Engineering Applications of Artificial Intelligence 61 (2017) 1–7

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

A novel data clustering algorithm based on modified gravitational search algorithm

XiaoHong Han*, Long Quan, XiaoYan Xiong, Matt Almeter, Jie Xiang, Yuan Lan

Key Laboratory of Advanced Transducers and Intelligent Control Systems, Ministry of Education China, Taiyuan University of Technology, Taiyuan, Shanxi, China

ARTICLE INFO

Keywords:
Gravitational search algorithm
Learning algorithm
Collective behavior
Data clustering
Clustering Validation
Nature-inspired algorithm

ABSTRACT

Data clustering is a popular analysis tool for data statistics in many fields such as pattern recognition, data mining, machine learning, image analysis, and bioinformatics. The aim of data clustering is to represent large datasets by a fewer number of prototypes or clusters, which brings simplicity in modeling data and thus plays a central role in the process of knowledge discovery and data mining. In this paper, a novel data clustering algorithm based on modified Gravitational Search Algorithm is proposed, which is called Bird Flock Gravitational Search Algorithm (BFGSA). The BFGSA introduces a new mechanism into GSA to add diversity, a mechanism which is inspired by the collective response behavior of birds. This mechanism performs its diversity enhancement through three main steps including initialization, identification of the nearest neighbors, and orientation change. The initialization is to generate candidate populations for the second steps and the orientation change updates the position of objects based on the nearest neighbors. Due to the collective response mechanism, the BFGSA explores a wider range of the search space and thus escapes suboptimal solutions. The performance of the proposed algorithm is evaluated through 13 real benchmark datasets from the well-known UCI Machine Learning Repository. Its performance is compared with the standard GSA, the Artificial Bee Colony (ABC), the Particle Swarm Optimization (PSO), the Firefly Algorithm (FA), K-means, and other four clustering algorithms from the literature. The simulation results indicate that the BFGSA can effectively be used for data clustering.

1. Introduction

Data clustering is one of the most important and popular data analysis techniques, which involves the process of classifying an unlabeled dataset into clusters of similar objects. Each cluster consists of objects that are similar within the cluster and dissimilar to objects of other clusters (Barbakh et al., 2009; Jain, 2010; Berikov, 2014). Clustering has been applied in many applications such as web mining, text mining, image processing, stock prediction, signal processing, biology and other fields of science and engineering (Everitt et al., 2011; Bishop, 2006).

There are many clustering algorithms that have been proposed for clustering problems in the literature. Traditional clustering algorithms fall into two main categories: hierarchical algorithms and partitional algorithms (Everitt et al., 2001; Xu and Wunsch, 2005).

Hierarchical algorithms create a tree structure of clusters in the absence of any prior knowledge about the number of clusters (Nanda and Panda, 2014). These algorithms can be carried out through two modes: agglomerative mode or divisive mode. In agglomerative mode, each object is regarded as a separate cluster in the beginning and then two most similar clusters are merged at each step. This process reoccurs until termination criteria are satisfied. In divisive mode, all objects are considered as one cluster in the beginning and then each cluster is divided into two clusters until termination criteria are met.

In partitional algorithms, each cluster is assigned initially a centroid. Then based on the similarity between each object and each centroid, all objects will be classified into a corresponding cluster. The objective of these algorithms is to maximize the similarity within one cluster while minimizing connectivity among different clusters (Everitt et al., 2001; Xu and Wunsch, 2005).

Apart from traditional clustering algorithms, there are other clustering algorithms (Chang et al., 2009). Ensity-based methods (Ankerst et al., 1999; Ester et al., 1996) and nearest neighbor methods (Lu and Fu, 1978) are based on the idea that neighbor objects ought to belong to the same cluster. Bi-clustering algorithms (Madeira and Oliveira, 2004) make their clustering through row and column simultaneously. Multi-objective clustering algorithms (Dehuri et al., 2006) and clustering ensembles approaches (Hong et al., 2008) are

* Corresponding author.
E-mail address: jmqchs@sohu.com (X. Han).

http://dx.doi.org/10.1016/j.engappai.2016.11.003
Received 30 March 2016; Received in revised form 27 September 2016; Accepted 7 November 2016

multi-objective clustering algorithms which optimize different characteristics of the dataset. Overlapping clustering algorithms are different from most of clustering algorithms, in which each object belongs to only one cluster. While in overlapping clustering, each object can belong to more than one cluster. Fuzzy C-means is one of the most popular overlapping clustering algorithms (Nanda and Panda, 2014).

In recent years, meta-heuristic algorithms are widely used to solve clustering problems (Nanda and Panda, 2014). From an optimization perspective, clustering problems can be formally considered as a particular kind of NP-hard grouping problem (Falkenauer, 1998). This type of algorithms includes searching for an optimal solution for clustering problems and reducing the risk of trapping in local optima. These algorithms include but not limited to genetic algorithms (GA) (Maulik and Bandyopadhyay, 2000), simulated annealing (SA) (Selim and Alsultan, 1991), Tabu search (Al-Sultan, 1995; Glover and Laguna, 1997), Artificial Bee Colony (ABC) (Karaboga and Ozturk, 2011a), Greedy Randomized Adaptive Search Procedure(GRASP) (Feo and Resende, 1989), Iterated Local Search(ILS) (Stutzle, 1999), Variable Neighborhood Search (VNS) (Mladenovic and Hansen, 1997), ant colony optimization (ACO) (Shelokar et al., 2004), Particle swarm optimization (PSO) (Chen and Ye, 2004; De et al., 2016), and so on.

