IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 25, 2020, accepted June 30, 2020, date of publication July 6, 2020, date of current version July 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3007498

A Novel Hybrid PSO-K-Means Clustering Algorithm Using Gaussian Estimation of Distribution Method and Lévy Flight

HANJIE GAO[1], YINTONG LI[2], PETR KABALYANTS[3, 6], HAO XU[1,5], AND RODRIGO MARTÍNEZ-BÉJAR[4]

[1]College of Computer Science and Technology, Jilin University, Changchun 130012, China
[2]College of Aeronautics Engineering, Air Force Engineering University, Xi'an 710038, China
[3]Mathematics and Computer Science School, V. N. Karazin Kharkiv National University, 61022 Kharkiv, Ukraine
[4]Department of Information Engineering and Communications, Faculty of Informatics, University of Murcia, 30100 Murcia, Spain
[5]Department of Computer Science and Technology, Zhuhai College of Jilin University, Zhuhai 519041, China
[6]Department of Software Engineering for Computers and Computer-Based Systems, Belgorod V G Shukhov State Technology University, 308012 Belgorod, Russia

Corresponding author: Hao Xu (xuhao@jlu.edu.cn)

This work was supported in part by the Development Project of Jilin Province of China under Grant 20170203002GX and Grant 20170101006JC, in part by the Ministry of Science and Technology of the People's Republic of China under Grant 2018YFC2002500, in part by the Jilin Province Development and Reform Commission, China, under Grant 2019C053-1, in part by the Education Department of Jilin Province, China, under Grant JJKH20200993K, and in part by the Department of Science and Technology of Jilin Province, China, under Grant 20200801002GH.

ABSTRACT Clustering is an important data analysis technique, which has been applied to many practical scenarios. However, many partitioning based clustering algorithms are sensitive to the initial state of cluster centroids, may get trapped in a local optimum, and have poor robustness. In recent years, particle swarm optimization (PSO) has been regarded as an effective solution to the problem. However, it has the possibility of converging to a local optimum, especially when solving complex problems. In this paper, we propose a hybrid PSO-K-means algorithm, which uses the Gaussian estimation of distribution method (GEDM) to assist PSO in updating the population information and adopts Lévy flight to escape from the local optimum. The proposed algorithm is named a GEDM and Lévy flight based PSO-K-means (GLPSOK) clustering algorithm. Firstly, during initialization, a few particles are initialized using the cluster centroids generated by K-means, while other particles are randomly initialized in the search space. Secondly, GEDM and PSO are selected with different probability to update the population information at different optimization stages. Thirdly, Lévy flight is adopted to help the search escape from the local optimum. Finally, the greedy strategy is carried out to select the promising particles from the parents and the newly generated candidates. Experimental results on both synthetic data sets and real-world data sets show that the proposed algorithm can produce better clustering results and is more robust than existing classic or state-of-the-art clustering algorithms.

INDEX TERMS Data clustering, K-means, particle swarm optimization, Gaussian estimation of distribution method, Lévy flight.

I. INTRODUCTION

In recent years, data mining is widely used to find useful patterns and knowledge which are hidden inside Large-scale data from different sources [1]. Clustering is one of the research hotspots in the field of data mining. It is an unsupervised learning method and aims to group the objects based on the principle of maximizing the intra-cluster similarity and minimizing the inter-cluster similarity [2]. In the past few decades, various clustering algorithms [3]–[6] have been proposed. So far, clustering has been used in a number of applications, such as image segmentation [7], bioinformatics–gene expression data analysis [8], and object recognition [9]. Partitioning based clustering algorithms, such as K-means [10], Fuzzy C-means (FCM) [6], and K-Harmonic Means (KHM) [11] are widely used because of their simplicity and efficiency.

The associate editor coordinating the review of this manuscript and approving it for publication was N. Ramesh Babu.

However, these algorithms also present some serious inconveniences, i.e., they are sensitive to the initial state of cluster centroids, may get trapped in a local optimum, and have poor robustness. The above inconveniences often lead to unsatisfactory clustering results. Since clustering tasks can be modeled as optimization problems, it is a natural choice to solve the clustering problems with optimization algorithms. So far, many intelligent optimization algorithms such as particle swarm optimization (PSO) [12], [13], ant colony optimization (ACO) [14], cuckoo search algorithm (CSA) [15], genetic algorithm (GA) [16], and differential evolution (DE) [17] have been regarded as effective solutions to clustering problems.

Among all the optimization algorithms, PSO has received much attention owing to its simplicity and competitiveness in finding a better solution [18]. In 2015, Liang *et al*. [19] proposed an adaptive clustering-based PSO, which considered the population topology and individual behavior control together to balance local and global search in an optimization process and proved the superiority of the algorithm. Furthermore, among the clustering approaches based on optimization algorithms, PSO has proven to be a strong competitor [20]. For example, the PSO-based clustering technique was firstly proposed in [21], where the initial swarm was fed by the clustering results of K-means. In 2011, Izakian and Abraham [22] proposed a clustering algorithm based on FCM and PSO, which applied FCM to the particles in each generation to improve the fitness of each particle. However, as with almost all optimization algorithms, the basic PSO algorithm also has the possibility of converging to a local optimum, especially when solving complex multimodal problems. In addition, most PSO-based clustering algorithms have the problem of low convergence efficiency.

K-means is one of the most classic clustering algorithms and is widely used for its simplicity and low computation cost. PSO is an effective global optimization algorithm and has a strong ability to search for solutions. To take full advantage of both algorithms and solve the above problems, we propose a hybrid PSO-K-means algorithm, which adopts the Gaussian estimation of distribution method (GEDM) and Lévy flight to improve the performance of the algorithm. The proposed algorithm is named a GEDM and Lévy flight based PSO-K-means (GLPSOK) clustering algorithm. Firstly, during initialization, a few particles are initialized using the cluster centroids generated by K-means to ensure that there are some relatively good particles in the initial swarm, while other particles are randomly initialized in the search space to ensure the diversity of the initial swarm. Secondly, we adopt GEDM to assist PSO in updating the population information. At the early stage of the optimization process, the GEDM is selected with a high probability to estimate a better evolution direction using promising particles. The purpose is to accelerate the convergence speed. At the later stage of the optimization process, PSO is used with a high probability to make full use of the great local search ability of PSO. The purpose is to improve the convergence accuracy. Thirdly, when the search is stagnant, the Lévy flight strategy is adopted to generate stagnation disturbance, which can increase the diversity of the population to help the search escape from the local optimum. Finally, the greedy strategy is carried out to select the promising particles from the parents and the newly generated candidates according to their fitness.

The rest of this paper is organized as follows. Section II reviews the related work of data clustering based on PSO. Section III includes the problem definition of clustering. In Section IV, the proposed clustering algorithm GLPSOK is described. Section V describes the experimental results and analysis. In Section VI, we conclude this work and indicate directions for future research.

II. RELATED WORK

In the past few decades, researchers made significant progress in PSO-based data clustering. Depending on the research method, the research carried out in this respect can be divided into two categories. The first category is the hybrid clustering approaches, which combine PSO with traditional clustering algorithms. The second category is the effective PSO variants for data clustering.

The goal of the hybrid clustering approaches is to take full advantage of PSO and traditional clustering algorithms to get better clustering results. As early as 2003, Merwe and Engelbrecht [21] proposed the first PSO-based clustering algorithm, denoted as HPSOK-means in our paper, where the initial swarm was fed with the clusters generated by K-means. In 2008, Ahmadyfard and Modares [23] proposed a clustering algorithm based on PSO and K-means. In this paper, PSO was adopted for global search at the initial stages of the optimization process and K-means was used to achieve faster convergence to an optimum solution when around global optimum. A hybrid algorithm based on PSO, ACO and k-means for cluster analysis was proposed in [24] to solve nonlinear partitioning problems in data clustering. In order to process the large-scale gene expression data generated by microarray experiments, Deepthi and Thampi [25] proposed a clustering algorithm, which adopted PSO to search for the best subset and then used k-means as a wrapper algorithm to evaluate the obtained subsets. In 2019, Xu *et al.* [26] proposed an accelerated two-stage particle swarm optimization (ATPSO) for clustering not-well-separated data. In ATPSO, K-means is utilized to accelerate particles convergence during the population initialization. Recently, Liu *et al.* [27] proposed an effective algorithm based on K-means and randomly occurring distributed delayed PSO (RODDPSO) to group patients from emergency center. To avoid falling into a local optimum, this algorithm introduced randomly occurring time-delays to the velocity updating model using a distributed form. Yang *et al*. [28] proposed a hybrid clustering algorithm (PSOKHM) combining K-Harmonic Means (KHM) with PSO and the experimental results indicated the superiority of the PSOKHM algorithm. A hybrid clustering algorithm based on KHM, PSO, and GA was proposed in [29], where PSO was combined with another evolutionary

algorithm such as GA to enhance the standard PSO. Bouyer and Hatamlou [1] proposed a new clustering algorithm which combined KHM with an Improved Cuckoo Search (ICS) and PSO where ICS was used to help PSO escape from the local optimum. An improved FCM clustering algorithm based on PSO was proposed in [30], in which, firstly, each object in the data set was distributed on the basis of distance to meet the constraints of FCM. And then, PSO was adopted to search for a global optimal solution. Zhao *et al.* [4] proposed an alternate PSO-based adaptive interval type-2 intuitionistic Fuzzy C-means clustering algorithm (A-PSO-IT2IFCM) and used it to solve a problem of image segmentation. With all, most hybrid clustering algorithms only combine PSO with traditional clustering algorithms in a simple way. Because PSO also has the possibility of converging to a local optimum, this kind of approaches cannot completely solve the problems that the traditional clustering algorithms possess, like being easily trapped in local optima and having low accuracy.

In recent years, researchers have proposed various PSO variants for data clustering. In 2008, Alam *et al.* [31] proposed a new algorithm called Evolutionary Particle Swarm Optimization (EPSO), which was based on the evolution of swarm generations for clustering. The swarm attempted to dynamically update itself to optimal positions during each iteration. In 2010, Szabo *et al.* [32] proposed a Modified Particle Swarm Clustering (mPSC) algorithm on the basis of Particle Swarm Clustering (PSC). This work modified the metaphor of natural social order to decrease the input parameters of the system and particles velocity's memory. The purpose of this paper is to reduce the computational complexity of PSC, but Szabo has acknowledged that this improvement is not significant. A version of PSC-based algorithm was proposed by Yuwono *et al.* [33], in which the rapid centroid estimation (RCE) was adopted to simplify the update rules of PSC for clustering. In this algorithm, which is denoted as PSC-RCE in this paper, each particle represents the centroid of a cluster, and all particles form a solution of the problem. By introducing New Substitution Strategy, Particle Reset, Swarm Strategy, and White Noise Update Scheme, this algorithm improves the efficiency of the clustering process and could better approximate the global optimal solution. In general, the PSO variants for data clustering can reduce the possibility of getting trapped in local optima. However, this kind of algorithms also has some problems. For example, in PSO, the update of the population mainly depends on the current global optimal solution and the individual historical optimal solution, which often leads to low convergence efficiency at the early stage of the search and a significant decrease in population diversity at the later stage of the search.

In summary, although clustering algorithms based on PSO, have emerged endlessly, none of them can simultaneously solve all the three problems of premature convergence, low convergence efficiency and low clustering accuracy in a satisfactory manner.

