Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 6748056, 15 pages
https://doi.org/10.1155/2020/6748056

Research Article

A Novel Hierarchical Clustering Approach Based on Universal Gravitation

Peng Zhang[1,2] **and Kun She**[1]

[1]*School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China*
[2]*School of Science, Southwest University of Science and Technology, Mianyang 621010, China*

Correspondence should be addressed to Kun She; kun@uestc.edu.cn

Received 20 May 2019; Revised 29 October 2019; Accepted 24 December 2019; Published 1 February 2020

Academic Editor: Bogdan Smolka

The target of the clustering analysis is to group a set of data points into several clusters based on the similarity or distance. The similarity or distance is usually a scalar used in numerous traditional clustering algorithms. Nevertheless, a vector, such as data gravitational force, contains more information than a scalar and can be applied in clustering analysis to promote clustering performance. Therefore, this paper proposes a three-stage hierarchical clustering approach called GHC, which takes advantage of the vector characteristic of data gravitational force inspired by the law of universal gravitation. In the first stage, a sparse gravitational graph is constructed based on the top k data gravitations between each data point and its neighbors in the local region. Then the sparse graph is partitioned into many subgraphs by the gravitational influence coefficient. In the last stage, the satisfactory clustering result is obtained by merging these subgraphs iteratively by using a new linkage criterion. To demonstrate the performance of GHC algorithm, the experiments on synthetic and real-world data sets are conducted, and the results show that the GHC algorithm achieves better performance than the other existing clustering algorithms.

1. Introduction

Clustering is one of the major unsupervised learning techniques and has been applied in many fields such as pattern recognition [1], image processing [2, 3], community detection [4, 5], bioinformatics [6, 7], information retrieval [8, 9], and so on. The main task of clustering is to classify a dataset into some nonoverlapping clusters based on a suitable similarity metric so that the elements in the same cluster are similar, while any elements from different clusters are dissimilar. A range of various clustering methods have been proposed and classified as partition-based, hierarchical, grid-based, density-based, model-based clustering, and so on.

K-means [10] and its successors [2, 11] are typical partition-based clustering approaches. They need to be specified the number of clusters in advance. Each data point of the dataset is assigned to its closest cluster according to the Euclidean distances among data points. The new centroids of clusters are repeated to be calculated until the elements are consistently assigned to the same cluster. Then, the cluster centers have stabilized and will remain the same forever. However, these approaches are not able to detect nonspherical clusters because an element is always assigned to the nearest center. Numerous studies have been done to overcome the drawback of *K*-means type algorithms, particularly by using density distribution. In density-based clustering, clusters which have arbitrary shape are considered as the dense regions separated by sparse region in data space [12]. DBSCAN [13] is the most representative density-based clustering algorithm that needs to be specified a density threshold, discards the points with densities lower than this threshold as noises, and assigns to different clusters disconnected regions of high density. DP [14] is a novel algorithm that efficiently discovers the centers of clusters by finding the density peaks. It assumes that cluster centers are surrounded by neighbors with lower local density and are at a relatively large distance from any points with a higher local density. Furthermore, hierarchical clustering is a significant method of cluster analysis which seeks to build a hierarchy of clusters. The hierarchical clustering algorithms can be divided into two categories including agglomerative and divisive algorithms. Agglomerative hierarchical clustering algorithm starts with every single element in a dataset. Then it aggregates the closest clusters with a linkage criterion in each iteration until all elements form one cluster. The divisive hierarchical clustering algorithm starts with the dataset considered a single cluster which is separated into many subclusters until every element forms a cluster. The other differences among hierarchical clustering approaches are determined by the diverse choices of similarity criteria and the linkage criteria. BIRCH is one of the most effective hierarchical clustering methods [15]. It constructs a tree data structure with the cluster centroids being read off the leaf, which can be either the final cluster centroids or can be provided as input to another clustering algorithm. In addition, there are many multistage hierarchical clustering algorithms, such as Chameleon [16], which is a representative approach and can detect the arbitrary shape of the cluster effectively. In the first stage, Chameleon uses a graph-partitioning algorithm to cluster the data items into several relatively small subclusters. In the second stage, it is to find the genuine clusters by repeatedly combining these subclusters based on its selections on both interconnectivity and closeness. These classical clustering approaches usually only utilize one kind of internal evaluation function to determine clustering quality [17]. Many scholars focused on the study of multiobjective clustering to overcome the defect of conventional clustering algorithms. Peng et al. [18] proposed fuzzy multiobjective clustering based on PSO to obtain well-separated, connected, and compact clusters. Saha and Maulik [19] proposed the multiobjective clustering based on incremental learning for categorical data. Moreover, lots of clustering algorithms, such as DenPEHC [20], GHFHC [21], Muenc [22], and so on, were also put forward to improve the clustering performance. Meanwhile, new gravity-based clustering approaches were also proposed, such as the LGC algorithm, which would be discussed in Section 2.

It is an important task to design a new clustering algorithm because every algorithm has its own advantages and disadvantages. In conventional clustering algorithms, the similarity or distance is usually scalar, which only contains the partial information among data points. To obtain more information among data points, the vector can be adopted to represent the similarity of two data points. Data gravitational force, which is like the universal gravitational force, is employed to cluster data points. Then, we propose a novel hierarchical clustering based on the sparse gravitational graph in which the vertex denotes each object of a data set, and the edge denotes that data gravitation force exists between its two vertices. In the clustering process, a weighted graph firstly is constructed based on universal gravitation. Then the graph is divided into several subgraphs based on the gravitational influence coefficients between each vertex and its adjacent vertices. Finally, it is iterative to merge two subgraphs based on a new linkage measure until the genuine clusters are found.

There are three highlights in this paper as follows. At first, the sparse gravitational graph is defined based on the data gravitation model. Meanwhile, a new measure is used to extract more valuable information between each vertex and its adjacent vertices in the sparse gravitational graph. Secondly, a new linkage measure which makes the best of the data gravitation's characteristics is proposed to merge the subclusters iteratively. Thirdly, a novel three-stage gravity-based hierarchical clustering method named GHC is proposed. The GHC algorithm can be used to detect arbitrary clusters effectively and achieves an excellent clustering performance on the synthetic and real-life data sets in this study.