Gravitational search algorithm (GSA) is one of the newest meta-heuristic optimization algorithms inspired by the Newtonian laws of gravity and motion (Rashedi et al., 2009). In GSA, an object in the search space attracts every other one with a force that is directly proportional to the product of their masses and inversely proportional to the square of the distance between them. The GSA has been proved to be an excellent optimization method for different types of applications, including data clustering (Dowlatshahi and Nezamabadi-pour, 2014), fuzzy model identification (Li et al., 2012), classification (Han et al., 2014; Zhang et al., 2013; Li et al., 2015), economic emission load dispatch (Jiang et al., 2014), wind turbine control (Chatterjee et al., 2014), and power systems (Shuaib et al., 2015).

Motivated by the success of the GSA with variant optimization problems, this paper proposes a novel data clustering algorithm based on a modified GSA (called Bird Flock GSA, BFGSA). The aim of the modified GSA is to enhance the capability of GSA in exploration without reducing the capability in exploitation. In the proposed BFGSA, we introduce bird flock behavior into GSA, which is a collective response process of how birds flock together. The concept of collective behavior of birds is inspired by the concept previously proposed (Hereford and Blum, 2011; Netjinda et al., 2015), which was used to enhance the original PSO. In their proposed method, the original velocity and position are updated by new equations which simulates the collective behavior of starlings. In this paper, we integrate collective response process of birds into GSA to enhance its performance. In our work, we maintain the original velocity and position updating equation of GSA most of the time, except when the global best stops in a local optimum, the position of objects are updated by the mechanism called bird flock behavior.

The rest of this paper is organized as follows. The cluster analysis problem is discussed in Section 2. Section 3 provides a review of standard GSA. The description of the proposed clustering algorithm is presented in Section 4. In Section 5, the experiments of the proposed algorithm for data clustering are given. Finally, a brief conclusion is offered in Section 6.

2. The data clustering problem

Data clustering is a process of classifying a set of objects into groups in which similarity in the same group must be maximized and objects that belong to different groups must be dissimilar as possible (Nanda and Panda, 2014; Hruschka et al., 2009).

Mathematically, a data set with N objects, each of which has d attributes, is denoted by X={X_1, X_2, ..., X_N}^{T}, where Xi={x_i^1, x_i^2, ..., x_i^d} is a vector denoting the i^{th} object and x_i^j is a scalar denoting the j^{th} attribute of x_i. The number of attributes is called the dimensionality of the data set. Let $\boldsymbol{X}_{n\times d}$ be the profile data matrix, with n rows and d columns. Given $\boldsymbol{X}_{n\times d}$, the goal of clustering is to classify X into K groups or clusters, C_1, C_2, ..., C_k, such that objects in the same cluster are as similar to each other as possible, while objects in different clusters are quite distinct. And also the following criteria should be satisfied (Nanda and Panda, 2014):

$$\bigcup_{i=1}^{K} C_i = X \tag{1}$$

$$C_i \bigcap C_j = \varnothing, \quad i, j = 1, \ldots, K; i \neq j \tag{2}$$

$$C_i \neq \varnothing, \quad i = 1, \ldots, K \tag{3}$$

Eq. (1) and Eq. (2) show that all objects in datasets must be classified and each object must belong to only one cluster, and Eq. (3) indicates that each cluster must contain objects.

To find optimal cluster centers with meta-heuristic algorithms, the objective function should be minimized. In this paper, we use quantization error formula as objective function and the optimization problem can be defined as follows:

$$J_e = \frac{\sum_{j=1}^{N_c}\left[\sum_{\forall z_p \in C_{ij}} d\,(z_{p,m_j})/|C_{ij}|\right]}{N_c} \tag{4}$$

in which $|C_{ij}|$ indicates the number of cluster C_{ij}; d indicates Euclidean distance between each data vector and the centroid. This Euclidean distance can be calculated by the following expression:

$$d\,(z_p, m_j) = \sqrt{\sum_{k=1}^{N_d} (z_{pk} - m_{jk})^2} \tag{5}$$

in which k indicates the dimension; N_d indicates the number of attributes of each data vector; z_p indicates p^{th} data vector and m_j indicates centroid vector of cluster j. The cluster centroid vectors are recalculated through the following expression:

$$m_j = \frac{1}{n_j} \sum_{\forall z_p \in C_j} z_p \tag{6}$$

in which n_j indicates the number of data vectors in cluster j and C_j indicates the subset of data vectors from cluster j.

3. The gravitational search algorithm

The gravitational search algorithm (GSA) is a newly developed stochastic population-base heuristic optimization algorithm based on the law of gravity and mass interactions, which was first introduced by Rashedi et al. Rashedi et al. (2009), Rashedi (2007), Rashedi et al. (2007). The algorithm provides an iterative process that simulates mass interactions, and develops through a multi-dimensional search space under the influence of gravitation. In GSA, all solutions are called agents or objects whose performances are evaluated by their masses; these agents or objects attract each other by gravitational force which causes a global movement of all objects towards objects with heavier masses (Rashedi et al., 2007, 2009; Rashedi, 2007).

Assumed there are k objects, the position of the ith object is defined as Eq. (7):

$$X_i = (x_i^1, \ldots, x_i^d, \ldots, x_i^n),\ i = 1, 2, \ldots, k, \tag{7}$$

where x_i^d denotes the position of ith object in the dth direction. The force exerting on the object i from the object j is defined as Eq. (8):

$$F_{ij}^d\,(t) = G\,(t)\frac{M_i\,(t)\times M_j\,(t)}{R_{ij}\,(t)+\varepsilon}[x_j^d\,(t) - x_i^d\,(t)], \tag{8}$$

where M_j is the mass related to object j, $M_i$is the mass related to object i, ε is a small constant, and $R_{ij}\,(t)$is the Euclidian distance between the object $i$and object j. G is a function of the initial value $G_0$and iteration$t$, which is defined as Eq. (9):

$$G(t) = G_0 e^{-\alpha\frac{t}{T}} \tag{9}$$

in which G_0 is initial gravitational constant and α is a specified constant by user, t is the current iteration and T is the maximum number of iterations. The total force $F_i^d(t)$ that exerts on object i in the dth direction is a randomly weighted sum of dth components of the forces from other objects:

$$F_i^d(t) = \sum_{j=1,j\neq i}^{k} rand_j F_{ij}^d(t), \tag{10}$$

where $rand_j$ is a uniform random variable in the interval [0,1].