III. PROBLEM DEFINITION

Clustering is an unsupervised learning method that aims to organize each data point in the data set into the corresponding cluster. It is based on the principle of maximizing the intra-cluster similarity and minimizing the inter-cluster similarity. For a given data set $\chi = \{\boldsymbol{x}_i\}_{i=1}^{N}, \boldsymbol{x}_i \epsilon \mathcal{R}^D$, where N is the number of data points and D is the dimension of the data to be clustered, K-means clustering partitions it into K clusters $\{C_j\}_{j=1}^{K}$ by minimizing the sum of the intra-cluster variances defined as Eq.(1). K-means seeks better solutions by alternately optimizing cluster centroids and the assignment of data points to clusters.

$$\varepsilon_{sum} = \sum_{i=1}^{K} \sum_{\boldsymbol{x} \in C_i} \left\| \boldsymbol{x} - \boldsymbol{\mu}_i \right\|^2 \tag{1}$$

$$\boldsymbol{\mu}_i = \frac{1}{|C_i|} \sum_{\boldsymbol{x} \in C_i} \boldsymbol{x} \tag{2}$$

where $\boldsymbol{\mu}_i$ is the centroid of the cluster $C_i, \boldsymbol{x}$ are objects that belong to the cluster C_i.

In the context of clustering, the proposed GLPSOK algorithm is used to optimize the cluster centroids. A single particle represents K cluster centroid vectors. That is, each particle $\boldsymbol{X}_t$ ($t = 1, 2, \ldots S$, where S is the population size) is constructed as follows:

$$\boldsymbol{X}_t = [\boldsymbol{C}_1^t, \boldsymbol{C}_2^t, \ldots, \boldsymbol{C}_K^t] \tag{3}$$

$$\boldsymbol{C}_k^t = [M_{k,1}^t, M_{k,2}^t, \ldots, M_{k,D}^t] \tag{4}$$

where $\boldsymbol{C}_k^t$ refers to the k-th cluster centroid vector of the particle $\boldsymbol{X}_t$, $M_{k,d}^t$ is the d-th dimensional position of the centroid $\boldsymbol{C}_k^t$, K is the number of clusters, D is the dimension of the data to be clustered. For example, considering a search space with $K = 3$ and $D = 3$, the particle [2.4, 2.7, 0.3, 1.8, 0.2, 0.8, 2.8, 2.9, 0.4] encodes the centroids [2.4, 2.7, 0.3], [1.8, 0.2, 0.8] and [2.8, 2.9, 0.4]. For a more general example, Fig.1 shows the encoding of a solution that is formed by K centroids, where $D = 3$. Therefore, a swarm represents a number of candidate clustering results for the current data vectors.

A variety of validity metrics have been proposed as the objective functions of clustering algorithms, such as sum of Euclid Distance (SED) [34], Mean Squared Error (MSE) [1], and J_m-index[35]. In this paper, SED has been adopted as the fitness function of GLPSOK following the suggestion of Kaufman [34]. For each particle $\boldsymbol{X}_t$, its fitness function SED is defined as follows:

$$SED = \sum_{i=1}^{K} \sum_{\boldsymbol{x} \in C_i} \left\| \boldsymbol{x} - \boldsymbol{\mu}_i \right\| \tag{5}$$

where $\boldsymbol{\mu}_i$ refers to the centroid vector of cluster C_i within particle $\boldsymbol{X}_t, \boldsymbol{x}$ are objects belonging to the cluster C_i.

IV. PROPOSED GLPSOK ALGORITHM

A. POPULATION INITIALIZATION

For most PSO-based clustering algorithms, two initialization methods [36] are often adopted. The first method randomly locates all initial particles throughout the search space. The

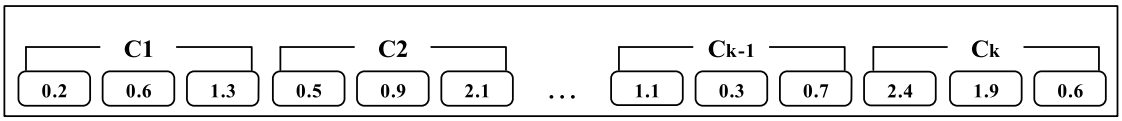


FIGURE 1. Particle representation.

other one randomly chooses K objects from the data set as the cluster centroids to form the initial particles.

In this paper, a hybrid approach is used. The specific description is as follows: 5% of the particles in the initial swarm are initialized with the cluster centroids generated by K-means. The remaining 95% of the particles are initialized using the second method described above, i.e., K objects are randomly chosen from the data set as the centroids to form the initial particles.

We use K-means' clustering results to initialize a few particles in the initial swarm for two reasons: (1) Clustering results of K-means may not be ideal, but they are much better than the randomly obtained initialization results in most cases. Therefore, this method can guarantee there are some relatively good particles in the initial swarm; (2) K-means is one of the most classic, simplest, and most efficient clustering algorithms. Initializing particles with the clustering results of K-means can significantly improve the performance of clustering at the cost of a little more computational time.

B. UPDATE RULES OF THE POPULATION

In basic PSO, the update of the population mainly depends on the current global optimal solution and the individual historical optimal solution, which will lead to low convergence efficiency at the early stage of the search and a significant decrease in population diversity at the later stage of the search. GEDM, whose core is the weighted covariance matrix, can make full use of promising particles to estimate a better evolution direction and has the ability to avoid local optimum. Therefore, we adopt GEDM to improve the performance of the proposed clustering algorithm.

1) PARTICLE SWARM OPTIMIZATION

PSO was first proposed by Kennedy and Eberhart in 1995 [12]. This algorithm imitates the process of bird foraging and guides optimization search by swarm intelligence generated by cooperation and competition among particles in a swarm.

In PSO, the swarm of particles is considered as a set of potential solutions, and the fly process of the particles can be regarded as a search process [37]. Each particle i is associated with two vectors, i.e., the position vector $\boldsymbol{X}_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,n}]$ and the velocity vector $\boldsymbol{V}_i = [v_{i,1}, v_{i,2}, \ldots v_{i,n}]$. Particles search for new solutions by constantly adjusting their positions $x_{i,d}$. For each particle, it can remember its individual historical optimal position that it has searched for, as $\boldsymbol{pbest}_i$, and the current global optimal position found by the entire particle swarm, as $\boldsymbol{gbest}$. When the two optimal positions are found, each particle updates its velocity and position according to Eq.(6) and Eq.(7), respectively.

$$v_{i,d}(t+1) = \omega \cdot v_{i,d}(t) + \mathrm{c}_1 \cdot rand_1 \cdot (\boldsymbol{pbest}_{i,d} - x_{i,d}(t)) + \mathrm{c}_2 \cdot rand_2 \cdot (\boldsymbol{gbest}_d - x_{i,d}(t)) \quad (6)$$

$$x_{i,d}(t+1) = x_{i,d}(t) + v_{i,d}(t+1) \quad (7)$$

where t represents the t-th iteration, d represents the d-th dimension of the particle, ω is the inertia weight, c_1 and c_2 are the acceleration constants, $rand_1$ and $rand_2$ are random numbers randomly distributed in the interval [0, 1].

It has been proved that the performance of PSO can be improved if the inertia weight ω decreases linearly [38]. The linearly decreased inertia weight is given as follows:

$$\omega = \omega_{\max} - (\omega_{\max} - \omega_{\min}) \cdot \frac{t}{t_{\max}} \quad (8)$$

where ω_{max} and ω_{min} are the maximal and minimal weights, respectively; t is the number of the current iteration, and t_{max} is the number of the maximum iteration.

2) GAUSSIAN ESTIMATION OF DISTRIBUTION METHOD

The estimation of distribution algorithm (EDA) can estimate the evolution direction of the promising population using probabilistic model learning and sampling. Some studies have shown that EDA has promising performance when dealing with complex optimization problems [39] [40]. GEDM is the core component of EDA. Therefore, it has been introduced into PSO to estimate the better evolution direction for the purpose of improving the convergence speed and enhancing the local optimal avoidance ability of PSO. The model of GEDM based on the weighted covariance matrix is as follows.

$$\omega_i = \frac{\ln(m+1) - \ln(i)}{\sum_{i=1}^{m} (\ln(m+1) - \ln(i))} \quad (9)$$

$$\boldsymbol{X}(t)_{mean} = \sum_{i=1}^{m} \omega_i \cdot \boldsymbol{X}_i(t) \quad (10)$$

$$\boldsymbol{Cov}(t) = \frac{1}{m-1} \cdot \sum_{i=1}^{m} (\boldsymbol{X}_i(t) - \boldsymbol{X}(t)_{mean})(\boldsymbol{X}_i(t) - \boldsymbol{X}(t)_{mean})^T \quad (11)$$

$$\boldsymbol{X}_i(t+1) = Gaussian(\boldsymbol{X}(t)_{mean}, \boldsymbol{Cov}(t)) + rand \cdot (\boldsymbol{X}(t)_{mean} - \boldsymbol{X}_i(t)) \quad (12)$$

In Eq.(9), m is the number of solutions that are selected as the promising solutions to estimate the evolutionary direction. The solution, which has a high rank, would have a great weight when calculating the weighted mean using Eq.(10). The set of $\{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_m\}$ represents the promising solutions with fitness values ranked from high to low in Eq.(10). In Eq.(11), $\mathbf{Cov}(t)$ is the weighted covariance matrix of the

promising solutions. When using GEDM, the population update its position using Eq.(12). The first addend of Eq. (12) corresponds to Gaussian random walk.

In order to integrate the GEDM and PSO together effectively, the GEDM and PSO are selected with different probabilities at different stages of the optimization process. At the early stage of the optimization process, the GEDM is selected with a high probability to estimate the better evolution direction using promising particles, for the purpose of accelerating the convergence speed. At the later stage of the optimization process, PSO is used with a high probability to make full use of its great local search ability, for the purpose of improving the convergence accuracy of GLPSOK. The model of the probability is as follows:

$$\gamma = \gamma_{\max} - (\gamma_{\max} - \gamma_{\min}) \cdot \frac{t}{t_{\max}} \tag{13}$$

where γ_{max} and γ_{min} are the maximal and minimal probabilities to select GEDM, respectively.

3) STAGNATION DISTURBANCE STRATEGY

The Lévy process is a stochastic process of continuous time, which was first proposed by the French scientist Paul Lévy. Researchers have found that the behavior of many animals in nature is consistent with the characteristics of the Lévy process and have proposed the Lévy flight theory [41] based on the Lévy process. Lévy flight is an optimal exploration behavior to search for randomly distributed objects. It is characterized by long-term random walks with small steps and occasionally jumping with a large step, which is similar to the global search and local search features in the intelligent optimization algorithm. Thus, Lévy flight is widely used by researchers in various optimization algorithms to generate random step sizes. In this paper, once the search is stagnant, the Lévy flight has been adopted to generate disturbance for the purpose of escaping from the local optimum. The random walk step of Lévy flight follows a heavy tail probability distribution, called Lévy distribution, whose exponential form is as follows:

$$L(s) \sim |s|^{-1-\beta}, \quad \beta \in (0, 2) \tag{14}$$

where s is the random step size, β is the exponential parameter that determines the shape of the Lévy distribution. The value of β is inversely proportional to the generated random step size. Because the exponential form of Lévy distribution is difficult to implement using MATLAB programming language, the method of generating the Lévy flight random search path proposed by Mantegna [42] is used to generate Lévy flight random step size in this paper. The model of Lévy flight is as follows:

$$\begin{cases} \sigma_u = \left(\dfrac{\Gamma(1+\beta) \cdot \sin(\pi \cdot \frac{\beta}{2})}{\Gamma((1+\beta)/2) \cdot \beta \cdot 2^{(\beta-1)/2}} \right)^{1/\beta} \\ \sigma_v = 1 \end{cases} \tag{15}$$

$$u \sim N(0, \sigma_u^2) \tag{16}$$

$$v \sim N(0, \sigma_v^2) \tag{17}$$

$$s = \frac{u}{|v|^{1/\beta}} \tag{18}$$

where s is the random walk step size, and $\Gamma(x)$ is the gamma function.