The remainder of the paper is organized as follows. In Section 2, the related work of gravity-based clustering is reviewed. The novel gravity-based hierarchical clustering (GHC) is proposed and analyzed in detail in Section 3. In Section 4, the experiments on the synthetic and real-world data sets are conducted and discussed. In Section 4, the conclusions are drawn.

2. Related Work of Gravity-Based Clustering

Using gravity theory in clustering is not a new idea. Numerous gravity-based clustering algorithm, which simulates the process of the attraction and merging of objects by their gravity force, has been studied. Usually, these algorithms consider each data point as an object and assign a mass to it in feature space. Wright [23] proposed the first version of gravitational clustering, which updates the position of each data point at each iteration and aggregates the data points into clusters when they are close. Yung [24] employ the gravitational clustering approach to segment color images. Each pixel with a unit mass maps to a location (as a particle) in RGB space. The mass of a particle is the total number of pixels mapped to it. The gravity causes the particles to move in the space under constraint. The particles are clustered when they move to the same location in RGB space. Wang et al. [25] proposed two novel clustering approaches based on the local gravitation model. In this model, each data item is considered as an object with mass and associated with a local resultant force (LRF) generated by its neighbors in the local region. The clustering process is realized by using the differences among the LRFs of the data points close to the cluster centers and at the boundary of the clusters. Bahrololoum et al. [26] proposed another approach that finds the best positions of the cluster centroids determined by employing the law of gravity. In the approach, the data points and cluster centroids are considered as fixed celestial objects and movable objects, respectively. The celestial objects apply a gravity force to the movable objects and change their positions in the feature space. The best cluster centroids are obtained until the sum of the forces on each centroid approaches zero. Mohammed Alswaitt et al. [27] proposed a modification over a gravity-based data clustering algorithm. The modified algorithm adopts the dependence of the agent on velocity and an initialization step of centroid positions to impose a balance between exploitation ability and exploration ability of gravity-based clustering approach. Besides, a serial of approaches based on gravity theory and Newton's second law of motion was proposed by Gómez et al. [28], Kundu [29], and Sanchez et al. [30]. In these approaches, points of the same cluster will move toward the direction of their cluster center. Inspired by the phenomena of gravitation and the black hole, Hatamlou [31] proposed a new heuristic optimization approach called the black hole algorithm. Other heuristic algorithms inspired by gravitational phenomena have been designed for clustering. For instance, a heuristic gravitational search algorithm (GSA) was proposed by Rashedi et al. [32] and was applied in solving wind-hydro-thermal CO problem by Shukla and Singh [33]. Yin et al. [34] designed a hybrid data clustering algorithm based on GSA.

To the best of our knowledge, each data point of a dataset is considered a movable object with mass in the most existing gravity-based clustering algorithms. Data points can move around in feature space in the influence of the law of gravity and merge into several clusters when they move close enough to each other. In our approach, we establish the data gravitation model and utilize the relation between each data point and its neighbors which exert the largest gravitational forces on it to group a dataset into many subclusters. And then, two subclusters with the largest resultant gravity force are merged. To boost the effectiveness of clustering, we define the sparse gravitational graph based on the data gravitation model, which can be divided into many subgraphs based on the relation between each vertex and its adjacent vertexes. Next, subgraphs can be repeatedly merged to form a larger subgraph until the terminal condition is satisfied.

3. Data Gravitation Model

Newton's law of universal gravitation states that every point mass attracts every other point mass with force acting along the line through those points, which is proportional to the product of their masses and inversely proportional to the square of the distance between them. The gravitational force can be calculated as follows:

$$\vec{P}_{ij} = \kappa \frac{\sigma_i \sigma_j}{\delta_{ij}^2} \hat{\delta}_{ij}, \tag{1}$$

where $\vec{P}_{ij}$ denotes the gravitational force exerted on point mass i by point mass j, σ_i and σ_j are the masses of the two points, respectively, δ_{ij} is the distance between point mass i and point mass j, $\hat{\delta}_{ij}$ is the unit vector from point mass i to j, and κ is the gravitational constant.

Similar to the gravitational force, it is assumed that data gravitation exists among any two data points in data space. The data gravitation can be given as follows:

$$\vec{F}_{ij} = \begin{cases} m_i m_j g\left(d_{ij}\right) \hat{d}_{ij}, & d_{ij} > 0, \\ 0, & d_{ij} = 0, \end{cases} \tag{2}$$

where $g(x)$ is a decreasing function of x ($g(x) = x^{-2}$), m_i and m_j are, respectively, the masses of the data points i and j, d_{ij} is the distance between the two data points, $\hat{d}_{ij}$ is the unit vector from data point i to j. The mass of data point i can be defined by

$$m_i = 1 + \sum_{j \neq i} Z\left(d_{ij} - \gamma\right), \tag{3}$$

where $Z(d_{ij} - \gamma) = 1$ if $d_{ij} < \gamma$ and $Z(d_{ij} - \gamma) = 0$ otherwise, and $\gamma (\gamma \geq 0)$ is a cutoff distance. In other words, m_i equals the number of points from which the distances to point i are less than γ. Especially, the mass of a data point is equal to 1 when $\gamma = 0$. Moreover, we assume that the gravitational forces exerted on a data point are the top k gravitational forces between it and other data points. Therefore, the gravitational resultant force (GRF) of data point i can be obtained as follows:

$$\vec{F}_i = \sum_{j \in \Omega_i} \vec{F}_{ij} = m_i \sum_{j \in \Omega_i} m_j g\left(d_{ij}\right) \hat{d}_{ij}, \tag{4}$$

where Ω_i is the set of neighboring data points which exert the top k gravitational forces on data point i. The gravitational force between two data points changes with the cutoff distance γ because their masses are related to γ. Then GRF of a data point also changes with γ according to equation (4). For example, Figure 1 shows the gravitational force and GRF when γ is specified to different values in a 2D data set $\{x_1, x_2, x_3, x_4, x_5\}$. In Figure 1(a), the mass of each data point is 1 when $\gamma = 1$. It can be noticed that the GRF of the data point x_1 is directed towards the data points x_2 and x_4. It indicates that the data points x_2 and x_4 provide more influence on x_1. In Figure 1(b), the masses of x_1, x_2, x_3, x_4, and x_5 are 4, 1, 2, 3, and 3, respectively. The GRF of the data point x_1 is directed towards the data points x_4 and x_5. It indicates that the data points x_4 and x_5 provide more influence on x_1. Then the gravitational influence coefficient (GIC) is introduced to represent the relationship between the RGF of a data point and the gravitational forces exerted on it by other data points. The GIC of data points i and j is defined as follows:

$$\mathrm{GIC}_j^i = \cos\left(\hat{F}_i, \hat{F}_{ij}\right) = \frac{\hat{F}_i \cdot \hat{F}_{ij}}{\left\|\hat{F}_i\right\| \times \left\|\hat{F}_{ij}\right\|}, \tag{5}$$

where $\vec{F}_i$ is the resultant force of data point i, $\vec{F}_{ij}$ is the gravitation force exerted on data point i by its neighboring data point j. GIC_j^i ranges from −1 to 1. The bigger the GIC_j^i, the point j provides more influence on data point i. Intuitively, the gravitational influence coefficient can be adopted to realize the data cluster analysis. The data point i and j are grouped into a cluster if there are the bigger GIC_j^i and GIC_i^j than a threshold. Otherwise, they are clustered into different clusters. In this way, a course clustering method can be obtained.

4. The Proposed Gravity-Based Hierarchical Clustering Algorithm

Though the course clustering algorithm based on the data gravitation model can be employed to cluster a dataset, its clustering performance is not good. Therefore, a novel

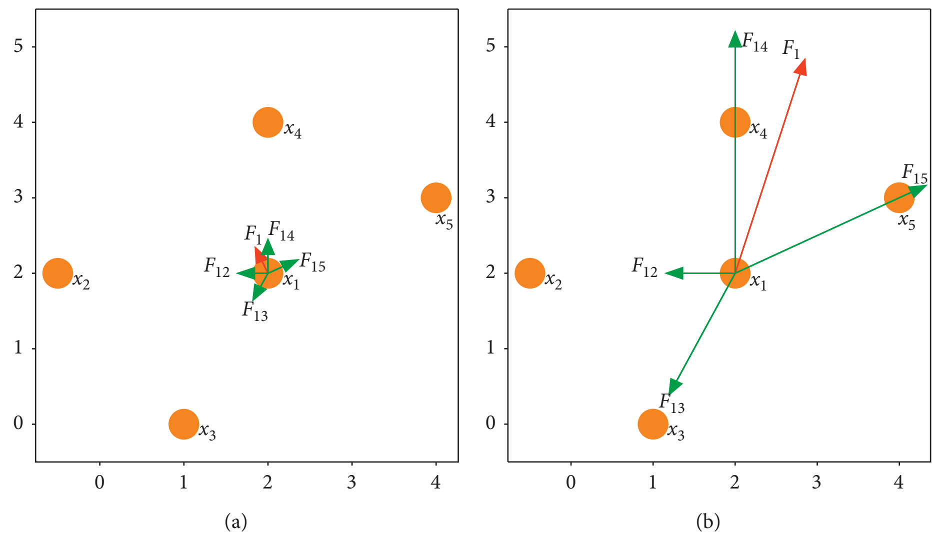


Figure 1: The gravitational resultant force of a data point and the gravitational forces between it and other data points when the cutoff distance γ is set to different values. (a) $\gamma = 1$ and $k = 4$. (b) $\gamma = 2.4$ and $k = 4$.

hierarchical clustering algorithm (GHC) is proposed based on sparse gravitational graph which can make the algorithm implement easily and perform effectively. The time complexity of GHC algorithm is analyzed at the end of this section.

4.1. Sparse Gravitational Graph. Let $X = \{x_1, x_2, \ldots, x_n\}$ denote a data set with n data points, in which each data point $x_i\,(0 \le i \le n)$ has v features. The sparse gravitational graph $G = (V, E)$ is composed of vertices $V = X$ in which the weight of each x_i is calculated by equation (3) and the set of edges $E = \{(x_i, x_j) \mid x_i, x_j \in V\}$ in which the weight of each edge (x_i, x_j) is $\|\widehat{F}_{ij}\|$ calculated by equation (2). The smaller the value of k, the sparser the graph. Figure 2 shows the different gravitational graphs of a dataset when various parameters are specified. Figures 2(a) and 2(c) show the graph when $k = 3$, while Figures 2(b) and 2(d) show when $k = 5$. Meanwhile, the weights of vertexes and edges all will be influenced by the value of γ.