The acceleration of the object $i$in the dth direction at time$t$ is $a_i^d(t)$, which is given as Eq. (11):

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)}, \tag{11}$$

where M_{ii} is the inertial mass of the object i. Its next velocity $v_i^d(t+1)$ and its next position $x_i^d(t+1)$are calculated according to Eqs. (12) and (13):

$$v_i^d(t+1) = rand_i \times v_i^d(t) + a_i^d(t) \tag{12}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{13}$$

where $rand_i$ is a uniform random variable in the interval [0,1]. This random number is used to give a randomized characteristic to the search, $v_i^d(t)$ and $x_i^d(t)$ are its current velocity and position, respectively.

The masses of objects are evaluated by the fitness function. Assuming the equality of the gravitational and inertia mass, the mass $M_i(t)$is updated according to Eqs. (15), (16), (17) and (18):

$$M_i = M_{ii},\ i = 1, 2, \ldots, k, \tag{14}$$

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)}, \tag{15}$$

$$M_i(t) = \frac{m_i(t)}{\sum_{j=1}^{k} m_j(t)}, \tag{16}$$

$$best(t) = \min_{j\in\{1,\ldots,k\}} fit_j(t), \quad \text{(for minimization problem)} \tag{17}$$

$$worst(t) = \max_{j\in\{1,\ldots,k\}} fit_j(t), \quad \text{(for minimization problem)} \tag{18}$$

where $fit_i(t)$ represents the fitness value of the object$i$ at time t. The flow chart of GSA is shown in Fig. 1.

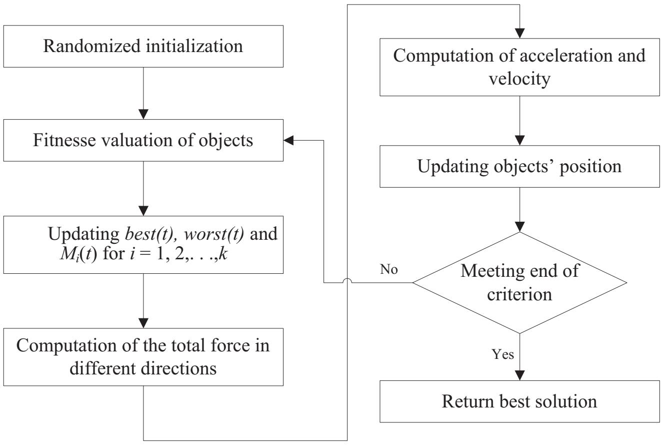
Randomized initialization
Fitnesse valuation of objects
Updating best(t), worst(t) and M_i(t) for i = 1, 2,. . .,k
Computation of the total force in different directions
Computation of acceleration and velocity
Updating objects' position
No
Meeting end of criterion
Yes
Return best solution

Fig. 1. The flow chart of GSA algorithm.

4. The GSA based clustering algorithm

Data clustering is one of the NP problems. GSA algorithm is an effective technique for solving optimization problems that works based on probability rules and population. So it is feasible to solve clustering problem using GSA. This view allows us to apply GSA algorithms for finding a set of candidate centroids and thus determining a near optimal classifying of the dataset at hand.

4.1. Solution encoding

In order to use GSA to solve clustering problems, the first step is to encode an appropriate solution encoding to encode cluster centers. Initially, candidate solutions for clustering problems are created randomly. Each of these candidate solutions (called mass or agent) denotes all centroids of datasets. After creating randomly candidate solutions, they will interact like masses in the universe through Newtonian gravitational law. The value of mass for each agent will be computed by objective function for that candidate solution. Good agents, which have less value for objective function, will have great masses and vice versa. In order to apply Newtonian gravitational law to cluster analysis, we have used arrays to encode cluster centers. If $X_{n\times d}$ is the profile matrix and k is the number of clusters G={C_1, C_2, ..., C_k} of the set of N data objects X={X_1, X_2, ..., X_N}^{T}, each candidate solution in the population consists of a one-dimensional vector of size $d\times k$, where k is the number of clusters and d is the number of attributes for each object in the dataset. Fig. 2 shows an example of a candidate solution for a problem with k clusters, where every data object has d attributes.

4.2. Position adjustment

The standard GSA sometimes has the problem of premature convergence due to rapid reduction of diversity. We introduce bird flock behavior into GSA to explore a wider range of the search space and thus escape suboptimal solutions.

In the standard GSA, each object can be specified by its position, inertial mass, active gravitational mass and passive gravitational mass. Taking into account the aspects of the above mentioned masses, Newton's law can be rewritten as follows.

$$F_{ij} = G(t)\frac{M_j \times M_i}{R^2} \tag{19}$$

$$\alpha_i = \frac{F_{ij}}{M_{ij}} \tag{20}$$

Considering a system with N agents, the position of the ith agent can be defined by

$$X_i = (x_i^1, \ldots, x_i^d, \ldots, x_i^n)\text{for i} = 1, 2, 3, \ldots, \text{N} \tag{21}$$

At a specific time 't', the force acting on mass 'i' from mass 'j' can be defined as

$$F_{ij}^d(t) = G(t)\frac{M_i(t)\times M_j(t)}{R_{ij}(t)+\varepsilon}[x_j^d(t) - x_i^d(t)] \tag{22}$$

The position and its velocity of an agent can be calculated as

$$x_i^d(t+1) = x_i^d(t)+v_i^d(t+1) \tag{23}$$

$$v_i^d(t) = rand_i \times v_i^d(t)+\alpha_i^d(t) \tag{24}$$

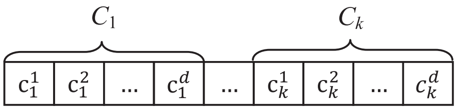

Fig. 2. Examples of a candidate solution encoding with k clusters and d attributes.

