] K. Krishna, M.N. Murty, Genetic k-means algorithm, IEEE Trans. Syst. Man Cybern.—Part B: Cybernet. 29 (3) (1999) 433–439.

[13] Y. Marinakis, M. Marinaki, M. Doumpos, et al., A hybrid stochastic genetic-GRASP algorithm for clustering analysis, Oper. Res. 8 (1) (2008) 33–46.

[14] S. Bandyopadhyay, S. Saha, A point symmetry-based clustering technique for automatic evolution of clusters, IEEE Trans. Knowl. Data Eng. 20 (11) (2008) 1441–1456.

[15] S. Das, A. Abraham, A. Konar, Automatic clustering using an improved differential evolution algorithm, IEEE Trans. Syst. Man Cybern. Part A: Syst. Humans 38 (1) (2008) 218–237.

[16] S. Paterlini, T. Krink, Differential evolution and particle swarm optimisation in partitional clustering, Comput. Stat. Data Anal. 50 (5) (2006) 1220–1247.

[17] T. Cura, A particle swarm optimization approach to clustering, Expert Syst. Appl. 39 (1) (2012) 1582–1588.

[18] F. Yang, T. Sun, C. Zhang, An efficient hybrid data clustering method based on K-harmonic means and Particle Swarm Optimization, Expert Syst. Appl. 36 (6) (2009) 9847–9852.

[19] C.-Y. Tsai, I.W. Kao, Particle swarm optimization with selective particle regeneration for data clustering, Expert Syst. Appl. 38 (6) (2011) 6565–6576.

[20] L. Zhang, Q. Cao, A novel ant-based clustering algorithm using the kernel method, Inf. Sci. 181 (20) (2011) 4658–4672.

[21] A. Banharnsakun, B. Sirinaovakul, T. Achalakul, The best-so-far ABC with multiple patrilines for clustering problems, Neurocomputing 116 (2013) 355–366.

[22] X. Yan, Y. Zhu, W. Zou, et al., A new approach for data clustering using hybrid artificial bee colony algorithm, Neurocomputing 97 (2012) 241–250.

[23] F. Beloufa, M.A. Chikh, Design of fuzzy classifier for diabetes disease using Modified Artificial Bee Colony algorithm, Comput. Methods Prog. Biomed. 112 (1) (2013) 92–103.

[24] M. Taherdangkoo, M. Hossein Shirzadi, M. Yazdi, et al., A robust clustering method based on blind, naked mole-rats (BNMR) algorithm, Swarm Evol. Comput. 10 (2013) 1–11.

[25] M. Taherdangkoo, M.H. Bagheri, A powerful hybrid clustering method based on modified stem cells and fuzzy C-means algorithms, Eng. Appl. Artif. Intell. 26 (5–6) (2013) 1493–1502.

[26] A. Hatamlou, In search of optimal centroids on data clustering using a binary search algorithm, Pattern Recognit. Lett. 33 (13) (2012) 1756–1760.

[27] C. Granell, S. Gómez, A. Arenas, Unsupervised clustering analysis: a multiscale complex networks approach, Int. J. Bifurc. Chaos 22(07) (2012) 1230023(10 pages).

[28] J. Senthilnath, S.N. Omkar, V. Mani, Clustering using firefly algorithm: Performance study, Swarm Evol. Comput. 1 (3) (2011) 164–171.

[29] C. Granell, S. Gmez, A. Arenas, Data clustering using community detection, Int. J. Complex Syst. Sci. 1 (2011) (2011) 21–24.

[30] Â. Cardoso, A. Wichert, Iterative random projections for high-dimensional data clustering, Pattern Recognit. Lett. 33 (13) (2012) 1749–1755.

[31] T. Niknam, B. Amiri, An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis, Appl. Soft Comput. 10 (1) (2010) 183–197.

[32] T. Niknam, B. Amiri, J. Olamaei, et al., An efficient hybrid evolutionary optimization algorithm based on PSO and SA for clustering, J. Zhejiang Univ. Sci. A 10 (4) (2009) 512–519.

[33] X.-L. Yang, Q. Song, Y.-L. Wu, A robust deterministic annealing algorithm for data clustering, Data Knowl. Eng. 62 (1) (2007) 84–100.

[34] S. Bandyopadhyay, S. Saha, GAPS: a clustering method using a new point symmetry-based distance measure, Pattern Recognit. 40 (12) (2007) 3430–3451.

[35] S. Bandyopadhyay, U. Maulik, An evolutionary technique based on *K*-means algorithm for optimal clustering in R^n, Inf. Sci. 146 (2002) 221–237.

[36] Y.-T. Kao, E. Zahara, I.W. Kao, A hybridized approach to data clustering, Expert Syst. Appl. 34 (3) (2008) 1754–1762.