In the search process, the average fitness value of the promising solutions is used to determine whether the search is stagnant. If the average fitness value does not change in three consecutive iterations, the search would be regarded as stagnant. Once the search process is stagnant, in order to escape from the local optimum and overcome the premature convergence of GLPSOK, the Lévy flight strategy is adopted to generate disturbance to update the population information. In addition, the stagnation disturbance strategy can also increase the diversity of the population. The model of stagnation disturbance strategy based on Lévy flight is as follows:

$$\boldsymbol{X}_i(t+1) = \boldsymbol{X}_i(t) + randn \cdot Levy(\boldsymbol{X}_i(t)) \tag{19}$$

$$Levy(\boldsymbol{X}_i(t)) = \alpha \cdot s \cdot (\boldsymbol{gbest} - \boldsymbol{X}_i(t)) \tag{20}$$

where *randn* is a random number following a normal distribution, and $\alpha \in [-1, 1]$ is a scale factor.

C. BOUNDARY HANDLING

Some works [43] have shown that boundary handling has a significant impact on the performance of PSO algorithms, especially when solving complex problems. In recent years, many boundary handling schemas [36] have been proposed, including those based on either periodic, absorbing, invisible, damping, reflecting, random or zoom. Among the boundary handling schemas above, the most fundamental and widely used are random and absorbing schemas. A brief description of these two fundamental schemas [43] is addressed below.

1) Random Schema: In general, this schema is adopted as the default setting in PSO programs. If a particle flies outside any dimension j of the search space, a random value drawn from a uniform distribution between the lower and upper boundaries of the dimension j would be assigned as the corresponding component for the particle, as follows.

$$\boldsymbol{X}_t = [x_{t,1}, \ldots, x_{t,j-1}, U(b_l; b_u), x_{t,j+1}, \ldots, x_{t,n}] \tag{21}$$

where $t = 1, 2, \ldots, S$; S is the population size; b_l is the lower boundary of the *j-th* dimension of the *t-th* particle;b_u is the upper boundary of the *j-th* dimension of the *t-th* particle; $U(a; b)$ is a random value drawn from a uniform distribution between a and b.

2) Absorbing Schema: In this schema, when a particle flies outside any dimension j of the search space, the component corresponding to the *j-th* dimension of the particle is assigned the boundary of the dimension *j*.

$$\boldsymbol{X}_t = [x_{t,1}, \ldots, x_{t,j-1}, b_l \ (or\ b_u), x_{t,j+1}, \ldots, x_{t,n}] \tag{22}$$

In this paper, the random schema is adopted to handle the boundary of position to prevent particles from flying out of the search space. And the absorbing schema is adopted to

handle the boundary of velocity to prevent particles from moving too fast.

Finally, the greedy strategy is adopted to select the promising particles from the parents and the newly generated candidates according to the fitness. This step can guarantee the global convergence efficiency of the proposed GLPSOK algorithm. This mechanism can fully retain the dominant particles, which can improve the convergence speed of the algorithm and obtain better clustering results.

The pseudo code of the proposed GLPSOK is described as Algorithm 1.

Algorithm 1 The Procedure of GLPSOK

1. Initialize the population (see Sect.IV-A).
2. For each particle, assign each data point to its nearest cluster $C_j(j = 1, 2, \ldots K)$;
3. Calculate the fitness of each particle using Eq.(5); Update the values for both ***pbest*** and ***gbest***;
4. If $t < t_{max}$
   Execute step 5;
   else
   Output the ***gbest*** one X*;
   End if;
5. Calculate $\boldsymbol{X}(t)mean$ and $\boldsymbol{Cov}(t)$;
6. If the search stagnates
   Population is updated using Eq.(19);
   Else
   If rand $> \gamma$
   Population is updated using PSO;
   Else
   Population is updated using GEDM;
   End if
   End if
7. Boundary control;
8. For each newly generated candidate particle, assign each data point to its nearest cluster $C_j(j = 1, 2, \ldots K)$;
9. Calculate the fitness of each generated candidates;
10. Greedy strategy is adopted to select the promising particles;
11. Update the values for both ***pbest*** and ***gbest***;
12. $t = t + 1$; Execute step 4;

D. COMPUTATION COMPLEXITY ANALYSIS

The computation complexity of GLPSOK is described below. At the initialization stage (from Step 1 to Step 3), the computation complexity is O($N{\cdot}K{\cdot}D{\cdot}NP$), where N is the number of data points to be clustered, K is the number of clusters, D is the dimension of data points, NP is the number of particles. The computation complexity of updating the population (Step 5 and Step 6) is O($K^2{\cdot}D^2{\cdot}NP$), where the computation complexity of computing $\boldsymbol{X}(\boldsymbol{t})_{mean}$ and $\boldsymbol{Cov}(\boldsymbol{t})$ is O($K{\cdot}D{\cdot}NP$) and O($K^2{\cdot}D^2{\cdot}NP$), respectively. The computation complexity of stage from Step 7 to Step 11 is O($N{\cdot}K{\cdot}D{\cdot}NP$). The computation complexity of stage from Step 4 to Step 12 is O($K{\cdot}D{\cdot}NP{\cdot}(K{\cdot}D{+}N){\cdot}t_{max}$), where t_{max} is the maximum number of iterations of GLPSOK. Let $MaxFEs$ be the maximum number of the fitness evaluations of PSO-based Clustering algorithm. Then, in GLPSOK, $MaxFEs = NP{\cdot}t_{max}$. Therefore, the overall computation complexity of GLPSOK can be expressed as $O(K{\cdot}D{\cdot}(K{\cdot}D{+}N){\cdot}MaxFEs)$. Table 1 summarizes the computation complexity of GLPSOK and eight related algorithms, which are introduced in detail in Section V as comparison algorithms. In non-PSO-based clustering algorithms, such as K-means, MinMaxK-means, and K-Multiple-Means (KMM), $MaxFEs$ represents the maximum number of calculations of the objective function of algorithms. In PSC-RCE, n_m is the number of groups in the swarm. In K-Multiple-Means, m is the number of sub-clusters and t_1 is the number of iterations of the sub-alternating system. In the actual program running, the running time of PSO-based clustering algorithms is much longer than that of non-PSO-based clustering algorithms. The number of fitness evaluations is close or equal to $MaxFEs$ for PSO-based clustering algorithms and the number of calculations of the objective function is usually much less than $MaxFEs$ for non-PSO-based clustering algorithms when the algorithm terminates after satisfying the algorithm termination condition.

TABLE 1. The computation complexity of GLPSOK and eight related algorithms.

Algorithm	Computation Complexity
K-means [10][44]	O $(N \cdot K \cdot D \cdot MaxFEs)$
BIRCH [5]	O (N)
HPSOK-means [21]	O $(N \cdot K \cdot D \cdot MaxFEs)$
MinMaxK-means[45]	O $(N \cdot K \cdot D \cdot MaxFEs)$
PSC-RCE [33]	O $(N \cdot K \cdot \mathrm{D} \cdot n_m \cdot MaxFEs)$
PSOLFK	O $(N \cdot K \cdot D \cdot MaxFEs)$
PSOSCALFK	O $(N \cdot K \cdot D \cdot MaxFEs)$
KMM [46]	O $(N \cdot ((m \cdot D + m^2 + m \cdot K) \cdot t_1 + mD) \cdot MaxFEs)$
GLPSOK	O $(K \cdot D \cdot (K \cdot D + N) \cdot MaxFEs)$

V. EXPERIMENTAL RESULTS AND ANALYSIS

The proposed GLPSOK is a hybrid clustering algorithm based on PSO and K-means. In order to evaluate its effectiveness, we tested it on six synthetic data sets. For the purpose of proving its superiority, we compared it with eight classic or state-of-the-art algorithms on five real-word data sets. The experiments were conducted using the Matlab R2016a platform on a computer with 1.99 GHz Inter(R) Core (TM) i7-8565U CPU, 16 GB memory and Windows 10 operating system.

A. EXPERIMENTAL DATA SETS

Six synthetic data sets and five real-world data sets were used to evaluate the proposed algorithm. The synthetic data sets ($S_1 - S_6$), which are presented in [47], [48], are generated using the Gaussian model and shown in Fig. 2a-7a. These data sets were selected for the following reasons: S_1

shows the asymmetric case where the clusters have different shapes and different numbers of data points. $S_2 - S_4$ shows the approximately symmetric case where the clusters take the similar shape and have the same numbers of data points. The difference between S_2, S_3, and S_4 is that the distance between cluster boundaries decreases in order, which can help in evaluating the performance of the proposed algorithm on data sets with unclear boundaries. S_5 and S_6 have been selected to evaluate the performance of the proposed algorithm on multi-dimensional data sets and data sets with multiple clusters, respectively. The real-world data sets (Iris, Wine, Glass, WDBC, and CMC) are obtained from the famous UCI Machine Learning Repository [49]. These data sets have been selected because they come from various domains and have been widely used in the field of machine learning [1] [50]. The information of the synthetic data sets and real-world data sets are described in Table 2 and Table 3, respectively. A detailed description of all data sets is shown below.

TABLE 2. The main properties of the selected synthetic data sets.

Order	Dataset	Size	Number of Features	Number of Clusters
1	S_1	900	2	3
2	S_2	1200	2	4
3	S_3	1200	2	4
4	S_4	1200	2	4
5	S_5	300	3	4
6	S_6	500	2	10

TABLE 3. The main properties of the selected real-world data sets.

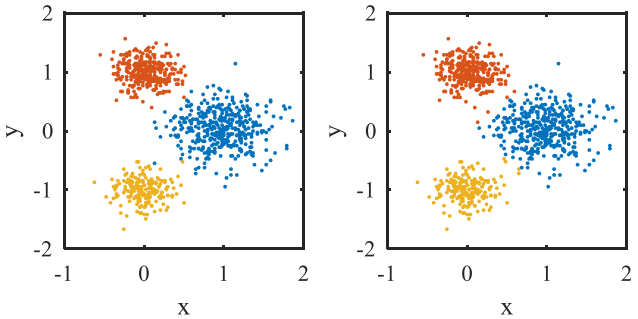
Order	Dataset	Size	Number of Features	Number of Clusters
1	Iris	150	4	3
2	Wine	178	13	3
3	Glass	214	9	6
4	WDBC	569	30	2
5	CMC	1473	9	3

1) SYNTHETIC DATA SET1 (n = 900, d = 2, k = 3)

This data set consists of three classes, those ones having 400,300, and 200 data points, respectively. These data points are drawn from three different bivariate Gaussian distributions with the following parameters:

$$\mu_1 = (1, 0),\ \mu_2 = (0, 1),\ \mu_3 = (0, -1),$$
$$\sum\nolimits_1 = \begin{bmatrix} 0.09 & 0 \\ 0 & 0.09 \end{bmatrix},\ \sum\nolimits_2 = \sum\nolimits_3 = \begin{bmatrix} 0.04 & 0 \\ 0 & 0.04 \end{bmatrix}$$

where μ_i is the mean vector, and Σ_i is the covariance matrix. The overall data distribution is shown in Fig.2a. Different colors in the illustration represent different classes (ground truths) of the data points.



(a) Classes distribution of S_1 (b) Clustering results of S_1.

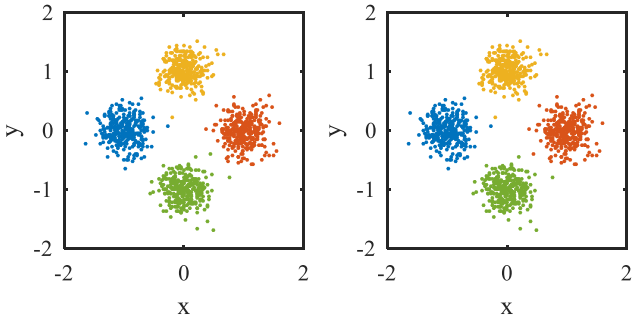
FIGURE 2. The illustration of S_1.

2) SYNTHETIC DATA SET2 (n = 1200, d = 2, k = 4)

This data set consists of four classes, each containing 300 data points. These data points are drawn from four different bivariate Gaussian distributions with the following parameters:

$$\mu_1 = (-1, 0),\ \mu_2 = (1, 0),\ \mu_3 = (0, 1),\ \mu_4 = (0, -1),$$
$$\sum\nolimits_1 = \sum\nolimits_2 = \sum\nolimits_3 = \sum\nolimits_4 = \begin{bmatrix} 0.04 & 0 \\ 0 & 0.04 \end{bmatrix}$$

where μ_i is the mean vector, and Σ_i is the covariance matrix. The overall data distribution is shown in Fig.3a.