However the algorithm may be trapped in local optimum because of the rapid reduction of diversity. To escape the trap, we observe the value of the global best in each iteration to check whether it changes or not. If the global best fitness does not change for several subsequent iterations, we start a process of position movement to avoid the stagnation, which we call the collective response of object reorientation. To explore a wider range of position change, we use Eq. (25) to every object. In the Eq. (25), $\hat{x}_i$ is the new position of the object $\vec{x}_i$ after the collective response of object reorientation. The new position is obtained by calculating the position mean of its seven nearest neighbors. The seven nearest neighbors are chosen according to their distance measure. The selection of distance measure methods has certain influence on the results of algorithm. Therefore, it is necessary to select an appropriate distance measure according to the characteristics of the input data. In clustering analysis field, there are many widely used distance measure methods such as Euclidean distance, Manhattan distance, Mahalanob distance, and so on. Euclidean distance is a simple, widely used distance metric, which measures the absolute distance between points in a multidimensional space. It does not consider the correlation between components. In this case, we choose Euclidean distance as our distance measure method to compute the distance between different objects. The distances from $\vec{x}_i$to the seven objects are the shortest distances among all distances from $\vec{x}_i$ to other objects. The random number $rand_i$is a real number in the interval [−1,1]. The set$N_i$contains the indexes of the seven nearest neighbors of object $\vec{x}_i$.

$$\hat{x}_i = \vec{x}_i + rand_i\left(\frac{1}{7}\sum_{n\in N_i}\vec{x}_n\right) \tag{25}$$

This equation is used to update the position of each object. To increase the opportunity of obtaining a better position adjustment, we generate*ALTER_NUM*sets of candidate position updating solutions from the original solutions. *ALTER_NUM*, which is determined through many trials, is used to control the number of generated candidate sets. We select the alternative which yields the best global fitness as a new set of object positions.

The characteristics of collective response in the orientation change can be reflected by Eq. (25), in which $\frac{1}{7}\sum_{n\in N_k}\vec{x}_n$ describes the collective response of a bird's seven closest neighbors. New solutions are formed after the collective flock behavior of old solutions. The new global best solution is selected from these new positions. Algorithm 1 gives the collective response of the position change.

Algorithm 1. Collective response of position change for minimization problem.

```
N: the number of objects; Gbest: the optimal value; Lbest: the
optimal solution;
for i=1:N
    Find the nearest seven neighbors of object i using Euclidean
    distance and keep the index of neighbor (n) in the set $N_k$;
    Compute the new position through Eq.(25);
End
Gbest=the minimal fitness of all object fitness; Lbest=the best
solution of all solution;
```

4.3. The proposed clustering approach

Based on the above description, the main steps of our proposed BFGSA clustering algorithm are as follows:

Step 1: Generate randomly an initial population *P* which includes *S* candidate solutions: *P={P*$_1$*, P*$_2$*, ..., P*$_S$*}*, in which each candidate solution *Pi={Z1, Z2, ..., Zk}* includes *k* centroids, and each centroid has *d* attributes. $Zj=\{z_j^1, z_j^2, \ldots, z_j^d\}$ is the j^{th} centroid for i^{th} agent (*i*=1, 2, ..., *S* and *j*=1, 2, ..., *K*). *S* is the number of masses or agents or candidate solutions; *K* is the number of centroids; *d* is the number of attributes of each centroid.

Step 2: Calculate the fitness values according to Eq. (4) for all agents (candidate solutions). Choose the best candidate solution as the a final solution, which has the minimal fitness value. And choose the worst candidate solution with the maximal fitness value to calculate the individual masses. At a specific iteration *t* we have:

$$best(t) = min\{fit_j(t)\},$$
$$j\in\{1, 2, \ldots, S\};\ worst(t) = max\{fit_j(t)\},\quad j\in\{1, 2, \ldots, S\} \tag{26}$$

Step 3: Calculate the masses of agents based on the fitness function and *Gworst*:

$$M_i(t) = \frac{fit_i(t) - worst(t)}{\sum_{j=1}^{S}(fit_j(t) - worst(t))},\ i\ = 1, 2, \ldots, S \tag{27}$$

Step 4: Calculate the resultant force of selected agent from all other agents using gravity law. And then calculate the acceleration of each agent.

$$F_i^d(t) = \sum_{i\neq j} rand_j G(t)\frac{M_i(t)\times M_j(t)}{R_{ij}(t)+\varepsilon}[x_j^d(t) - x_i^d(t)] \tag{28}$$

$$\alpha_i(t) = \frac{F_i(\mathrm{t})}{M_i(t)},\qquad i = 1, 2, \ldots, S \tag{29}$$

where R_{ij} is Euclidian distance between two agents and x_i and x_j are the positions of i^{th} and j^{th} agents, respectively. ε is a very small constant to avoid division by zero.

Step 5: Update the velocity of agent and then update the position of agent which indicates the cluster centers by adding the new velocity.

$$v_i^d(t+1) = rand_i\times v_i^d(t)+\alpha_i^d(t),\qquad i\ = 1, 2, \ldots, S \tag{30}$$

$$x_i^d(t+1) = x_i^d(t)+v_i^d(t+1),\qquad i\ = 1, 2, \ldots, S \tag{31}$$

Step 6: Check if stagnation has occurred for a minimization problem. If the stagnation situation happens, then the population calls the collective response of the position change.

Step 7: If the termination criteria are satisfied, then the best individual as a final solution, which has the minimum value for objective function is returned. Otherwise go to step 2 and repeat.

The proposed BFGSA clustering algorithm is summarized in Algorithm 2.