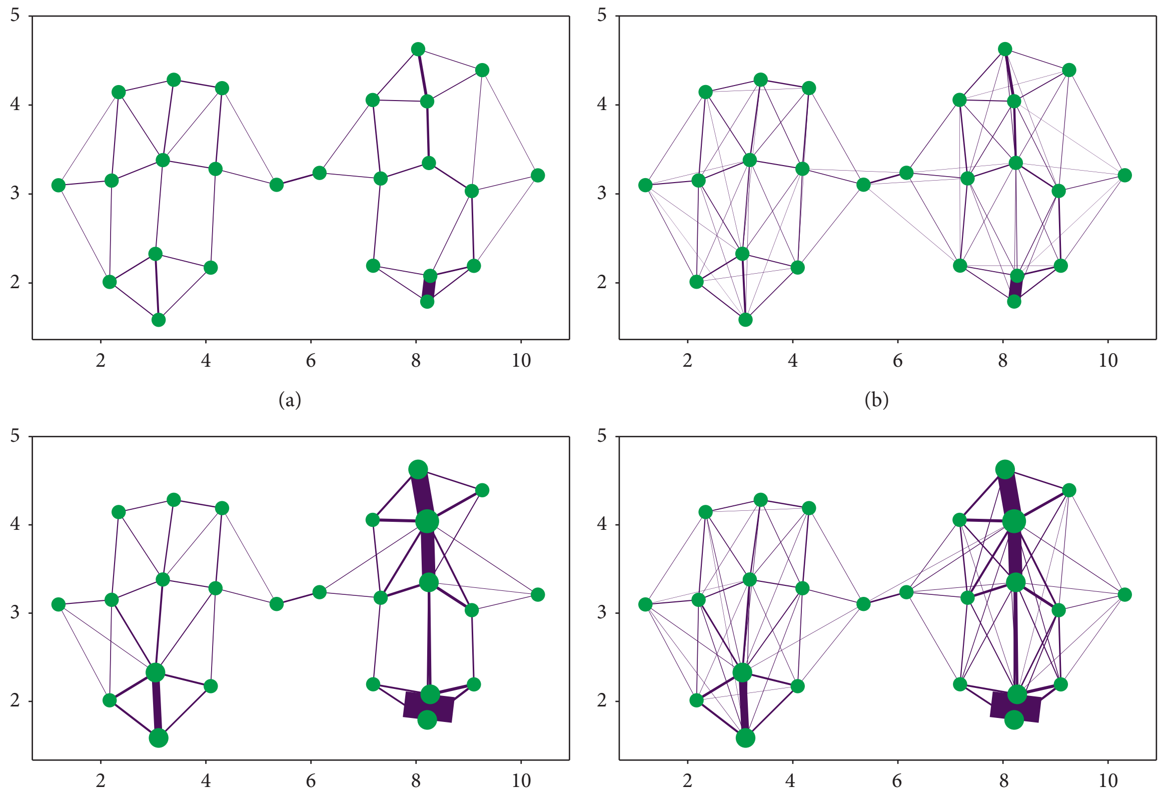
The vertex x_i is not only influenced by the vertex x_j but also by its other adjacent vertices in the sparse gravitational graph, though the relationship of two vertexes x_i and x_j can be described by the gravitational force between them simply. Considered the influence of each vertex to its adjacent vertex, the gravitational influence coefficient can also be introduced into the gravitational graph to describe the influence between two vertices. Two vertices i and j can be grouped into the same cluster if their GIC_i^j and GIC_j^i are larger than the threshold θ. The edges between two vertices in the same cluster are retained in the graph, while the edges of which the vertices belonged to different clusters are removed from the graph. Then the gravitational graph will be partition into many subgraphs. However, these subgraphs are not the final clustering results.

4.2. Gravity-Based Hierarchical Clustering Algorithm. Though the gravitational graph can be partition into many subgraphs which denote different subclusters, the performance of the clustering is poor. But these subgraphs can be considered as the intermediate results of clustering. Therefore, a new hierarchical clustering algorithm is proposed based on the intermediate results of partitioning the gravitational graph. The proposed clustering approach consists of the following three stages.

During the first stage, the data set is mapped into a sparse gravitational graph which is similar to the k-NN graph. Firstly, the data set is preprocessed by using feature transformation and dimension reduction technique. And then, the mass of each data point is calculated by equation (3), and the gravitational force between two vertices is computed by equation (2). The initial gravitational graph is constructed, in which the weights of vertex and edge are the corresponding mass and force. The procedure of constructing sparse gravitational graph is presented in Algorithm 1.

During the second phase, the gravitational graph is partitioned into many small connected subgraphs based on the gravitational influence coefficient among vertices. If the GIC_j^i and GIC_j^i are greater than the threshold θ, the edge (x_i, x_j) is retained in the graph. Otherwise, the edge would be removed from the graph. The process of the second stage is described in Algorithm 2.

In the last stage, the genuine clusters are found by emerging subgraphs iteratively. The core of merging the subgraphs is to define the linkage criterion between two clusters. The linkage criterion determines the similarity among the subgraphs. The common linkage criteria are complete linkage, single linkage, mean average linkage, centroid linkage, minimum energy linkage, graph degree linkage, and so on. Different from the above linkage criteria, a novel linkage measure is defined to determine the similarity of two subgraphs based on the vector property of gravitational forces. It is called as gravitational merging coefficient (GMC) and obtained by combining the gravitational forces between two subgraphs. Mathematically, GMC is formulated as follows:



(a)

(b)

(c)

(d)

FIGURE 2: The different sparse gravitational graph when (a) $\gamma = 0, k = 3$. (b) $\gamma = 0, k = 5$. (c) $\gamma = 0.8, k = 3$. (d) $\gamma = 0.8, k = 5$.

```
**Input:** $X$: the data set. $k$: the number of data points with top $k$ gravitational force. $\gamma$: the cutoff distance used to determine the mass of each point.
**Output:** $G$: the sparse gravitational graph.
Scale the data set $X$ using a feature transformation technique;
Calculate the Euler distance $d_{ij}$ between any two data points $i$ and $j$ in the data set $X$;
Calculate the mass $m_i$ of any data point $i$ in the data set $X$ by equation (3);
Calculate the data gravitational force $\vec{F}_{ij}$ between any two data points $i$ and $j$ in the data set $X$;
Initialize the sparse gravitational graph $G = (V, E)$. And set $V = X$ and $E = \{\}$;
**for** *each data point x in X* **do**
    Assign the mass of $x$ as the weight of the corresponding vertex in $V$;
    Select data points $y_1, y_2, \ldots, y_k$ with the top $k$ data gravitation exerted on data point $x$;
    **for** $i = 1$*to* $k$ **do**
        Insert the edges $(x, y_i)$ into the set $E$;
        Assign the data gravitational force of $x$ and $y_i$ as the weight of the edge $(x, y_i)$;
    **end**
    Calculate the gravitational resultant force of data point $x$ by equation (4) as the corresponding vertex in $V$;
**end**
return the sparse gravitational graph $G$;
```

ALGORITHM 1: CreateGravGraph (X, k, γ).

$$\mathrm{GMC}\left(C_i, C_j\right) = \frac{1}{N_i \cdot N_j} \left| \sum_{s \in C_i, t \in C_j} \vec{F}_{\mathrm{st}} \right|, \tag{6}$$

where C_i is the ist subgraph, C_j is the jst subgraph, N_i is the number of vertexes in C_i, and N_j is the number of vertexes in C_j. It each iteration, the two subgraphs with the biggest GMC are merged into a new subgraph. The clustering process is terminated until the end conditions are met. The processing steps are presented in Algorithm 3.