(a) Classes distribution of S_2 (b) Clustering results of S_2.

FIGURE 3. The illustration of S_2.

3) SYNTHETIC DATA SET3 (n = 1200, d = 2, k = 4)

This data set consists of four classes, each containing 300 data points. These data points are drawn from four different bivariate Gaussian distributions with the following parameters:

$$\mu_1 = (-1, 0),\ \mu_2 = (1, 0),\ \mu_3 = (0, 1),\ \mu_4 = (0, -1),$$
$$\sum\nolimits_1 = \sum\nolimits_2 = \sum\nolimits_3 = \sum\nolimits_4 = \begin{bmatrix} 0.09 & 0 \\ 0 & 0.09 \end{bmatrix}$$

where μ_i is the mean vector, and Σ_i is the covariance matrix. The overall data distribution is shown in Fig.4a.

4) SYNTHETIC DATA SET4 (n = 1200, d = 2, k = 4)

This data set consists of four classes, each containing 300 data points. These data points are drawn from four different bivariate Gaussian distributions with the following parameters:

$$\mu_1 = (-1, 0), \mu_2 = (1, 0), \mu_3 = (0, 1), \mu_4 = (0, -1),$$
$$\sum\nolimits_1 = \sum\nolimits_2 = \sum\nolimits_3 = \sum\nolimits_4 = \begin{bmatrix} 0.16 & 0 \\ 0 & 0.16 \end{bmatrix}$$

where μ_i is the mean vector, and Σ_i is the covariance matrix. The overall data distribution is shown in Fig.5a.

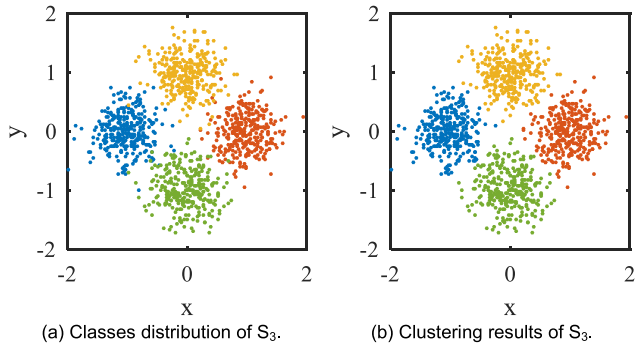


FIGURE 4. The illustration of S_3.

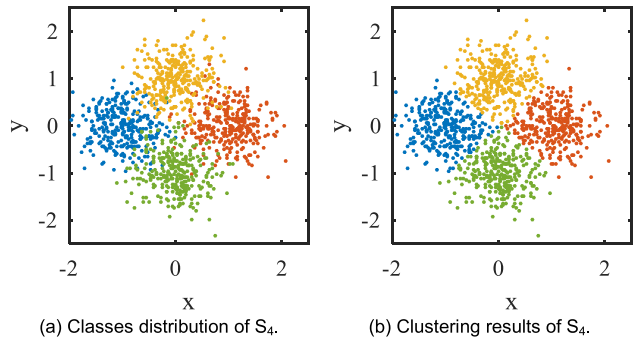


FIGURE 5. The illustration of S_4.

5) SYNTHETIC DATA SET5 (n = 300, d = 3, k = 4)

This data set consists of four classes, and the mixing ratio of the four classes is [0.1, 0.2, 0.3, 0.4]. These data points are drawn from four different tripartite Gaussian distributions with the following parameters:

$$\mu_1 = (0, 0, 0),\ \mu_2 = (3, 3, 2),$$
$$\mu_3 = (-3, 3, 1),\ \mu_4 = (0, -3, 3),$$
$$\sum_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},\ \sum_2 = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$
$$\sum_3 = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix},\ \sum_4 = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

where μ_i is the mean vector, and Σ_i is the covariance matrix. The overall data distribution is shown in Fig.6a.

6) SYNTHETIC DATA SET6 (n = 500, d = 2, k = 10)

This data set consists of ten classes, each containing 50 data points. These data points are drawn from ten different bivariate Gaussian distributions with the following parameters:

$$\mu_1 = (1, 1),\ \mu_2 = (1, 5),\ \mu_3 = (1, 9),\ \mu_4 = (5, 1),$$
$$\mu_5 = (5, 5),\ \mu_6 = (5, 9),\ \mu_7 = (9, 1),\ \mu_8 = (9, 5),$$
$$\mu_9 = (9, 9),\ \mu_{10} = (13, 5),$$
$$\sum_1 = \sum_2 = \cdots = \sum_{10} = \begin{bmatrix} 1.1 & 0 \\ 0 & 1.1 \end{bmatrix}$$

where μ_i is the mean vector, and Σ_i is the covariance matrix. The overall data distribution is shown in Fig.7a.

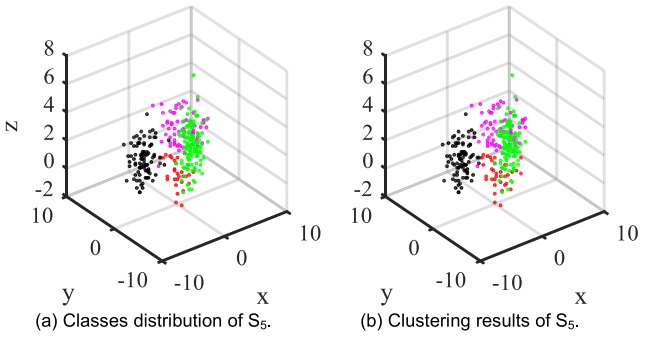


FIGURE 6. The illustration of S_5.

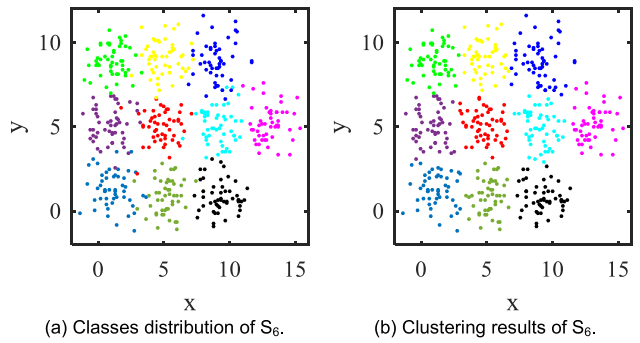


FIGURE 7. The illustration of S_6.

7) IRIS DATA SET (n = 150, d = 4, k = 3)

Iris is one of the most widely used data sets in the pattern recognition literature. This data set consists of three classes, each of which contains 50 data points. Each class refers to a type of Iris plant.

8) WINE DATA SET (n = 178, d = 13, k = 3)

M.Forina *et al.* [51] analyzed the chemical composition of three wines grown in the same region of Italy, hoping to determine the origins of the wines using chemical analysis. Wine data set was the analysis result of this experiment which determined the quantities of 13 constituents found in each of the three types of wines.

9) GLASS IDENTIFICATION DATA SET (n = 214, = 9, k = 6)

In criminology, some research has been done on the classification of glass types, because the glass left at the crime scene can be used as evidence. The Glass Identification Data Set contains a collection of attributes that are useful for categorizing glass types. The first attribute in the instances (objects), the ID number from 1 to 214, is ignored in our research, so the dimension of the data set is 9 instead of 10.

10) BREAST CANCER WISCONSIN (Diagnostic) DATA SET (n = 569, d = 30, k = 2)

This data set is denoted as WDBC in this paper. Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe the characteristics of the cell nuclei present in the image.

11) CONTRACEPTIVE METHOD CHOICE DATA SET (n = 1473, d = 9, k = 3)

This data set is a subset of the 1987 National Indonesia Contraceptive Prevalence Survey and is denoted as CMC in this paper. The woman investigated is either not pregnant or not sure if she is pregnant. The survey is to predict the current contraceptive approach (no use, long-term methods, or short-term methods) of a woman based on her demographic and socio-economic characteristics.

B. COMPARISION ALGORITHMS

Our proposed algorithm is a hybrid clustering one based on K-means and PSO. Therefore, we have selected several classic or state-of-the-art algorithms related to K-means or PSO to carry out a convenient comparative analysis. In addition, we also chose a classic hierarchical clustering algorithm, named BIRCH. Below further details of the comparison algorithms are briefly explained.

1) K-means is one of the most popular clustering algorithms and the basis of the proposed GLPSOK algorithm. This algorithm partitions the data set into K clusters by minimizing the sum of the intra-cluster variances.
2) MinMaxK-means [45] is an effective K-means variant. This algorithm assigns weights to the corresponding clusters according to their variance and optimizes a weighted version of the objective function of K-means. Weights are learned together with the cluster assignments through an iterative procedure.
3) K-Multiple-Means (KMM) [46] is the state-of-the-art multi-prototype clustering algorithm based on K-means, which organizes the data set with multiple sub-cluster centroids into specified k clusters.
4) HPSOK-means [21] is the first work to apply PSO to data clustering. This algorithm uses PSO to find the centroids of specified k clusters, where the clusters generated by K-means are used to initialize the initial swarm of PSO.
5) PSC-RCE [33] is a famous variant of particle swarm clustering (PSC). This algorithm simplifies the update rules of PSC, and significantly reduces computational complexity by improving the efficiency of the particle trajectories.
6) PSOLF [52] is an Enhanced Particle Swarm Optimization with Lévy Flight for global optimization. In order to prove that our proposed GLPSOK clustering can outperform other partitioning based clustering methods that are assisted by present state-of-art PSO variants, we combine PSOLF and K-means in the same way used in this paper. The combination of PSOLF and K-means is denoted as PSOLFK in this paper.
7) PSOSCALF [53] is a state-of-the-art PSO variant. It is based on sine cosine algorithm and Lévy flight for solving optimization problems. Similarly, we combine PSOSCALF and K-means in the same way used in this paper for the purpose of proving that our proposed GLPSOK clustering can outperform other partitioning based clustering methods that are based on present state-of-art PSO variants. The combination of PSOSCALF and K-means is denoted as PSOSCALFK in this paper.
8) BIRCH [5] is a classic hierarchical clustering algorithm which can typically find a good clustering result only with a single data scan and improve the quality of the clustering result further with a few additional scans. The reason that we select BIRCH as the comparison algorithm is to further prove that GLPSOK can achieve superior performance over different types of clustering algorithms.

C. PARAMETER SETTINGS

To ensure the fairness of the experiments, for all PSO-based algorithms, the maximum number of fitness evaluations (MaxFEs) was set to 30000. As for other algorithms such as K-means, BIRCH, MinMaxK-means, and KMM, the maximum number of calculations for their objective functions was also set to 30000. K-means has no additional parameters. Other parameters of the other seven comparison algorithms were set as indicated in the corresponding original papers or as default values. The parameter settings of the eight algorithms are shown in Table 4. In order to avoid experimental deviations, each algorithm was run 30 times on each data set independently.

TABLE 4. Parameter setting.

Algorithm	Parameter Settings
BIRCH	*branching_factor* = 50, *threshold* = 0.5
HPSOK-means	$c_1 = c_2 = 1.49$, $\omega = 0.72$
MinMaxK-means	p_{init} =0, p_{max} = 0.5, p_{step} = 0.01, β = 0.3
PSC-RCE	*swarm_num* = 5, *max_stagnation* = 200, *m* = 1.4
PSOLFK	c_1=1.2, c_2=1.8, ω=01.+0.8·(1-*iteration*/*Maxiter*)
PSOSCALFK	c_{1min}=c_{2min}=0.5,c_{1max} = c_{2max}=2.5 *limit*=10,β=1.5,ω_{min}=0.4,ω_{max}=0.9,*scale*=0.01
KMM	$k = 5$
GLPSOK	ω_{max}=0.9, ω_{min}=0.5, c_1=c_2=1.49445 *m*=*NP*/2, γ_{max}=0.7, γ_{min}=0.3, α=0.05, β=0.5

D. EVALUATION METRIC

In order to evaluate and compare the performance of the proposed algorithm with the other six comparison algorithms, four metrics were selected. The first metric is SED as defined in Eq.(5). The other three metrics are Normalized Mutual

Information [54], F-measure [1] and Accuracy [54]. Next, these metrics are described.