Algorithm 2. The proposed BFGSA clustering algorithm.

```
Define initial parameters.
Initialize each agent with K random cluster centers.
for Iteration_count=1 to maximum_iterations do
    for all agents i do
    for all pattern $X_p$ in the dataset do
        calculate Euclidean distance of $X_p$ with all cluster
    centroids;
        assign $X_p$ to the cluster that have nearest centroid to $X_p$
    end for
    calculate the fitness function based on Eq.(4); calculate
    Gbest and Gworst based on Eq.(17) and Eq.(18);
    calculate mass value for all agents based on Eq.(27);
    calculate the acceleration and velocity of agents based on Eq. (29) and Eq.(30);
    calculate the position of each agent based on Eq.(31);
    if $stagnant\_count$ > STAGNANT_NUM % check if stagnation has occurred for minimization problem
        call the process of collective response of position change
    End
    end for
    find the global best position
    update the cluster centroids according to velocity updating and coordinate updating formula of GSA
end for
```

5. Experiments

In this section, the BFGSA clustering algorithm is evaluated on 13 real benchmark datasets from the UCI databases (Blake and Merz, 1998), which is a well-known database repository. The selected benchmark problems include examples with low, medium, and high dimensions.

The structure of this section is as follows. First, we describe the characteristics of the 13 selected standard classification datasets. Then, we present the comparison results of the BFGSA clustering algorithm with the other nine clustering algorithms.

5.1. Datasets description

In this paper, we use 13 benchmark classification datasets which are well-known and well-used datasets in the machine learning community. Table 1 provides the characteristics of these datasets including the number of instances, the number of features, and the number of classes. We select randomly 75% of each dataset and use it as a training set in the process of training. The remaining 25% of each dataset is used as a test set in the process of testing. Table 1 also offers the number of the training and testing sets. After training phase, we obtain the cluster centers as an extracted knowledge form training set, that can be used for classifying the test set.

5.2. Results and comparisons

In this section, the performance of the proposed BFGSA clustering algorithm is investigated by applying the proposed algorithm to solve different benchmark datasets. The proposed BFGSA clustering algorithm was implemented in MATLAB language and run on a PC with an Intel Core i5-4440 CPU @3.10 GHz and 8 GB memory. The population size was 50 and the max-number of iterations was set to 500. For the BFGSA clustering algorithm, the $stagnant\_count$(the number of the global best fitness not changing at continuous iterations.) was set to 2 and the number of candidate populations ($ALTER\_NUM$) in collective response was set to 14. The values of $stagnant\_count$ and $ALTER\_NUM$ were set based on a large number of trials and the stopping criterion is satisfied after 500 iterations. The algorithm was tested on a set of 13 well-known benchmark datasets. we compared the BFGSA results with a standard version of GSA (Bahrololoum et al., 2012), Artificial Bee Colony(ABC) (Karaboga and Ozturk, 2011b), K-means (Nanda and Panda, 2014), PSO (De Falco et al., 2007), NM-PSO (Fan et al., 2004), K-PSO (Kao et al., 2008), K-NM-PSO (Kao et al., 2008), CPSO (Chuang et al., 2011), and Firefly Algorithm (FA) (Senthilnath et al., 2011).

Table 1
Main characteristics of the 13 used benchmark datasets.

Dataset	# of data objects	# of training data	# of testing data	# of attributes	# of classes
Balance	625	469	156	4	3
Cancer	569	427	142	30	2
Cancer-Int	699	524	175	9	2
Credit	690	518	172	15	2
Dermatology	366	274	92	34	6
Diabetes	768	576	192	8	2
E. Coli	327	245	82	7	5
Glass	214	161	53	9	6
Heart	303	227	76	35	2
Horse	364	273	91	58	3
Iris	150	112	38	4	3
Thyroid	215	162	53	5	3
Wine	178	133	45	13	3

For each dataset, we report the error rate and the sum of the intra-cluster distances.

1. Error rate: the percentage of misclassification on the test set. The error rate is calculated as follows: first, the number of misclassifications is counted after the test data is classified. It is possible because the actual class label of each data instance is known in the test data set. Second, the number of misclassified instances is divided by total number of instances in the test set. The error rate is calculated through the following Eq. (32):

$$\text{Error Rate} = 100 \times \frac{\text{Number of misclassified instances}}{\text{Total size of test set}} \tag{32}$$

2. Sum of the intra-cluster distances: the distances between data vectors within a cluster and the centroid of the cluster, as defined in Eqs. (33) and (34). The less the sum of the intra-cluster distances is, the higher the quality of clustering results is.

$$D(x_p - z_j) = \sqrt{\sum_{i=1}^{d} (x_{pi} - z_{ji})^2} \tag{33}$$

$$z_j = \frac{1}{n_j} \sum_{\forall x_p \in c_j} x_p \tag{34}$$

In which z_j denotes the center vector of cluster j; x_p denotes the pth data vector; the d subscript represents the number of features of each center vector; n_j is the number of data vectors in cluster j, and C_j is the subset of data vectors that form cluster j.

Table 2 summarizes the intra-cluster distances obtained from the 10 clustering algorithms for the datasets in Table 1. The values reported are the averages of the sums of intra-cluster distances over 25 simulations. From Table 2, it can be seen that the test results of Cancer, Credit and Diabetes datasets indicate that K-PSO and K-NM-PSO outperforms the K-means method. K-PSO is a hybrid of the K-means and PSO algorithm. K-NM-PSO is a hybrid of the K-means, Nelder–Mead simplex search (Nelder and Mead, 1965) and PSO. CPSO has better results than NM-PSO, except for a tiny slight loss for Balance, Cancer-Int, and Dermatology datasets. As can be seen from these results, PSO offers better results than ABC and FA for Diabetes, E. Coli, Glass, Heart, and Iris datasets. For all experimental datasets except Horse, Thyroid, and Wine datasets, CPSO outperforms standard GSA, PSO, ABC, FA, and K-means. For Cancer, Cancer-Int, Dermatology and E. Coli datasets, the averages of CPSO are smaller than those of K-PSO and K-NM-PSO. In the Credit, Glass, Heart, and Horse datasets, the averages obtained with standard GSA are smaller than the ones obtained with ABC and K-means. For all experimental data sets, the BFGSA outperforms the other nine methods in terms of the averages of intra-cluster distances, which indicates that the BFGSA can be used as an efficient algorithm for data clustering.