The overall procedure of GHC is presented in Algorithm 4. To illustrate the GHC algorithm, the main clustering steps are shown in Figure 3. Figure 3(a) shows the first stage of GHC when $\gamma = 0.8$ and $k = 3$. The artificial data

```
**Input:** $G$: the sparse gravitational graph. $\theta$: the threshold of gravitational influence coefficient.
**Output:** $G'$: the separated gravitational graph.
$G' = G$;
for each vertex $v$ in the graph $G'$ do
    Search the adjacent vertices $u_1, u_2, \ldots, u_t$ of the vertex $v$;
    for $i = 1$ to $t$ do
        Calculate the $\mathrm{GIC}^i_j$ of the vertex $u_i$ for the vertex $v$ by equation (5);
        Calculate the $\mathrm{GIC}^{u_i}_v$ of the vertex $v$ for the vertex $u_i$ by equation (5);
        if $\mathrm{GIC}^v_{u_i} < \theta$ or $\mathrm{GIC}^{u_i}_v < \theta$ then
            Remove the edge $(v, u_i)$ from the edge set $E$ of $G'$;
        end
    end
end
return $G'$;
```

Algorithm 2: PartGravGraph (G, θ).

```
Input: $G$: the original sparse gravitational graph. $G'$: the separated gravitational graph.
Output: $G''$: the merged gravitational graph.
Search out the connected subgraphs $G_1, G_2, \ldots, G_p$ in the gravitational graph $G'$;
Calculate the $\mathrm{GMC}(G_i, G_j)$ of the connected subgraphs $G_i$ and $G_j$ by equation (6);
Select the connected subgraphs $G_s$ and $G_t$ which have the largest $\mathrm{GMC}(G_s, G_t)$;
for each vertex $v$ in $G_s$ do
    for each vertex $u$ in $G_s$ do
        if the edge $(v, u)$ in $G$ then
            Insert the edge $(v, u)$ into $G'$;
        end
    end
end
return $G'' = G'$;
```

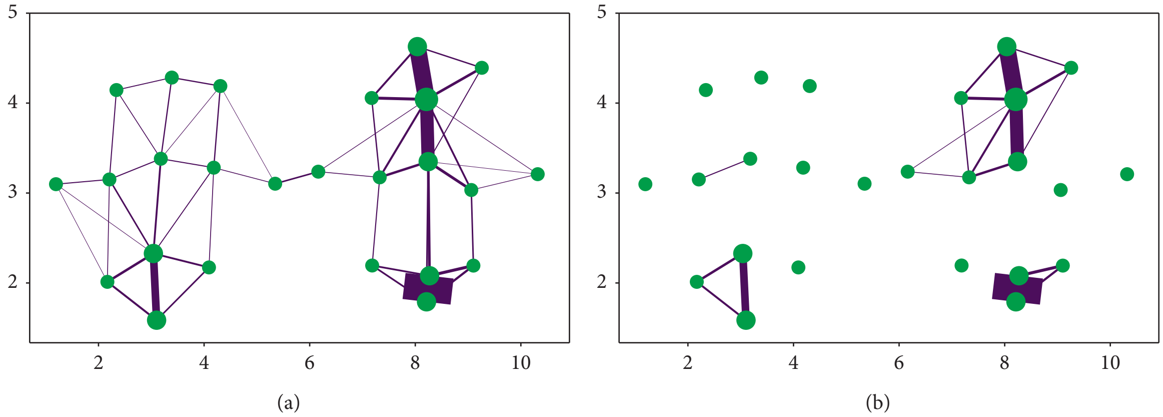
Algorithm 3: MergeGravGraph (G, G').

```
Input: $X$: the data set. $\gamma$: the cutoff distance. $k$: the number of data points with top $k$ gravitational force. $\theta$: the threshold of
gravitational influence coefficient. numClasses: the number of the final clusters.
Output: labels: the.
$G$ = CreateGravGraph ($X$, $k$, $\gamma$);
$G'$ = PartGravGraph ($G$, $\theta$);
Calculate the number numClusters of the connected subgraphs in the graph $G'$;
while numClusters > numClasses do
    $G'$ = MergeGravGraph ($G$, $G'$);
    Calculate the number numClusters of the connected subgraphs in the graph $G'$;
end
Assign cluster label for each subgraph in $G'$;
return the cluster labels labels;
```

Algorithm 4: GHC algorithm.

set with 25 data points is mapped into a gravitation graph by Algorithm 1. Figure 3(b) shows the second stage of GHC when $\theta = 0.5$. The gravitational graph is partitioned into many subgraphs by Algorithm 2. Figures 3(c) and 3(d) show the third stage of GHC. The subgraphs with highest GMC is merged by using Algorithm 3. Figure 3(c) shows the gravitational graph after six iterations. Figure 3(d) shows the clustering result after twelve iterations. Obviously, the data set is grouped into two clusters correctly.

4.3. Complexity Analysis. The time complexity can be defined as the sum of the complexities of each stage of GHC algorithm. For the first stage, each data point needs to calculate the masses, find its neighbors with the top k gravitational forces, and then construct the gravitational graph. Considering a data set with n data points, the time complexity of Algorithm 1 is $O(n^2 \log(n))$. During the second stage, the GRF of each data point is calculated, and the gravitation graph is divided into some subgraphs. Thus,



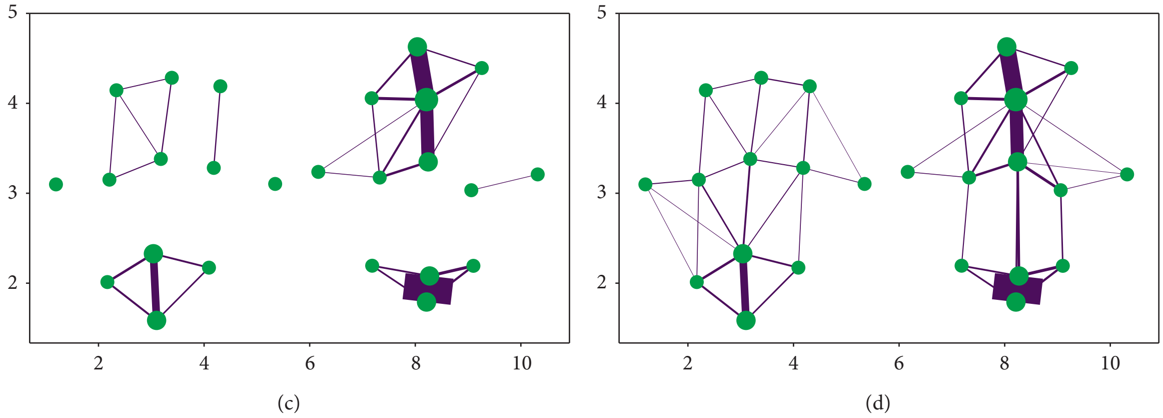


Figure 3: Clustering stages in the 2D dataset. (a) The gravitational graph mapped from the dataset with 26 elements in 2D space. The size of vertex denotes the mass of the mapped data points. The width of edge denotes the magnitude of gravity force. (b) The gravitational graph partitioned into lots of subgraphs by Algorithm 2. (c) The graph by merging the subgraphs six times using Algorithm 3. (d) The graph by merging the subgraphs twelve times using Algorithm 3 (the bigger the vertex is, the bigger the mass is; the wider the edge, the bigger the gravitational force between two vertexes).