1) NORMALIZED MUTUAL INFORMATION (NMI)

Suppose C represents the set of classes obtained from the ground truth and C' represents the set of clusters obtained from the algorithm. Their mutual information measure MI(C,C$^{'}$) is defined as follows:

$$MI(C, C') = \sum_{c_i \in C, c'_j \in C'} p(c_i, c'_j) \cdot \log_2 \frac{p(c_i, c'_j)}{p(c_i) \cdot p(c'_j)} \quad (23)$$

where $p(c_i)$ and $p\left(c'_j\right)$ are the probabilities that an object randomly selected from the data set belongs to the class c_i and cluster c'_j, respectively; and $p\left(c_i, c'_j\right)$ is the probability that an object randomly selected from the data set belongs to c_i and also belongs to c'_j. In our experiments, NMI is defined as follows:

$$NMI(C, C') = \frac{MI(C, C')}{\max(H(C), H(C'))} \quad (24)$$

where $H(C)$ and $H\left(C'\right)$ are the entropy of C and C', respectively; NMI ranges from 0 to 1. NMI = 1 if the C and C$^{'}$ are identical; and NMI = 0 if C and C$^{'}$ are mutually independent.

2) F-MEASURE

Precision and recall are two metrics that have been widely used in information retrieval and statistics. It is desirable that the precision and recall are as high as possible. However, in some cases, the two metrics contradict to each other. F-measure is an approach to consider precision and recall together. In clustering, the higher the F-measure is, the better the clustering performance is. Formally, each class i (as given by the ground truth of the input data set) is regarded as the set of n_i instances desired for a query; each cluster j (obtained from the algorithm) is regarded as the set of n_j instances retrieved from a query; n_{ij} represents the number of instances of class i within cluster j. For each class i and cluster j, F-measure (F), precision (p), and recall (r) are defined by means of the following equations:

$$F = \sum_i \frac{n_i}{n} \cdot \max_j \{F(i, j)\} \quad (25)$$

$$F(i, j) = \frac{(b^2 + 1) \cdot p(i, j) \cdot r(i, j)}{b^2 \cdot p(i, j) + r(i, j)} \quad (26)$$

$$p(i, j) = \frac{n_{ij}}{n_j} \quad (27)$$

$$r(i, j) = \frac{n_{ij}}{n_i} \quad (28)$$

In Eq.(26), b is assigned 1 to keep equal weights for precision and recall.

3) ACCURACY (AC)

Given an object x_i, let s_i and r_i be the ground truth of x_i and the cluster label obtained by algorithms, respectively. The definition of AC is as follows:

$$AC = \frac{\sum_{i=1}^{n} \delta(s_i, map(r_i))}{n} \quad (29)$$

where n is the number of total objects; $\delta(x, y)$ is the delta function that equals 1 if $x = y$, 0 otherwise; and $map(r_i)$ is the permutation mapping function which maps each cluster label r_i to the equivalent ground truth.

E. RESULTS ON SYNTHETIC DATA SETS

In this section, the experimental results of applying the proposed algorithm GLPSOK on six synthetic data sets are described.

TABLE 5. Results of GLPSOK on synthetic data sets.

Data Set	F-measure	AC
	Mean (SD)	Mean (SD)
S_1	9.9222E-01(3.3876E-16)	9.9222E-01(4.5168E-16)
S_2	9.9917E-01(6.7752E-16)	9.9917E-01(5.6460E-16)
S_3	9.8417E-01(6.7752E-16)	9.8417E-01(1.1292E-16)
S_4	9.2582E-01(4.5168E-16)	9.2583E-01(0.0000E+00)
S_5	9.5228E-01(1.1675E-02)	9.5089E-01(1.3389E-02)
S_6	9.3625E-01(1.3597E-16)	9.3600E-01(0.0000E+00)

Table 5 shows the overall experimental results (including the mean and standard deviation (SD) of F-measure and AC) of the proposed algorithm on six synthetic data sets. The proposed algorithm was run independently 30 times on each data set. From the table we can see that the proposed algorithm achieves good clustering results on all synthetic data sets, where the F-measure and AC on all data sets are greater than 0.92, on data set 3 are greater than 0.98, and on data set 1 and data set 2 are even greater than 0.99.

In addition, Fig.2b, Fig.3b, Fig.4b, Fig.5b, Fig.6b, and Fig.7b illustrate the clustering results of the proposed algorithm on six synthetic data sets, respectively. Each color in each figure corresponds to a cluster. Since we only care about which data points belong to the same cluster and do not care what color is used to represent a cluster, we have unified the color representation of the clustering results and ground truth of each data set to more intuitively observe the clustering performance. From the figures, we can see that the proposed algorithm can achieve good clustering results.

F. RESULTS ON REAL-WORD DATA SETS

In this section, we test the performance of GLPSOK on five real-world data sets from the UCI Machine Learning Repository and compare it to eight comparison algorithms in terms of four evaluation metrics, namely, SED, NMI, F-measure, and AC. For each data set, we ran each algorithm 30 times independently. Tables 6-10 show the statistical results (Mean and standard deviation (SD)) obtained by all algorithms on data set Iris, Wine, Glass, WDBC, CMC, respectively. The best results are shown in bold.

Table 6 shows the experimental results on data set Iris. From this table we can see that GLPSOK outperforms the

TABLE 6. Statistical results obtained by nine algorithms on data set Iris in 30 independent runs.

Algorithm	SED	NMI	F-measure	AC
	Mean(SD)	Mean(SD)	Mean(SD)	Mean(SD)
K-means	1.0344E+02(1.1251E+01)	6.9425E-01(9.1285E-02)	8.4048E-01(8.8572E-02)	8.1044E-01(1.4681E-01)
BIRCH	1.0307E+02(1.4454E-14)	6.7471E-01(3.3876E-16)	7.9980E-01(0.0000E+00)	8.1333E-01(5.6464E-16)
HPSOK-means	1.2568E+02(6.0593E+00)	5.7081E-01(3.6325E-02)	7.4055E-01(4.8701E-02)	6.1267E-01(8.3659E-02)
MinMax	9.7523E+01(5.7815E-14)	7.1667E-01(1.1292E-16)	8.8545E-01(5.6460E-16)	8.8667E-01(0.0000E+00)
PSC-RCE	9.7397E+01(2.3303E-01)	7.4634E-01(6.8331E-03)	8.9013E-01(2.9856E-03)	8.9156E-01(2.9985E-03)
PSOLFK	9.7281E+01(9.3823E-02)	7.5149E-01(1.1292E-16)	8.9177E-01(5.6460E-16)	8.9333E-01(3.3876E-16)
PSOSCALFK	9.6715E+01(8.3254E-02)	7.5467E-01(7.2652E-03)	8.9614E-01(3.2904E-03)	8.9733E-01(3.3218E-03)
KMM	1.0284E+02(6.4995E+00)	7.2471E-01(7.3862E-02)	8.4411E-01(6.5592E-02)	8.3867E-01(7.6542E-02)
GLPSOK	**9.6655E+01(3.7413E-14)**	**7.6036E-01(1.1292E-16)**	**8.9878E-01(5.6460E-16)**	**9.0000E-01(4.5168E-16)**

TABLE 7. Statistical results obtained by nine algorithms on data set Wine in 30 independent runs.

Algorithm	SED	NMI	F-measure	AC
	Mean(SD)	Mean(SD)	Mean(SD)	Mean(SD)
K-means	1.6981E+04 (7.8704E+02)	4.2189E-01(1.2586E-02)	6.9751E-01 (3.2293E-02)	6.7378E-01(5.3193E-02)
BIRCH	1.6605E+04 (7.4003E-12)	4.1586E-01(5.6460E-17)	7.0688E-01 (3.3876E-16)	6.9663E-01(2.2584E-16)
HPSOK-means	1.6348E+04 (4.8800E+01)	**4.2992E-01 (2.6034E-03)**	7.2673E-01 (2.2414E-03)	7.1592E-01(2.8315E-03)
MinMax	1.6993E+04 (7.0034E+01)	4.2127E-01 (1.3709E-05)	6.8356E-01 (4.9513E-04)	6.6854E-01 (0.0000E+00)
PSC-RCE	1.6555E+04 (9.9790E+01)	4.2837E-01(3.0705E-03)	7.1439E-01 (5.3244E-03)	7.0187E-01(5.3058E-03)
PSOLFK	1.6350E+04(3.0227E+01)	4.2810E-01(2.8448E-03)	7.1590E-01(3.2882E-03)	7.0356E-01(3.8141E-03)
PSOSCALFK	**1.6292E+04(7.4384E-01)**	4.2785E-01(5.4742E-03)	7.2595E-01(3.8250E-03)	7.1536E-01(4.2590E-03)
KMM	2.2314E+04 (3.7419E+03)	3.3383E-01 (7.0575E-02)	6.7277E-01 (5.5102E-02)	6.1348E-01(5.9407E-02)
GLPSOK	1.6295E+04 (8.9799E+00)	4.2978E-01(4.6271E-03)	**7.2729E-01(3.0932E-03)**	**7.1685E-01(3.4913E-03)**

TABLE 8. Statistical results obtained by nine algorithms on data set Glass in 30 independent runs.

Algorithm	SED	NMI	F-measure	AC
	Mean(SD)	Mean(SD)	Mean(SD)	Mean(SD)
K-means	2.2751E+02(1.4059E+01)	3.5125E-01(3.5042E-02)	5.2730E-01(2.9325E-02)	5.0903E-01(3.4425E-02)
BIRCH	2.4286E+02(1.4454E-13)	2.9828E-01(1.6938E-16)	5.5736E-01(1.1292E-16)	4.9065E-01(2.2584E-16)
HPSOK-means	3.3799E+02(3.5434E+01)	1.8789E-01(5.8665E-02)	4.2381E-01(5.7445E-02)	4.1277E-01(5.9721E-02)
MinMax	2.2215E+02(5.4859E+00)	3.4631E-01(4.0999E-02)	5.2890E-01(3.3480E-02)	5.0265E-01(3.9910E-02)
PSC-RCE	2.2046E+02(3.8618E+00)	3.6084E-01(1.6308E-02)	5.3733E-01(1.7838E-02)	5.1729E-01(2.2432E-02)
PSOLFK	2.1632E+02(1.6927E+00)	3.7783E-01(2.0572E-02)	5.4635E-01(2.1435E-02)	5.2850E-01(2.7257E-02)
PSOSCALFK	2.1620E+02(1.5675E+00)	3.8290E-01(2.0175E-02)	5.4743E-01(2.1817E-02)	5.3037E-01(2.9008E-02)
KMM	2.3326E+02(1.0375E+01)	3.5115E-01(2.9039E-02)	**5.6110E-01(1.7502E-02)**	5.0857E-01(1.3391E-02)
GLPSOK	**2.1514E+02(1.6923E+00)**	**3.8389E-01(1.8353E-02)**	5.5121E-01(1.7929E-02)	**5.3520E-01(2.2851E-02)**

TABLE 9. Statistical results obtained by nine algorithms on data set WDBC in 30 independent runs.