Table 3 shows the mean error rate from the 25 simulation runs. For all data sets except the Balance, Cancer, Cancer-Int, Thyroid, Iris, and Wine data sets, CPSO exhibits a significantly smaller mean error rate compared to ABC, FA, K-means, PSO, NM-PSO and K-PSO. For the Credit, Dermatology, Diabetes, and Horse data sets, the mean error rate of CPSO is smaller than those of K-NM-PSO, standard GSA. For E.

Table 2
Average intra-cluster distances of each of the ten clustering algorithms BFGSA, standard GSA, PSO, ABC, FA, K-means, NM-PSO, K-PSO, K-NM-PSO, and CPSO executed on 13 UCI datasets.

Dataset	BFGSA	Standard GSA	PSO	ABC	FA	K-means	NM-PSO	K-PSO	K-NM-PSO	CPSO
Balance	**10052.32**	131293.30	61987.01	66329.16	37640.44	20137.76	57313.68	10966.71	17153.45	18167.91
Cancer	**21.38**	34.89	169.88	107.39	126.67	90.25	163.94	29.62	195.20	22.10
Cancer-Int	**38.34**	260.61	230.30	170.08	117.52	148.82	267.76	272.94	150.33	149.38
Credit	**1569.90**	1398.57	5090.77	4256.66	4036.46	3349.21	2773.60	1973.18	4194.19	2037.86
Dermatology	**102.78**	654.98	1862.78	796.85	1448.48	442.87	210.33	742.20	555.49	183.03
Diabetes	**113.93**	6747.03	3140.59	8210.04	5488.42	156.52	947.84	5148.62	9525.25	131.90
E. Coli	**32.93**	571.97	40.37	276.32	57.78	675.21	617.95	408.93	566.36	33.34
Glass	**107.84**	166.49	169.76	477.17	376.69	167.91	317.16	208.42	289.04	120.22
Heart	**1371.61**	4350.15	2298.83	8043.72	9095.10	10284.16	5707.29	2144.42	6693.52	2100.27
Horse	**6.41**	14.85	8.42	71.92	94.37	19.69	66.58	15.10	70.41	63.92
Iris	**67.04**	520.79	138.98	265.43	651.34	631.04	100.22	126.58	332.60	90.25
Thyroid	**1124.65**	8255.67	4577.00	5635.12	1432.76	2385.84	3251.74	4995.94	2668.91	2923.22
Wine	**70.89**	234.58	175.93	151.61	264.25	118.10	73.54	85.44	272.69	211.75

Coli, Glass, and Heart data sets, it is equal to those of ABC and FA. For all the data sets except Cancer-Int, Heart, and Credit data sets, the BFGSA exhibits a significantly smaller mean error rate compared to ABC, FA, standard GSA, K-means, PSO, NM-PSO, K-PSO, K-NM-PSO, and CPSO. Again, the BFGSA is superior to the other nine algorithms with respect to the intra-cluster distance. Although the BFGSA in the E. Coli and Heart data sets does not obtain the best error rate, the intra-cluster distance is the smallest (Table 2). It should be noted that the intra-cluster distance is not proportional to the error rate (Kao et al., 2008). The actual data distribution is not regular and therefore a smaller intra-cluster distance does not necessarily indicate a lower error rate.

To be able to make a good comparison among ten clustering algorithms, The ranking of the BFGSA and the other nine clustering algorithms based on their error rates is reported, which can be seen in Table 4. This table shows the average error rates of all datasets obtained by the BFGSA and the other nine clustering algorithms. The ranking is ordered through the ascending sequence of average error rates.

To statistically compare the performance differences among the BFGSA and the other 9 clustering algorithms, A Wilcoxon signed-rank test (Derrac et al., 2011) is conducted. Table 5 shows the resultant p-values of comparing the BFGSA with the other 9 clustering algorithms on the 13 benchmark datasets. From Table 5 therefore, it can be seen that the BFGSA is better than the seven clustering algorithms with a level of significance of $\alpha = 0.05$. It also can be seen that the BFGSA has an improvement over the six clustering algorithms with a level of significance of $\alpha = 0.01$.

Table 4
Ranking of the BFGSA and 9 clustering algorithms based on their error rates.

Rank	Clustering algorithms	Average (%)
1	BFGSA	8.90
2	CPSO	16.54
3	PSO	16.67
4	Standard GSA	18.19
5	K-PSO	18.36
6	FA	19.37
7	K-NM-PSO	20.04
8	ABC	20.35
9	K-means	23.60
10	NM-PSO	24.27

Table 5
Wilcoxon signed-rank test between BFGSA with other 9 clustering algorithms on benchmark datasets.

BFGSA vs.	p-value
Standard GSA	4.62–02
PSO	4.31–04
ABC	2.29–03
FA	5.07–03
K-means	1.17–03
NM-PSO	6.72–03
K-PSO	1.08–03
K-NM-PSO	8.91–02
CPSO	8.42–02

Table 3
Average error rate of each of the ten clustering algorithms BFGSA, standard GSA, PSO, ABC, FA, K-means, NM-PSO, K-PSO, K-NM-PSO, and CPSO executed on 13 UCI datasets.