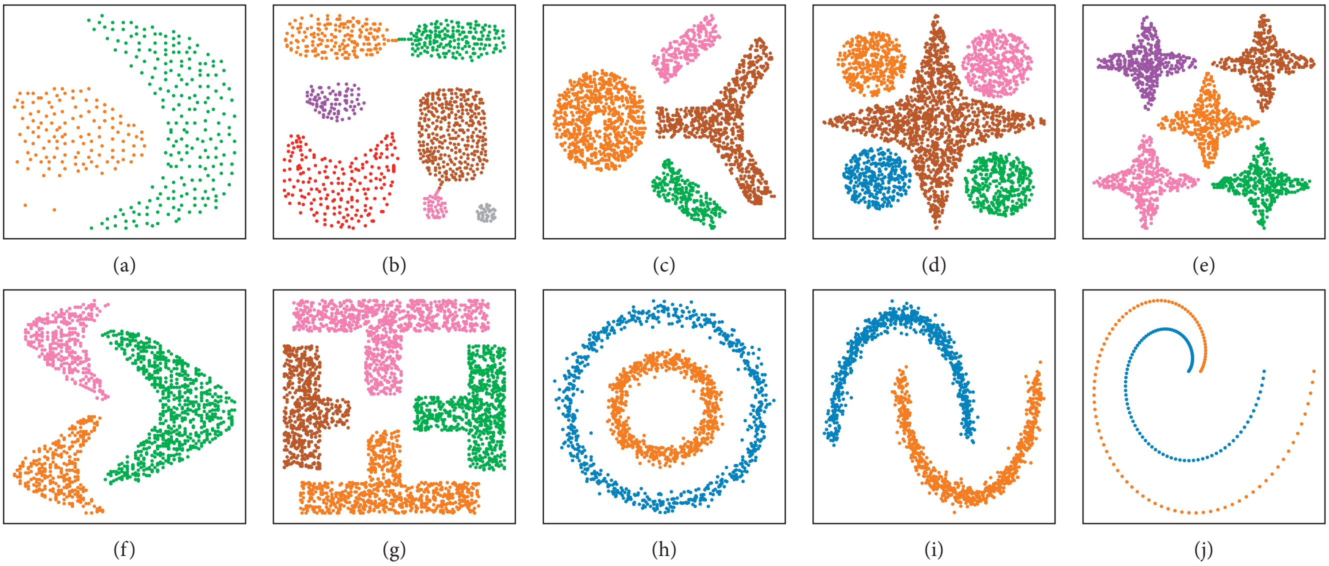


Figure 4: The flowchart of the power-driven grey model.

the time complexity of Algorithm 2 is $O(\kappa n)$ where κ is the mean number of adjacent vertexes and n is the number of vertexes in gravitational graph. Usually, κ is much smaller than n. During the third stage, it is required to calculate GMC of any two subgraphs and merge two subgraphs which have the largest GMC. The worst case time complexity of

Algorithm 3 is $O(kn^2)$. Therefore, the worst case time complexity of GHC algorithm is $O(n^2 \log(n))$.

5. Experiments

5.1. Performance Metrics. In this study, four clustering performance metrics, such as Purity [27], Rand Index (RI) [35], *F*measure [35], and Normalized Mutual Information (NMI) [36], are used to evaluate the performance of clustering algorithms. Given a dataset $X = \{x_1, x_2, \ldots, x_n\}$ with p categories and n data points, the set $P = \{p_1, p_2, \ldots, p_p\}$ denotes the real classes in which $P_j\,(1 \le j \le p)$ is the subset of X. The clustering result is the set $Q = \{Q_1, Q_2, \ldots, Q_q\}$ in which $Q_i\,(1 \le i \le q)$ also is the subset of X. Purity is the external evaluation criterion of cluster quality. The purity of a cluster Q_i with n_i data points is defined as follows:

$$P(Q_i) = \frac{1}{n_i} \max_j \left(n_i^j\right), \tag{7}$$

where n_i^j is the number of the data points in jth class that are assigned to ith cluster. The overall purity of a clustering result is defined as

$$\text{Purity} = \sum_{i=1}^{q} \frac{n_i}{n} P(Q_i). \tag{8}$$

In general, larger Purity denotes better clustering result. Rand Index is calculated as follows:

$$\text{RI} = \frac{a+b}{n^2 - n}, \tag{9}$$

where $\text{RI} \in [0, 1]$, a is the number of pairs of data items in X that are in the same subset of Q and in the same subset of P, b is the number of pairs of data items in X that are in different subsets of Q and in different subsets of P. *F*measure is like RI with the exception that true negatives are not taken into account. Mathematically, *F*measure is calculated as follows:

$$F\text{measure} = \frac{2a+b}{2a+b+c}, \tag{10}$$

where c is the number of pairs of data items in X that are in different subsets of Q and in the same subset of P. The normalized mutual information (NMI) is also adopted in this paper. The NMI is computed as

$$\begin{aligned} \text{NMI}(P,Q) &= \frac{2H(P,Q)}{H_0(P) + H_0(Q)}, \\ H(P,Q) &= -\sum_{i=1}^{p}\sum_{j=1}^{q} \frac{\left|P_i \cap Q_j\right|}{n} \log \frac{n\left|P_i \cap Q_j\right|}{\left|P_i\right|\left|Q_j\right|}, \\ H_0(P) &= -\sum_{i=1}^{p} \frac{\left|P_i\right|}{n} \log \frac{\left|P_i\right|}{n}, \\ H_0(Q) &= -\sum_{j=1}^{q} \frac{\left|Q_j\right|}{n} \log \frac{\left|Q_j\right|}{n}. \end{aligned} \tag{11}$$

The larger NMI denotes a better performance of clustering.