Algorithm	SED	NMI	F-measure	AC
	Mean(SD)	Mean(SD)	Mean(SD)	Mean(SD)
K-means	1.5265E+05(1.1841E-10)	4.2229E-01(2.2584E-16)	8.4434E-01(5.6460E-16)	8.5413E-01(1.1292E-16)
BIRCH	1.7176E+05(0.0000E+00)	2.6215E-01(5.6460E-17)	7.4835E-01(2.2584E-16)	7.7856E-01(0.0000E+00)
HPSOK-means	2.5144E+05(2.3379E+03)	1.2260E-01(6.2353E-03)	6.3034E-01(2.7753E-03)	5.2830E-01(7.1142E-03)
MinMax	1.5606E+05(5.9203E-11)	3.6670E-01(1.1292E-16)	8.1707E-01(4.5168E-16)	8.3128E-01(2.2584E-16)
PSC-RCE	1.5203E+05(7.4852E+02)	4.2393E-01(4.7502E-03)	8.4509E-01(2.1882E-03)	8.5477E-01(1.8738E-03)
PSOLFK	1.5077E+05(5.5759E+02)	4.4040E-01(1.7306E-02)	8.5365E-01(8.8531E-03)	8.6210E-01(7.5800E-03)
PSOSCALFK	1.4953E+05(7.1627E+01)	4.5869E-01(2.2584E-16)	8.6060E-01(1.1292E-16)	8.6819E-01(4.5168E-16)
KMM	2.1576E+05(4.4363E+04)	1.6658E-01(1.7564E-01)	7.3213E-01(7.3603E-02)	7.2050E-01(9.1403E-02)
GLPSOK	**1.4948E+05(2.4936E+01)**	**4.5869E-01(2.2584E-16)**	**8.6060E-01(1.1292E-16)**	**8.6819E-01(4.5168E-16)**

TABLE 10. Statistical results obtained by nine algorithms on data set CMC in 30 independent runs.

Algorithm	SED	NMI	F-measure	AC
	Mean(SD)	Mean(SD)	Mean(SD)	Mean(SD)
K-means	5.5431E+03(1.4594E+00)	3.2432E-02(5.7265E-04)	4.0338E-01(1.6791E-03)	**3.9819E-01(2.2462E-03)**
BIRCH	5.7516E+03(9.2504E-13)	3.1560E-02(2.1173E-17)	**4.1889E-01(1.1292E-16)**	3.9579E-01(5.6460E-17)
HPSOK-means	5.5330E+03(2.9686E-01)	3.1606E-02(1.2540E-03)	4.0163E-01(5.9880E-04)	3.9613E-01(1.1098E-03)
MinMax	5.5426E+03(0.0000E+00)	3.0597E-02(1.4115E-17)	4.0011E-01(5.6460E-17)	3.9172E-01(1.1292E-16)
PSC-RCE	5.5457E+03(2.9071E+00)	3.1918E-02(7.7300E-04)	4.0386E-01(2.3923E-03)	3.9717E-01(3.1917E-03)
PSOLFK	5.5415E+03(9.6531E-01)	**3.2825E-02(7.0575E-18)**	4.0245E-01(2.8230E-16)	3.9715E-01(1.6938E-16)
PSOSCALFK	5.5340E+03(1.5397E+00)	3.1500E-02(4.0573E-04)	4.0094E-01(7.9057E-04)	3.9507E-01(1.1266E-03)
KMM	6.1044E+03(8.4081E+02)	2.8181E-02(4.0513E-03)	4.1762E-01(2.9709E-02)	3.9396E-01(1.9413E-02)
GLPSOK	**5.5323E+03(2.4391E-01)**	3.1329E-02(1.4115E-17)	4.0054E-01(2.8230E-16)	3.9443E-01(2.8230E-16)

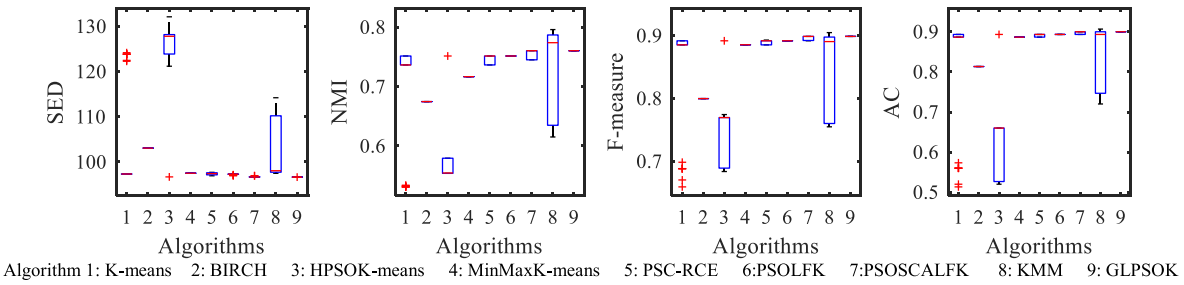


FIGURE 8. Box plots of SED, NMI, F-measure and AC obtained by nine algorithms on data set Iris with 30 independent runs.

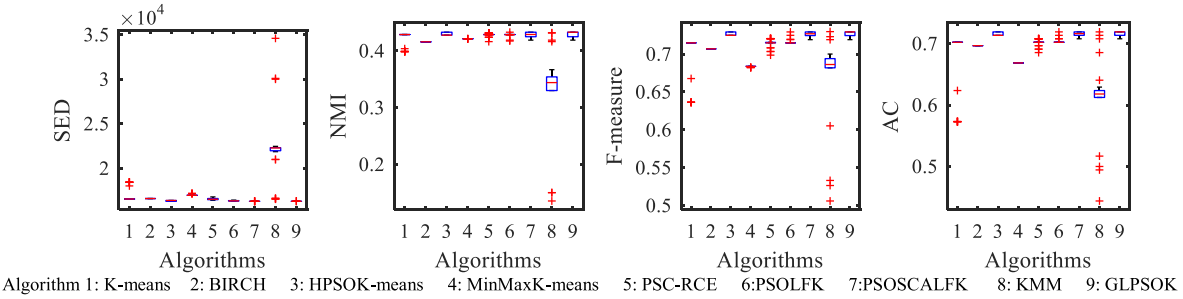


FIGURE 9. Box plots of SED, NMI, F-measure and AC obtained by nine algorithms on data set Wine with 30 independent runs.

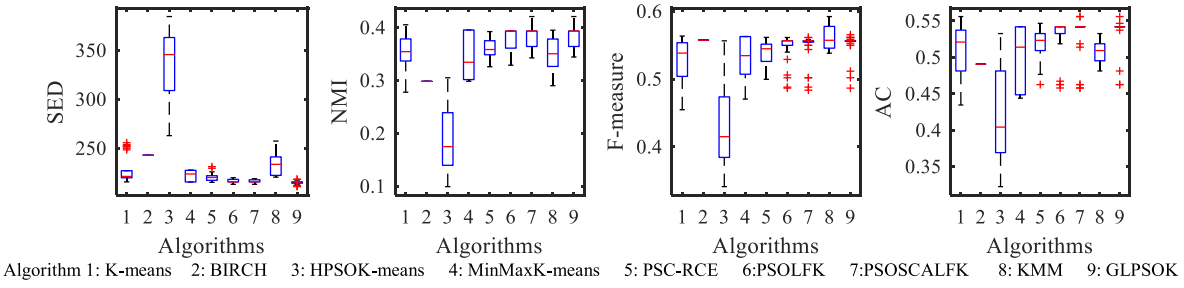


FIGURE 10. Box plots of SED, NMI, F-measure and AC obtained by nine algorithms on data set Glass with 30 independent runs.

other eight algorithms for all evaluation metrics. In addition, the standard deviation of experimental results obtained by GLPSOK is very small, which proves that the proposed algorithm is stable on this data set. Table 7 shows the experimental results on data set Wine. For metrics F-measure and AC, the proposed algorithm outperforms all other eight algorithms. For metric SED, PSOSCALFK achieves the best results, slightly better than our proposed algorithm. For metric NMI, HPSOK-means achieves the best results, slightly better than our proposed algorithm. Table 8 shows the

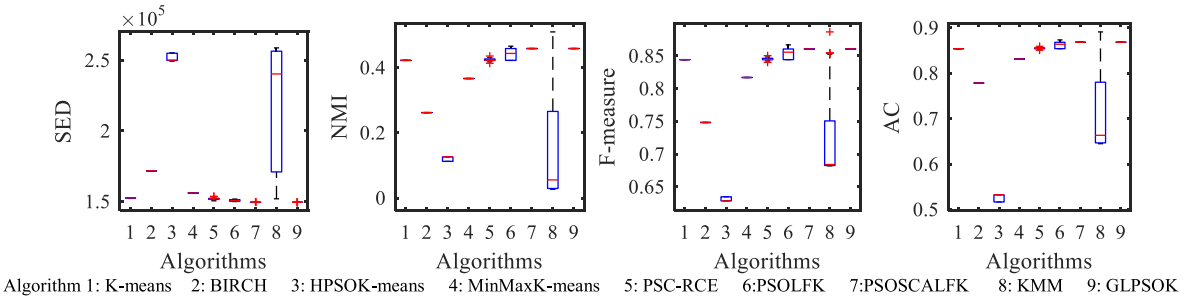


FIGURE 11. Box plots of SED, NMI, F-measure and AC obtained by nine algorithms on data set WDBC with 30 independent runs.

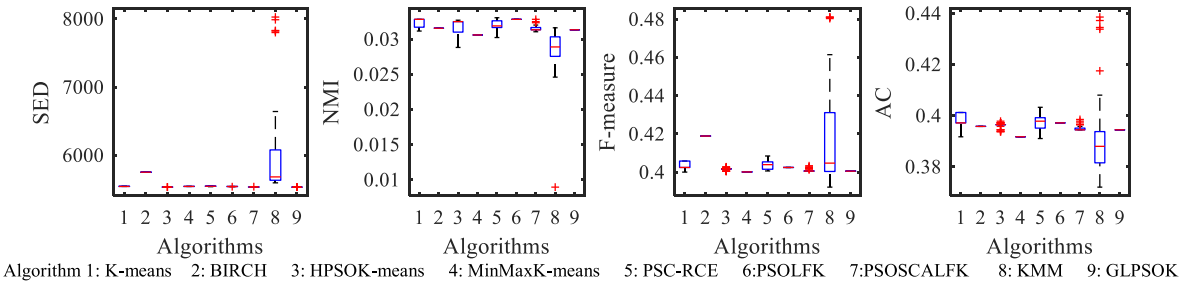


FIGURE 12. Box plots of SED, NMI, F-measure and AC obtained by nine algorithms on data set CMC with 30 independent runs.

experimental results on data set Glass. For metrics SED, NMI and AC, the proposed algorithm outperforms all other eight algorithms. For metric F-measure, KMM achieves the best results, followed by BIRCH, but only a little better than our proposed algorithm. Other algorithms all get worse results than our proposed algorithm. Table 9 shows that the proposed algorithm outperforms all other comparison algorithms for all evaluation metrics on data set WDBC. Table 10 shows that GLPSOK underperforms some other algorithms in most metrics on the date set CMC. This is because the data in CMC are categorical values. For example, one of the attributes is ''Wife's now working?'' where ''0'' represents ''not working'' and ''1'' represents ''working''. However, GLPSOK is a distance-based clustering algorithm, which is suitable for continuous numeric data and not suitable for categorical data. The traditional way to treat categorical attributes as numeric does not always produce meaningful results because many categorical domains are not ordered.

To better observe the overall results and robustness of all algorithms, box plots of SED, NMI, F-measure and AC obtained by eight algorithms on data set Iris, Wine, Glass, WDBC, CMC, are shown. According to Fig.8, Fig.9, and Fig.11, compared with the eight comparison algorithms, the overall results obtained by GLPSOK are the best on the data sets Iris, Wine and WDBC. In addition, the whole distribution of experimental results obtained by our proposed algorithm is very concentrated, with almost no outliers, indicating that our proposed algorithm has strong robustness and stability on these three data sets; From Fig. 10, we can see that GLPSOK also obtains the best overall results on the data set Glass. Fig.12 shows that the clustering result of GLPSOK is at a medium level on data set CMC. In summary, GLPSOK achieves the best overall results and proved to be very robust.