Dataset	BFGSA	Standard GSA	PSO	ABC	FA	K-means	NM-PSO	K-PSO	K-NM-PSO	CPSO
Balance	**2.96**	3.38	11.34	24.47	2.59	15.11	11.86	36.87	24.19	24.48
Cancer	**5.84**	12.72	13.74	9.19	29.44	14.62	18.76	6.37	11.77	34.38
Cancer-Int	39.00	10.09	8.41	34.23	7.92	26.06	24.87	**5.76**	25.86	23.06
Credit	30.63	14.66	18.05	17.78	24.09	27.67	36.87	36.16	33.33	**14.21**
Dermatology	**3.35**	34.29	14.39	22.61	25.37	40.47	9.17	27.51	5.13	4.54
Diabetes	**1.98**	7.59	12.47	22.14	31.54	38.30	39.85	2.62	30.76	2.01
E. Coli	**2.54**	7.15	38.82	8.71	8.71	15.51	18.95	21.71	26.31	2.90
Glass	**4.00**	21.43	16.30	35.90	35.90	8.70	32.77	36.30	19.25	4.05
Heart	15.51	25.01	16.82	19.46	19.46	37.81	22.87	16.26	38.20	**1.44**
Horse	**1.88**	7.22	24.56	2.70	16.90	12.31	14.06	12.91	4.06	2.53
Iris	**2.14**	26.26	3.05	18.40	14.63	23.31	18.64	10.61	24.55	39.13
Thyroid	**8.43**	34.51	9.26	23.01	23.01	37.34	29.32	9.09	10.40	30.54
Wine	**6.05**	32.21	29.58	26.07	12.31	9.66	37.64	16.62	6.75	31.85
Average	9.56	18.19	16.67	20.35	19.37	23.60	24.27	18.36	20.04	16.54
Rank	1	4	3	8	6	9	10	5	7	2

6. Conclusions

Clustering algorithms have emerged and rapidly developed as an alternative powerful meta-learning tool to undertake a broad range of applications because it is particularly useful for segmenting large multidimensional data into distinguishable representative clusters. In this paper, we propose a data clustering algorithm based on a modified gravitational search algorithm. The proposed clustering algorithm is called Bird Flock Gravitational Search Algorithm (BFGSA), inspired by the collective response of birds. In our work, we maintain the original position equation of GSA most of the time, only when the global best stops in a local optimum will we call the mechanism of bird flock response. The use of the position updating of bird flock response is to solve the problem of premature convergence of standard GSA due to rapid reduction of diversity. We employ the BFGSA for evolving a set of candidate cluster centroids and thus determining robust data cluster centers in a multi-dimensional Euclidean space. The performance of the proposed algorithm is evaluated in terms of the error rate and the sum of the intra-cluster distances over 13 well-known benchmark data sets. Its performance is compared with the K-means, the Particle Swarm Optimization, the standard GSA, the Firefly Algorithm, and the other five clustering algorithms from the literature. The experimental results confirm the effectiveness of the proposed algorithm and show that it can successfully be applied to data clustering.

Acknowledgements

This research has been supported by Natural Science Foundation of Shanxi Province of China (2014011021-1 and 2014021024-1) and National Natural Science Foundation program of China (51405327).

References

Al-Sultan, K.S., 1995. A tabu search approach to the clustering problem. Pattern Recognit. 28, 1443–1451.

Ankerst, M., Breunig, M., Kriegel, H.P., Sander, J., 1999. OPTICS: ordering points to identify the clustering structure. In: Proceedings of the International Conference on Management Data, pp. 49–60.

Bahrololoum, A., Nezamabadi-pour, H., Bahrololoum, H., Saeed, M., 2012. A prototype classifier based on gravitational search algorithm. Appl. Soft Comput. 12, 819–825, (GSA).

Barbakh, W., Wu, Y., Fyfe, C., 2009. Review of clustering algorithms. In: Non-Standard Parameter Adaptation for Exploratory Data Analysis. Springer, Berlin/Heidelberg, 7–28.

Berikov, V., 2014. Weighted ensemble of algorithms for complex data clustering. Pattern Recognit. Lett. 38, 99–106.

Bishop, C.M., 2006. Pattern recognition and machine learning vol. 4. springer, New York.

Blake, C.L., Merz, C.J., 1998. UCI repository of machine learning databases ⟨http://www.ics.uci.edu/mlearn/MLRepository.html⟩.

Chang, D.X., Zhang, X.D., Zheng, C.W., 2009. A genetic algorithm with gene rearrangement for K-means clustering. Pattern Recognit. 42, 1210–1222.

Chatterjee, A., Roy, K., Chatterjee, D., 2014. A gravitational search algorithm (GSA) based photo-voltaic (PV) excitation control strategy for single phase operation of three phase wind-turbine coupled induction generator. Energy 74 (1), 707–718.

Chen, C.-Y., Ye, F., 2004. Particle swarm optimization algorithm and its application to clustering analysis," in Networking, Sensing and Control, 2004 IEEE International Conference on, 2004, pp. 789-794.

Chuang, L.-Y., Hsiao, C.-J., Yang, C.-H., 2011. Chaotic particle swarm optimization for data clustering. Expert Syst. Appl. 38 (12), 14555–14563.

Dehuri, S., Ghosh, A., Mall, R., 2006. Genetic algorithms for multi-criterion classification and clustering in data mining. Int. J. Comput. Inf. Syst. 4 (3), 143–154.

Derrac, J., Garcia, S., Molina, D., Herrera, F., 2011. A practical tutorial on the use of non parametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evolut. Comput. 1 (1), 3–18.

De, A., Mamanduru, V.K.R., Gunasekaran, A., Subramanian, N., Tiwari, M.K., 2016. Composite particle algorithm for sustainable integrated dynamic ship routing and scheduling optimization. Comput. Ind. Eng. 96, 201–215.

De Falco, I., Della Cioppa, A., Tarantino, E., 2007. Facing classification problems with particle swarm optimization. Appl. Soft Comput. 7 (3), 652–658.

Dowlatshahi, M.B., Nezamabadi-pour, H., 2014. GGSA: a Grouping Gravitational Search Algorithm for data clustering. Eng. Appl. Artif. Intell. 36, 114–121.