[37] Y. Yang, M.S. Kamel, An aggregated clustering approach using multi-ant colonies algorithms, Pattern Recognit. 39 (7) (2006) 1278–1289.

[38] M. Eusuff, K. Lansey, Optimization of water distribution network design using the shuffled frog leaping algorithm, J. Water Resour. Plan. Manag. 129 (3) (2003) 210–225.

[39] M. Eusuff, K. Lansey, F. Pasha, Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization, Eng. Optim. 38 (2) (2006) 129–154.

[40] R.A. Johnson, D.W. Wichern, Applied Multivariate Statistical Analysis, 5th ed., Prentice-Hall, Upper Saddle River, New Jersey 07458, 2001.

[41] R. Storn, K. Price, Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (1997) 341–359.

[42] S. Rahnamayan, H.R. Tizhoosh, M.M.A. Salama, Opposition-based differential evolution, IEEE Trans. Evol. Comput. 12 (1) (2008) 64–79.

[43] A.M. Bagirov, Modified global *K*-means algorithm for minimum sum-of-squares clustering problems, Pattern Recognit. 41 (10) (2008) 3192–3199.

[44] A. Hatamlou, S. Abdullah, H. Nezamabadi-pour, A combined approach for clustering based on *K*-means and gravitational search algorithms, Swarm Evol. Comput. 6 (2012) 47–52.

[45] Wikipedia. Arithmetic Mean, 2013, Available from: ⟨http://zh.wikipedia.org/wiki/%E7%AE%97%E6%9C%AF%E5%B9%B3%E5%9D%87%E6%95%B0⟩.

[46] J. Derrac, S. García, D. Molina, et al., A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (1) (2011) 3–18.

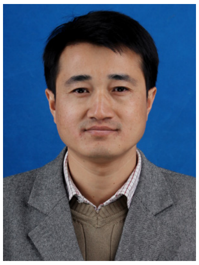
[47] J. Liu, J.A. Lampinen, Fuzzy adaptive differential evolution algorithm, Soft Comput. 9 (6) (2004) 448–462.

[48] J. Teo, Exploring dynamic self-adaptive populations in differential evolution, Soft Comput. 10 (8) (2006) 673–686.

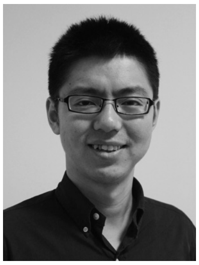
[49] W. Gong, Z. Cai, C.X. Ling, DE/BBO: a hybrid differential evolution with biogeography-based optimization for global numerical optimization, Soft Comput. 15 (4) (2010) 645–665.

[50] E.R. Hruschka, R.J.G.B. Campello, A.A. Freitas, et al., A survey of evolutionary algorithms for clustering, IEEE Trans. Syst. Man Cybern. Part C: Appl. Rev. 39 (2) (2009) 133–155.

[51] J. Brest, S. Greiner, B. Boî£ković, V. Žumer, Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems, IEEE Trans. Evol. Comput. 10 (6) (2006) 646–657.



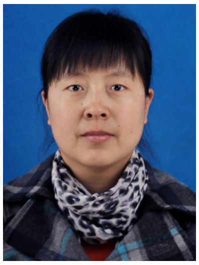
Wan-li Xiang received the Ph.D. degree from Tianjin University, Tianjin, China, in 2014. He is an Associate Professor with the School of Traffic and Transportation, Lanzhou Jiaotong University. His research interests include data mining, evolutionary computation and their applications to logistics and transportation.



Ning Zhu received the Ph.D. degree in Management Science and Engineering from Tianjin University, Tianjin, China, in 2012. He is an Assistant Professor with the Institute of Systems Engineering, Tianjin University. His research interests include intelligent algorithms, transportation management and planning, and intelligent transportation system and location problems in transportation.

Shou-feng Ma received the B.S., M.S., and Ph.D. degrees from Tianjin University, Tianjin, China, in 1988, 1991 and 1999, respectively. He is a Professor with the Institute of Systems Engineering, Tianjin University. His research interests include artificial intelligence, transportation management, traffic behavior and its impact on transportation networks, and traffic flow theory and its applications.

Xue-lei Meng received the Ph.D. degree in Planning and Management of Traffic and Transportation at the State Key Laboratory of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing, China, in 2011. He is currently an Associate Professor in Lanzhou Jiaotong University. His research interests are in the areas of train timetable design, optimization and evaluation, and line planning for railway.



Mei-qing An received the M.S. degree in Safety Technology and Engineering from Lanzhou Jiaotong University, Lanzhou, Gansu, China, in 2006. She is a Lecturer with the School of Traffic and Transportation, Lanzhou Jiaotong University. Her research interests include soft computing and logistics management.