In order to compare the experimental results more scientifically and comprehensively, the Wilcoxon signed rank test [55], a non-parametric hypothesis test method, was used for a paired difference test. This test can quantitatively indicate whether there is a significant difference in the performance between algorithms. It is noted that the smaller the value of SED is, the better the quality of the clustering results will be. In contrast, the larger the values of NMI, F-Measure and AC are, the better the quality of the clustering results will be. With the significance level $\alpha = 0.05$, the results of the Wilcoxon signed rank test are shown in Table 11. The meaning of each symbol in the table is explained as follows: 'p-value' is the probability that the random variable has values more 'extreme' than the currently observed value under the null hypothesis; 'w+' is the sum of the rank greater than 0; 'w−' is the sum of the rank less than 0; R is the result of the Wilcoxon signed rank test, where '+' means GLPSOK outperforms other algorithms, '−' means GLPSOK is worse than other algorithms and '=' means GLPSOK is similar to other algorithms. As can be seen from Table 11, on the whole, GLPSOK is significantly better than the other eight algorithms.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose a hybrid PSO-K-means clustering algorithm that adopts GEDM and Lévy flight strategy to improve the performance of the algorithm. The experimental results on six synthetic data sets show that the proposed algorithm can obtain good clustering results on all synthetic

TABLE 11. (a)The results of the Wilcoxon signed rank test, with the significance level $\alpha = 0.05$. (b) The results of the Wilcoxon signed rank test, with the significance level $\alpha = 0.05$. (c) The results of the Wilcoxon signed rank test, with the significance level $\alpha = 0.05$.

(a) The results of the Wilcoxon signed rank test, with the significance level α = 0.05.

Metric	Dataset	K-means vs. GLPSOK				BIRCH vs. GLPSOK				HPSOK-means vs. GLPSOK			
		p-value	w+	w-	R	p-value	w+	w-	R	p-value	w+	w-	R
SED	Iris	1.24E-06	465	0	+	6.80E-08	465	0	+	1.73E-06	465	0	+
	Wine	1.73E-06	465	0	+	1.73E-06	465	0	+	8.94E-04	394	71	+
	Glass	4.23E-06	405	1	+	1.29E-06	465	0	+	1.73E-06	465	0	+
	WDBC	1.73E-06	465	0	+	1.73E-06	465	0	+	1.73E-06	465	0	+
	CMC	1.73E-06	465	0	+	1.73E-06	465	0	+	3.52E-06	458	7	+
NMI	Iris	1.24E-06	0	465	+	4.32E-08	0	465	+	9.94E-07	0	465	+
	Wine	7.71E-05	44	421	+	6.63E-07	0	465	+	1.00E+00	105	105	=
	Glass	1.88E-04	39	367	+	1.19E-06	0	465	+	1.73E-06	0	465	+
	WDBC	4.32E-08	0	465	+	4.32E-08	0	465	+	4.77E-07	0	465	+
	CMC	7.22E-07	464	1	-	4.32E-08	465	0	-	3.49E-01	243	163	-
F-measure	Iris	1.24E-06	0	465	+	4.32E-08	0	465	+	1.19E-06	0	465	+
	Wine	1.08E-06	0	465	+	6.63E-07	0	465	+	1.00E+00	105	105	=
	Glass	1.81E-03	66	340	+	3.06E-02	336	129	-	1.92E-06	1	464	+
	WDBC	4.32E-08	0	465	+	4.32E-08	0	465	+	4.77E-07	0	465	+
	CMC	7.22E-07	464	1	-	4.32E-08	465	0	-	3.99E-06	400	6	-
AC	Iris	1.24E-06	0	465	+	4.32E-08	0	465	+	1.09E-06	0	465	+
	Wine	1.08E-06	0	465	+	6.63E-07	0	465	+	3.02E-01	40	80	+
	Glass	2.79E-03	51.5	273.5	+	6.66E-07	6	459	+	1.72E-06	0	465	+
	WDBC	4.32E-08	0	465	+	4.32E-08	0	465	+	4.77E-07	0	465	+
	CMC	1.84E-06	454	11	-	4.32E-08	465	0	-	5.73E-06	372	6	=
+/=/-		17/0/3				16/0/4				15/3/2			

(b) The results of the Wilcoxon signed rank test, with the significance level α = 0.05.

Metric	Dataset	MinMaxK-means vs. GLPSOK				PSC-RCE vs. GLPSOK				PSOLFK vs. GLPSOK			
		p-value	w+	w-	R	p-value	w+	w-	R	p-value	w+	w-	R
SED	Iris	6.80E-08	465	0	+	1.73E-06	465	0	+	8.09E-07	465	0	+
	Wine	1.73E-06	465	0	+	1.73E-06	465	0	+	1.73E-06	465	0	+
	Glass	2.13E-05	439	26	+	3.88E-06	457	8	+	6.79E-03	227	49	+
	WDBC	1.73E-06	465	0	+	1.73E-06	465	0	+	1.73E-06	465	0	+
	CMC	1.73E-06	465	0	+	1.73E-06	465	0	+	1.73E-06	465	0	+
NMI	Iris	4.32E-08	0	465	+	8.58E-07	0	465	+	4.32E-08	0	465	+
	Wine	1.34E-06	3	462	+	1.49E-01	163	302	+	5.83E-04	64	371	+
	Glass	3.55E-04	59	406	+	1.74E-04	50	415	+	2.89E-01	85	146	+
	WDBC	4.32E-08	0	465	+	1.09E-06	0	465	+	5.08E-05	18	333	+
	CMC	4.32E-08	0	465	+	2.23E-03	381	84	-	4.32E-08	465	0	-
F-measure	Iris	4.32E-08	0	465	+	8.58E-07	0	465	+	4.32E-08	0	465	+
	Wine	9.73E-07	0	465	+	1.41E-06	0	465	+	1.54E-06	3	432	+
	Glass	3.13E-03	89	376	+	3.61E-03	91	374	+	2.44E-01	82	149	+
	WDBC	4.32E-08	0	465	+	1.09E-06	0	465	+	8.63E-04	46	305	+
	CMC	4.32E-08	0	465	+	1.69E-06	465	0	-	4.32E-08	465	0	-
AC	Iris	4.32E-08	0	465	+	4.77E-07	0	465	+	4.32E-08	0	465	+
	Wine	6.63E-07	0	465	+	1.26E-06	0	465	+	2.07E-06	0	378	+
	Glass	2.82E-04	15	238	+	3.27E-03	74	332	+	1.78E-01	30.5	74.5	+
	WDBC	4.32E-08	0	465	+	1.09E-06	0	465	+	6.38E-04	37.5	287.5	+
	CMC	4.32E-08	0	465	+	2.72E-04	409	56	-	4.32E-08	465	0	-
+/=/-		20/0/0				17/0/3				17/0/3			

(c) The results of the Wilcoxon signed rank test, with the significance level α = 0.05.

Metric	Dataset	PSOSCALFK vs. GLPSOK				KMM vs. GLPSOK			
		p-value	w+	w-	R	p-value	w+	w-	R
SED	Iris	1.73E-06	465	0	+	1.66E-06	465	0	+
	Wine	2.96E-03	88	377	-	1.73E-06	465	0	+
	Glass	1.26E-02	220	56	+	1.73E-06	465	0	+
	WDBC	8.92E-05	423	42	+	1.73E-06	465	0	+
	CMC	1.73E-06	465	0	+	1.73E-06	465	0	+
NMI	Iris	4.88E-04	0	78	+	3.37E-01	186	279	+
	Wine	1.64E-01	34	86	+	1.69E-06	0	465	+
	Glass	6.59E-01	123.5	152.5	+	2.61E-04	55	410	+
	WDBC	0.00E+00	0	0	=	3.43E-06	7	458	+
	CMC	7.81E-03	52	3	-	2.88E-06	5	460	+
F-measure	Iris	4.88E-04	0	78	+	1.06E-05	19	446	+
	Wine	1.96E-01	36	84	+	1.87E-06	1	464	+
	Glass	3.69E-01	108.5	167.5	+	1.99E-01	295	170	-
	WDBC	0.00E+00	0	0	=	3.43E-06	7	458	+
	CMC	1.95E-03	55	0	-	5.67E-03	367	98	-
AC	Iris	4.88E-04	0	78	+	1.34E-04	10	243	+
	Wine	1.76E-01	35	85	+	2.39E-06	0	435	+
	Glass	5.05E-01	35.5	55.5	+	2.96E-04	57	408	+
	WDBC	0.00E+00	0	0	=	3.43E-06	7	458	+
	CMC	1.95E-03	55	0	-	2.13E-01	172	293	+
+/=/-		13/3/4				18/0/2			

data sets. In addition, the proposed algorithm was compared with several classic or state-of-the-art clustering algorithms on five real-word data sets from the UCI Machine Learning Repository. The experimental results show that the proposed algorithm can achieve better performance than the six comparison algorithms on real-world data sets and has strong robustness. However, GLPSOK consumes a lot of time when dealing with large-scale data set. Future work can be summarized into two aspects: (1) how to further improve the convergence efficiency of GLPSOK; (2) how to apply the proposed GLPSOK algorithm to practical scenarios like image segmentation, and applications including clustering in the financial, ecological, and educative domains from data pressed in natural language texts.