Table 1: Characteristics of the real-world datasets obtained from UCI repository.

Dataset	Number of instances	Number of attributes	Number of classes
BTissue	103	9	6
Ecoli	336	8	8
Iris	150	4	3
Wine	178	13	3
PID	768	8	2
BSTC	748	4	2
Glass	214	9	6
Cancer	699	9	2
SControl	600	60	6

5.2. Parameter Settings. To investigate the performance of GHC, the experiments are performed on the synthetic datasets shown in Figure 4 and real-life datasets tabulated in Table 1. Six well-known clustering algorithms, such as *K*-means [10], *K*-means++ [37], Spectral Clustering (SC) [38], DBSCAN [13], Birch [39], and LGC [25], are employed to compare with GHC algorithm.

The well-tuned parameter settings of the GHC algorithm and the competitive algorithms are tabulated for each data set in Table 2. For *K*-means, *K*-means++, and SC, the parameter τ is the number of classes in each data set. For SC algorithm, the parameter σ^2 is sought from the set {0.01, 0.1, 0.5, 1, 1.5, 2, 5, 10, 20, 70, 80, 100, 150} and the one with the best RI value is selected. There are two parameters eps (e) and min-samples (m) of DBSCAN. The parameter e is chosen from the interval from 0.1 to 1 with an increment of 0.1. The parameter m is chosen from the set {4, 5, 6, 8, 10, 15, 20}. The parameters pairing (e, m) with the best RI value are considered as the well-tuned parameters. Birth has two tunable parameters threshold (t) and branching-factor (b) [39]. The parameter t is varied from 0.1 to 1 with an increment of 0.1. The parameter b is chosen from the set {10, 20, 40, 50, 60, 100, 150, 200}. The parameter pairing (t, b) with the best RI value is selected for each data set. For the three parameters IM (i), kn (n), and cFactor (f) of LGC algorithm, the value of i is chosen from the set {5, 8, 10, 20, 30, 40, 50, 60, 70, 80, 150, 250}, the value of n is chosen from the set {4, 5, 7, 10, 15, 20}, and the value of f is varied from 0 to 1 with increment 0.1. The parameters pair (m, n, f) with the best RI value is chosen for each dataset.

To demonstrate the performance of GHC algorithm, the three parameter γ, θ, and k are, respectively, equal to 0.2, 0.1, and 6 for all synthetic data sets. For all the real-world data sets, the pair (γ, θ, k) with the best RI value is chosen for each real-world dataset. The tunable parameter γ is varied from −1 to 10 with an increment 0.1. The parameter θ is chosen from −1 to 1 with an interval 0.1. The parameter k is chosen from the set {4, 5, 6, 10}. For all nondeterministic approaches, we run these algorithms 100 times on each data set and adopt the average of each performance criterion to evaluate the performance of GHC algorithm. For all deterministic approaches, the performance metrics are taken by running only once.

Table 2: Well-tuned parameter configurations for different synthetic and real-world datasets.

Dataset	*K*-means	*K*-means++	SC		DBSCAN		Birth		LGC			GHC		
	τ	τ	τ	σ^2	m	e	t	b	i	n	f	γ	θ	k
Figure 4(a)	2	2	2	10	5	0.3	0.1	10	40	7	0.3	0.2	0.1	6
Figure 4(b)	7	7	7	20	20	0.2	0.5	10	30	10	0.3	0.2	0.1	6
Figure 4(c)	4	4	4	10	10	0.2	0.5	20	20	5	0.1	0.2	0.1	6
Figure 4(d)	5	5	5	70	4	0.1	0.4	40	20	10	0.1	0.2	0.1	6
Figure 4(e)	5	5	5	10	4	0.1	0.8	10	30	4	0.7	0.2	0.1	6
Figure 4(f)	3	3	3	20	10	0.2	0.1	10	20	10	0.1	0.2	0.1	6
Figure 4(g)	4	4	4	10	10	0.2	0.1	20	20	4	0.1	0.2	0.1	6
Figure 4(h)	2	2	2	10	10	0.3	0.1	100	20	4	0.1	0.2	0.1	6
Figure 4(i)	2	2	2	10	10	0.2	0.1	40	20	4	0.1	0.2	0.1	6
Figure 4(j)	2	2	2	80	5	0.2	0.4	10	5	4	0.3	0.2	0.1	6
BTissue	6	6	6	5	4	0.7	0.2	10	5	10	0.3	0.3	0.1	4
Ecoli	8	8	8	0.01	8	0.9	0.7	20	8	4	0.5	0.2	0.5	4
Iris	3	3	3	0.5	4	0.7	0.5	20	5	7	0.9	0.4	0	4
Wine	3	3	3	0.01	5	0.5	1	40	5	5	0.3	0.1	0.1	4
PID	2	2	2	0.5	10	0.2	0.2	20	20	4	0.9	0.1	0.1	6
BSTC	2	2	2	150	6	0.8	0.8	10	70	4	0.9	0.1	0.1	5
Glass	6	6	6	0.05	4	0.9	0.6	20	20	7	0.3	0	0.3	5
Cancer	2	2	2	0.01	4	0.9	0.7	50	30	4	0.7	0	0	6
SControl	6	6	6	150	10	10	0.5	30	5	4	0.9	0	0.3	5

Table 3: Performance comparison on the synthetic datasets.