Ester, M., Kriegel, H.P., Sander, J., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, pp. 226–231.

Everitt, B.S., Landau, S., Leese, M., Stahl, D., 2011. cluster analysis 5. wiley series in probability and statistics.

Everitt, B., Landau, S., Leese, M., 2001. Cluster Analysis. Arnold, London.

Falkenauer, E., 1998. Genetic Algorithms and Grouping Problems. Wiley, NewYork.

Fan, S.-K.S., Liang, Y.-C., Zahara, E., 2004. Hybrid simplex search and particle swarm optimization for the global optimization of multimodal functions. Eng. Optim. 36, 401–418.

Feo, T.A., Resende, M.G.C., 1989. A probabilistic heuristic for a computationally difficult set covering problem. Oper. Res. Lett. 8, 67–71.

Glover, F., Laguna, M., 1997. Tabu Search. Kluwer Academic Publishers.

Han, X., Chang, X., Quan, L., et al., 2014. Feature subset selection by gravitational search algorithm optimization. Inf. Sci. 281, 128–146.

Hereford, J., Blum, C., 2011. Flock Opt: a new swarm optimization algorithm based on collective behavior of starling birds, in: 2011 Third World Congress on Nature and Biologically Inspired Computing (NaBIC).

Hong, Y., Kwong, S., Chang, Y.C., Ren, Q.S., 2008. Unsupervised feature selection using clustering ensembles and population based incremental learning algorithm. Pattern Recognit. 41, 2742–2756.

Hruschka, E.R., Campello, R.J.G.B., Freitas, A.A., De Carvalho, A.P.L.F., 2009. A survey of evolutionary algorithms for clustering, systems, man, and Cybernetics, Part C: applications and Reviews. IEEE Trans. on 39, 133–155.

Jain, A.K., 2010. Data clustering: 50 years beyond K-means. Pattern Recognit. Lett. 31, 651–666.

Jiang, S., Ji, Z., Shen, Y., 2014. A novel hybrid particle swarm optimization and gravitational search algorithm for solving economic emission load dispatch problems with various practical constraints. Int. J. Electr. Power Energy Syst. 55, 628–644.

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

Karaboga, D., Ozturk, C., 2011a. A novel clustering approach: artificial bee colony (ABC) algorithm. Appl. Soft Comput. 11, 652–657.

Karaboga, D., Ozturk, C., 2011b. A novel cluster approach: artificial bee colony(ABC) algorithm. Appl. Soft Comput. 11, 652–657.

Li, C., An, X., Li, R., 2015. A chaos embedded GSA-SVM hybrid system for classification. Neural Comput. Appl. 26 (3), 713–721.

Li, C., Zhou, J., Fu, B., et al., 2012. T–S fuzzy model identification with gravitational search based hyper-plane clustering algorithm. IEEE Trans. Fuzzy Syst. 20 (2), 305–317.

Lu, S.Y., Fu, K.S., 1978. A sentence-to-sentence clustering procedure for pattern analysis. IEEE Trans. Syst. Man Cybern. 8, 381–389.

Madeira, S.C., Oliveira, A.L., 2004. Biclustering algorithms for biological data analysis:asurvey. IEEETrans. Comput. Bioinf. 1 (1), 24–45.

Maulik, U., Bandyopadhyay, S., 2000. Genetic algorithm-based clustering technique. Pattern Recognit. 33, 1455–1465.

Mladenovic, M., Hansen, P., 1997. Variable neighborhood search. Comput. Oper. Res. 24, 1097–1100.

Nanda, S.J., Panda, G., 2014. A survey on nature inspired meta-heuristic algorithms for partitional clustering. Swarm Evolut. Comput. 16, 1–18.

Nelder, J.A., Mead, R., 1965. A simplex method for function minimization. Comput. J. 7, 308–313.

Netjinda, N., Achalakul, T., Sirinaovakul, B., 2015. Particle Swarm Optimization inspired by starling flock behavior. Appl. Soft Comput. 35, 411–422.

Rashedi, E., Nezamabadi-pour, H., Saryazdi, S., 2009. GSA: a gravitational search algorithm. Inf. Sci. 179, 2232–2248.

Rashedi, E., 2007. Gravitational search algorithm (M.Sc. Thesis). Electrical Engineering Department, Shahid Bahonar University of Kerman, Iran.

Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S., 2007. Allocation of static var compensator using gravitational search algorithm. Proceedings of the First Joint Conference on Fuzzy and Intelligent Systems. Mashhad, Iran.

Selim, S.Z., Alsultan, K., 1991. A simulated annealing algorithm for the clustering problem. Pattern Recognit. 24, 1003–1008.

Senthilnath, J., Omkar, S.N., Mani, V., 2011. Clustering using firefly algorithm: performance study. SwarmEvolut. Comput. 1, 164–171.

Shelokar, P., Jayaraman, V.K., Kulkarni, B.D., 2004. An ant colony approach for clustering. Anal. Chim. Acta 509, 187–195.

Shuaib, Y.M., Kalavathi, M.S., Rajan, C.C.A., 2015. Optimal capacitor placement in radial distribution system using gravitational search algorithm. Int. J. Electr. Power Energy Syst. 64, 384–397.

Stutzle, T., 1999. Local Search Algorithms for Combinatorial Problems: Analysis, Algorithms and New Applications (Ph.D.thesis). DISKI—Dissertationenzur Kunstliken Intelligenz, Sankt Augustin, Germany.

Xu, R., Wunsch, D., 2005. Survey of clustering algorithms. IEEE Trans. Neural Netw. 16 (3), 645–678.

Zhang, W., Niu, P., Li, G., et al., 2013. Forecasting of turbine heat rate with online least squares support vector machine based on gravitational search algorithm. Knowl. Based Syst. 39, 34–44.