REFERENCES

[1] A. Bouyer and A. Hatamlou, "An efficient hybrid clustering method based on improved cuckoo optimization and modified particle swarm optimization algorithms," *Appl. Soft Comput.*, vol. 67, pp. 172–182, Jun. 2018.
[2] S. Zhang, Z. Yang, X. Xing, Y. Gao, D. Xie, and H.-S. Wong, "Generalized pair-counting similarity measures for clustering and cluster ensembles," *IEEE Access*, vol. 5, pp. 16904–16918, 2017.
[3] L. Chen, Q. Jiang, and S. Wang, "Model-based method for projective clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 7, pp. 1291–1305, Jul. 2012.
[4] F. Zhao, Y. Chen, H. Liu, and J. Fan, "Alternate PSO-based adaptive interval type-2 intuitionistic fuzzy C-Means clustering algorithm for color image segmentation," *IEEE Access*, vol. 7, pp. 64028–64039, 2019.
[5] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An efficient data clustering method for very large databases," *ACM SIGMOD Rec.*, vol. 25, pp. 103–114, Jun. 1996.
[6] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *J. Cybern.*, vol. 3, no. 3, pp. 32–57, Jan. 1973.
[7] C. Wang, W. Pedrycz, J. Yang, M. Zhou, and Z. Li, "Wavelet frame-based fuzzy C-Means clustering for segmenting images on graphs," *IEEE Trans. Cybern.*, early access, Jul. 10, 2019, doi: 10.1109/TCYB.2019.2921779.
[8] V. Saveetha, S. Sophia, and P. D. R. Vijayakumar, "Appliance of effective clustering technique for gene expression datasets using GPU," *Cluster Comput.*, vol. 22, no. S5, pp. 12381–12388, Sep. 2019.
[9] X. Zhang, H. Gao, G. Li, J. Zhao, J. Huo, J. Yin, Y. Liu, and L. Zheng, "Multi-view clustering based on graph-regularized nonnegative matrix factorization for object recognition," *Inf. Sci.*, vol. 432, pp. 463–478, Mar. 2018.
[10] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.*, Jun. 1967, pp. 281–297.
[11] B. Zhang, M. Hsu, and U. Dayal, "*K*-harmonic means—A spatial clustering algorithm with boosting," in *Temporal, Spatial, and Spatio-Temporal Data Mining* (Lecture Notes in Computer Science). Berlin, Germany: Springer, 2001, pp. 31–45.
[12] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization," *Swarm Intell.*, vol. 1, no. 1, pp. 33–57, Jun. 2007.
[13] W. Dong and M. C. Zhou, "A supervised learning and control method to improve particle swarm optimization algorithms," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 7, pp. 1149–1159, Jul. 2017.
[14] M. Dorigo and T. Stützle, "Ant colony optimization: Overview and recent advances," in *Handbook of Metaheuristics* (International Series in Operations Research & Management Science). Cham, Switzerland: Springer, 2019, pp. 311–351.
[15] J. Zhao, S. Liu, M. Zhou, X. Guo, and L. Qi, "Modified cuckoo search algorithm to solve economic power dispatch optimization problems," *IEEE/CAA J. Automatica Sinica*, vol. 5, no. 4, pp. 794–806, Jul. 2018.
[16] R. Tinos, L. Zhao, F. Chicano, and D. Whitley, "NK hybrid genetic algorithm for clustering," *IEEE Trans. Evol. Comput.*, vol. 22, no. 5, pp. 748–761, Oct. 2018.
[17] Y. Yu, S. Gao, Y. Wang, and Y. Todo, "Global optimum-based search differential evolution," *IEEE/CAA J. Automatica Sinica*, vol. 6, no. 2, pp. 379–394, Mar. 2019.
[18] Y. Cao, H. Zhang, W. Li, M. Zhou, Y. Zhang, and W. A. Chaovalitwongse, "Comprehensive learning particle swarm optimization algorithm with local search for multimodal functions," *IEEE Trans. Evol. Comput.*, vol. 23, no. 4, pp. 718–731, Aug. 2019.
[19] X. Liang, W. Li, Y. Zhang, and M. Zhou, "An adaptive particle swarm optimization method based on clustering," *Soft Comput.*, vol. 19, no. 2, pp. 431–448, Feb. 2015.
[20] M. Alswaitti, M. Albughdadi, and N. A. M. Isa, "Density-based particle swarm optimization algorithm for data clustering," *Expert Syst. Appl.*, vol. 91, pp. 170–186, Jan. 2018.
[21] D. W. van der Merwe and A. P. Engelbrecht, "Data clustering using particle swarm optimization," in *Proc. Congr. Evol. Comput. CEC*, Dec 2003, pp. 215–220.
[22] H. Izakian and A. Abraham, "Fuzzy C-means and fuzzy swarm for fuzzy clustering problem," *Expert Syst. Appl.*, vol. 38, no. 3, pp. 1835–1838, Mar. 2011.
[23] A. Ahmadyfard and H. Modares, "Combining PSO and k-means to enhance data clustering," in *Proc. Int. Symp. Telecommun.*, Aug. 2008, pp. 688–691.
[24] T. Niknam and B. Amiri, "An efficient hybrid approach based on PSO, ACO and k-means for cluster analysis," *Appl. Soft Comput.*, vol. 10, no. 1, pp. 183–197, Jan. 2010.
[25] P. S. Deepthi and S. M. Thampi, "Unsupervised gene selection using particle swarm optimization and k-means," in *Proc. 2nd ACM IKDD Conf. Data Sci. CoDS*, 2015, pp. 134–135.
[26] X. Xu, J. Li, M. Zhou, J. Xu, and J. Cao, "Accelerated two-stage particle swarm optimization for clustering not-well-separated data," *IEEE Trans. Syst., Man, Cybern. Syst.*, early access, Jun. 29, 2019, doi: 10.1109/TSMC.2018.2839618.
[27] W. Liu, Z. Wang, X. Liu, N. Zeng, and D. Bell, "A novel particle swarm optimization approach for patient clustering from emergency departments," *IEEE Trans. Evol. Comput.*, vol. 23, no. 4, pp. 632–644, Aug. 2019.
[28] F. Yang, T. Sun, and C. Zhang, "An efficient hybrid data clustering method based on K-harmonic means and Particle Swarm Optimization," *Expert Syst. Appl.*, vol. 36, no. 6, pp. 9847–9852, Aug. 2009.
[29] M. Danesh, M. Naghibzadeh, M. R. A. Totonchi, M. Danesh, B. Minaei, and H. Shirgahi, "Data clustering based on an efficient hybrid of K-harmonic means, PSO and GA," in *Transactions on Computational Collective Intelligence IV* (Lecture Notes in Computer Science). Berlin, Germany: Springer, 2011, pp. 125–140.
[30] Q. Niu and X. Huang, "An improved fuzzy C-means clustering algorithm based on PSO," *J. Softw.*, vol. 6, no. 5, pp. 873–879, May 2011.
[31] S. Alam, G. Dobbie, and P. Riddle, "An evolutionary particle swarm optimization algorithm for data clustering," in *Proc. IEEE Swarm Intell. Symp.*, Sep. 2008, pp. 1–6.
[32] A. Szabo, A. K. F. Prior, and L. N. de Castro, "The proposal of a velocity memoryless clustering swarm," in *Proc. IEEE Congr. Evol. Comput.*, Jul. 2010, pp. 1–5.
[33] M. Yuwono, S. W. Su, B. D. Moulton, and H. T. Nguyen, "Data clustering using variants of rapid centroid estimation," *IEEE Trans. Evol. Comput.*, vol. 18, no. 3, pp. 366–377, Jun. 2014.
[34] J. E. Gentle, L. Kaufman, and P. J. Rousseuw, "Finding groups in data: An introduction to cluster analysis," *Biometrics*, vol. 47, no. 2, p. 788, Jun. 1991.
[35] E. Figueiredo, M. Macedo, H. V. Siqueira, C. J. Santana, A. Gokhale, and C. J. A. Bastos-Filho, "Swarm intelligence for clustering—A systematic review with new perspectives on data mining," *Eng. Appl. Artif. Intell.*, vol. 82, pp. 313–329, Jun. 2019.
[36] B. Niu, Q. Duan, J. Liu, L. Tan, and Y. Liu, "A population-based clustering technique using particle swarm optimization and k-means," *Natural Comput.*, vol. 16, no. 1, pp. 45–59, Mar. 2017.
[37] F. Salajegheh and E. Salajegheh, "PSOG: Enhanced particle swarm optimization by a unit vector of first and second order gradient directions," *Swarm Evol. Comput.*, vol. 46, pp. 28–51, May 2019.
[38] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," in *Evolutionary Programming VII* (Lecture Notes in Computer Science). Berlin, Germany: Springer, 1998, pp. 591–600.
[39] X. Wang, H. Zhao, T. Han, Z. Wei, Y. Liang, and Y. Li, "A Gaussian estimation of distribution algorithm with random walk strategies and its application in optimal missile guidance handover for multi-UCAV in over-the-horizon air combat," *IEEE Access*, vol. 7, pp. 43298–43317, 2019.

[40] Y. Li, T. Han, H. Zhao, and H. Gao, "An adaptive whale optimization algorithm using Gaussian distribution strategies and its application in heterogeneous UCAVs task allocation," *IEEE Access*, vol. 7, pp. 110138–110158, 2019.
[41] Y. Hariya, T. Kurihara, T. Shindo, and K. Jin'no, "Lévy flight PSO," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, May 2015, pp. 2678–2684.
[42] R. N. Mantegna, "Fast, accurate algorithm for numerical simulation of Lévy stable stochastic processes," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 49, no. 5, pp. 4677–4683, May 1994.
[43] W. Chu, X. Gao, and S. Sorooshian, "Handling boundary constraints for particle swarm optimization in high-dimensional search space," *Inf. Sci.*, vol. 181, no. 20, pp. 4569–4581, Oct. 2011.
[44] S. Ghosh and S. Kumar, "Comparative analysis of K-means and fuzzy C-means algorithms," *Int. J. Adv. Comput. Sci. Appl.*, vol. 4, no. 4, pp. 35–39, 2013.
[45] G. Tzortzis and A. Likas, "The MinMax k-means clustering algorithm," *Pattern Recognit.*, vol. 47, no. 7, pp. 2505–2516, Jul. 2014.
[46] F. Nie, C.-L. Wang, and X. Li, "K-Multiple-means: A multiple-means clustering method with specified k clusters," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 959–967.
[47] C. Fang, W. Jin, and J. Ma, "κ'-means algorithms for clustering analysis with frequency sensitive discrepancy metrics," *Pattern Recognit. Lett.*, vol. 34, no. 5, pp. 580–586, Apr. 2013.
[48] Y. M. B. Ali, "Unsupervised clustering based an adaptive particle swarm optimization algorithm," *Neural Process. Lett.*, vol. 44, no. 1, pp. 221–244, Aug. 2016.
[49] D. Dua and K. T. Efi. *UCI Machine Learning Repository*. Accessed: Mar. 20, 2019. [Online]. Available: http://archive.ics.uci.edu/ml
[50] A. M. Almalawi, A. Fahad, Z. Tari, M. A. Cheema, and I. Khalil, "KNNVWC: An efficient k-nearest neighbors approach based on various-widths clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 1, pp. 68–81, Jan. 2016.
[51] B. Vandeginste,"PARVUS: An extendable package of programs for data exploration, classification and correlation, M. Forina, R. Leardi, C. Armanino and S. Lanteri, Elsevier, Amsterdam, 1988, price: US 645 ISBN 0-444-43012-1," *J. Chemometrics*, vol. 4, no. 2, pp. 191–193, Mar. 1990.
[52] R. Jensi and G. W. Jiji, "An enhanced particle swarm optimization with levy flight for global optimization," *Appl. Soft Comput.*, vol. 43, pp. 248–261, Jun. 2016.
[53] S. N. Chegini, A. Bagheri, and F. Najafi, "PSOSCALF: A new hybrid PSO based on sine cosine algorithm and levy flight for solving optimization problems," *Appl. Soft Comput.*, vol. 73, pp. 697–726, Dec. 2018.
[54] D. Cai, X. He, and J. Han, "Document clustering using locality preserving indexing," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 12, pp. 1624–1637, Dec. 2005.
[55] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bull.*, vol. 1, no. 6, pp. 80–83, 1945.

HANJIE GAO received the B.S. degree from the College of Computer Science and Technology, Jilin University, China, in 2018, where she is currently pursuing the M.S. degree. Her research interests include machine learning, evolutionary computing, and knowledge graph.

YINTONG LI received the M.S. degree from Air Force Engineering University, China, in 2019. His research interests include artificial intelligence optimization algorithms and UCAV maneuvering decision

PETR KABALYANTS received the M.S. degree in mathematics from Kharkiv State University (now V.N. Karazin Kharkiv National University), Ukraine, in 1993, and the Ph.D. degree in engineering specialized in mathematical modeling and computational mathematics from the Kharkiv National University of Radio Electronics, in 2007. Since 2007, he has been an Assistant Professor with the Mathematics and Computer Science School, V. N. Karazin Kharkiv National University. Since 2018, he has been an Assistant Professor with the Department of Software Engineering for Computers and Computer-Based Systems, V. G. Shukhov Belgorod State Technological University, Belgorod, Russia. His research interests include developing and analyzing network growth models, random networks, probabilistic methods, and machine learning.

HAO XU received the B.S. and M.E. degrees from the College of Computer Science and Technology, Jilin University, China, in 2005 and 2008, respectively, and the Ph.D. from the Department of Information Engineering and Computer Science, University of Trento, Italy, in 2012. He is currently a Professor with the College of Computer Science and Technology, Jilin University. His research interests include human-centered AI, knowledge graph, and human-computer interaction.

RODRIGO MARTÍNEZ-BÉJAR received the M.S. degree in computer science from the University of Malaga, in 1992, and the M.S. degree in applied sociology and the Ph.D. degree from the University of Murcia, in 1997. He is currently a Professor of computer science and artificial intelligence from the Department of Information and Communication Engineering, University of Murcia. He has held visiting positions at several European, Australian, and Latin-American Universities, including the Australian Universities of Adelaide and New South Wales, and the French University of Southern Paris. In 2006, he became the Head of the institutional and inter-disciplinary university research group. He has served on several Editorial Boards and has coauthored more than 100 scientific publications, most of them in peer-reviewed journals and conferences and led more than 30 national or international competitive grants and research contracts with industry.