Dataset	*K*-means	*K*-means++	SC	DBSCAN	Birch	LGC	GHC
Figure 4(a)	0.8	0.8	0.99	0.99	**1**	**1**	**1**
	0.89	0.89	0.99	**1**	**1**	**1**	**1**
	0.81	0.81	0.99	0.99	**1**	**1**	**1**
	0.55	0.55	0.95	0.98	**1**	**1**	**1**
Figure 4(b)	0.92	0.92	**1**	0.99	0.97	0.99	**1**
	0.93	0.93	**1**	0.99	0.96	0.99	**1**
	0.79	0.79	**0.99**	0.97	0.92	0.97	**0.99**
	0.86	0.86	**0.99**	0.96	0.92	0.97	**0.99**
Figure 4(c)	0.89	0.89	**1**	**1**	0.91	**1**	**1**
	0.85	0.85	**1**	**1**	0.9	**1**	**1**
	0.81	0.81	**1**	**1**	0.85	**1**	**1**
	0.76	0.76	**1**	**1**	0.81	**1**	**1**
Figure 4(d)	0.77	0.76	0.99	0.99	0.84	**1**	**1**
	0.74	0.73	0.99	**1**	0.83	**1**	**1**
	0.53	0.53	0.99	0.99	0.69	**1**	**1**
	0.61	0.6	0.98	0.97	0.76	**1**	**1**
Figure 4(e)	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	1	**1**	**1**	**1**	**1**	**1**	**1**
	1	**1**	**1**	**1**	**1**	**1**	**1**
	1	**1**	**1**	**1**	**1**	**1**	**1**
Figure 4(f)	0.86	0.86	0.99	**1**	**1**	**1**	**1**
	0.9	0.9	0.99	**1**	**1**	**1**	**1**
	0.81	0.81	0.99	**1**	**1**	**1**	**1**
	0.72	0.73	0.99	**1**	**1**	**1**	**1**
Figure 4(g)	0.96	0.96	**1**	**1**	**1**	**1**	**1**
	0.96	0.96	**1**	**1**	**1**	**1**	**1**
	0.92	0.92	**1**	**1**	**1**	**1**	**1**
	0.9	0.9	**1**	**1**	**1**	**1**	**1**
Figure 4(h)	0.5	0.5	**1**	**1**	0.51	**1**	**1**
	0.5	0.5	**1**	**1**	0.58	**1**	**1**
	0.5	0.5	**1**	**1**	0.54	**1**	**1**
	0	0	**1**	**1**	0.02	**1**	**1**

Table 3: Continued.

Dataset	*K*-means	*K*-means++	SC	DBSCAN	Birch	LGC	GHC
Figure 4(i)	0.75	0.75	**1**	**1**	0.86	**1**	**1**
	0.85	0.85	**1**	**1**	0.93	**1**	**1**
	0.75	0.75	**1**	**1**	0.86	**1**	**1**
	0.39	0.39	**1**	**1**	0.69	**1**	**1**
Figure 4(j)	0.5	0.5	0.77	0.59	0.5	0.53	**1**
	0.52	0.52	0.87	0.72	0.53	0.64	**1**
	0.52	0.52	0.79	0.63	0.56	0.48	**1**
	0	0	0.52	0.21	0	0.13	**1**

For each dataset, the numbers in the first row are RI, those in the second row are Purity, those in the third row are *F*measure, and those in the forth row are NMI.

Table 4: Performances of different clustering approaches on UCI datasets tested.

Dataset	*K*-means	*K*-means++	SC	DBSCAN	Birch	LGC	GHC
BTissue	0.71	0.6	0.65	0.66	0.7	0.73	**0.78**
	0.43	0.39	0.34	0.42	0.52	0.5	**0.57**
	0.36	0.31	0.25	0.44	0.44	0.39	**0.47**
	0.36	0.29	0.14	0.46	0.53	0.43	**0.54**
Ecoli	0.81	0.81	0.86	0.77	0.9	0.88	**0.91**
	0.82	0.83	0.79	0.62	0.8	0.82	**0.86**
	0.54	0.53	0.71	0.64	**0.83**	0.79	0.82
	0.61	0.61	0.6	0.47	**0.73**	0.64	0.7
Iris	0.88	0.88	0.88	0.77	0.82	0.89	**0.97**
	0.89	0.89	0.89	0.68	0.86	0.93	**0.97**
	0.82	0.82	0.82	0.73	0.72	0.82	**0.95**
	0.76	0.76	0.75	0.66	0.67	0.75	**0.9**
Wine	0.72	0.72	0.56	0.34	0.93	0.62	**0.95**
	0.7	0.7	0.53	0.4	0.96	0.66	**0.97**
	0.58	0.58	0.45	0.51	0.9	0.58	**0.93**
	0.43	0.43	0.12	0	0.82	0.42	**0.88**
PID	0.55	0.55	0.54	0.55	0.57	0.56	**0.6**
	0.66	0.66	0.65	0.65	0.7	0.68	**0.73**
	0.63	0.63	**0.7**	0.71	0.49	0.58	0.66
	0.03	0.03	0	0	0.08	0.05	**0.11**
BSTC	0.6	0.6	0.63	0.63	0.61	**0.64**	**0.64**
	0.76	0.76	0.76	0.76	**0.77**	0.76	**0.77**
	0.71	0.71	0.77	0.76	0.7	**0.78**	**0.78**
	0.02	0.02	0	0.01	**0.05**	0	0.02
Glass	0.68	0.68	0.71	0.63	0.71	0.58	**0.72**
	0.59	0.59	0.59	0.49	0.6	**0.68**	0.64
	0.5	**0.5**	0.41	0.43	0.48	0.49	0.45
	0.42	**0.42**	0.37	0.32	0.41	0.34	0.4
Cancer	0.92	0.92	0.93	0.89	0.87	**0.95**	0.94
	0.96	0.96	0.96	0.94	0.96	**0.97**	**0.97**
	0.93	0.93	0.94	0.9	0.87	**0.95**	0.94
	0.74	0.73	0.77	0.72	0.6	**0.82**	0.79
SControl	0.84	0.85	0.7	0.17	0.87	**0.89**	**0.89**
	0.64	0.64	0.23	0.17	0.67	**0.93**	0.67
	0.61	0.63	0.19	0.28	0.71	0.52	**0.75**
	0.69	0.71	0.01	0	0.82	0.68	**0.85**

For each dataset, the numbers in the first row are rand index, those in the second row are purity, those in the third row are *F*measure, and those in the forth row are NMI.

5.3. Experiments on Synthetic Datasets. In order to investigate the performance of the proposed approaches, a series of experiments on twelve synthetic datasets shown in Figure 3 are performed by using the proposed GHC and the other existing algorithms. The performance results are tabulated in Table 3. In Table 3, the first column denotes the used dataset, whereas the first row denotes the used algorithms. The digits in the other fields of the table are the evaluation results. For Purity, RI, *F*measure, and NMI, their values range from 0 to 1. The higher values denote the algorithm has a better

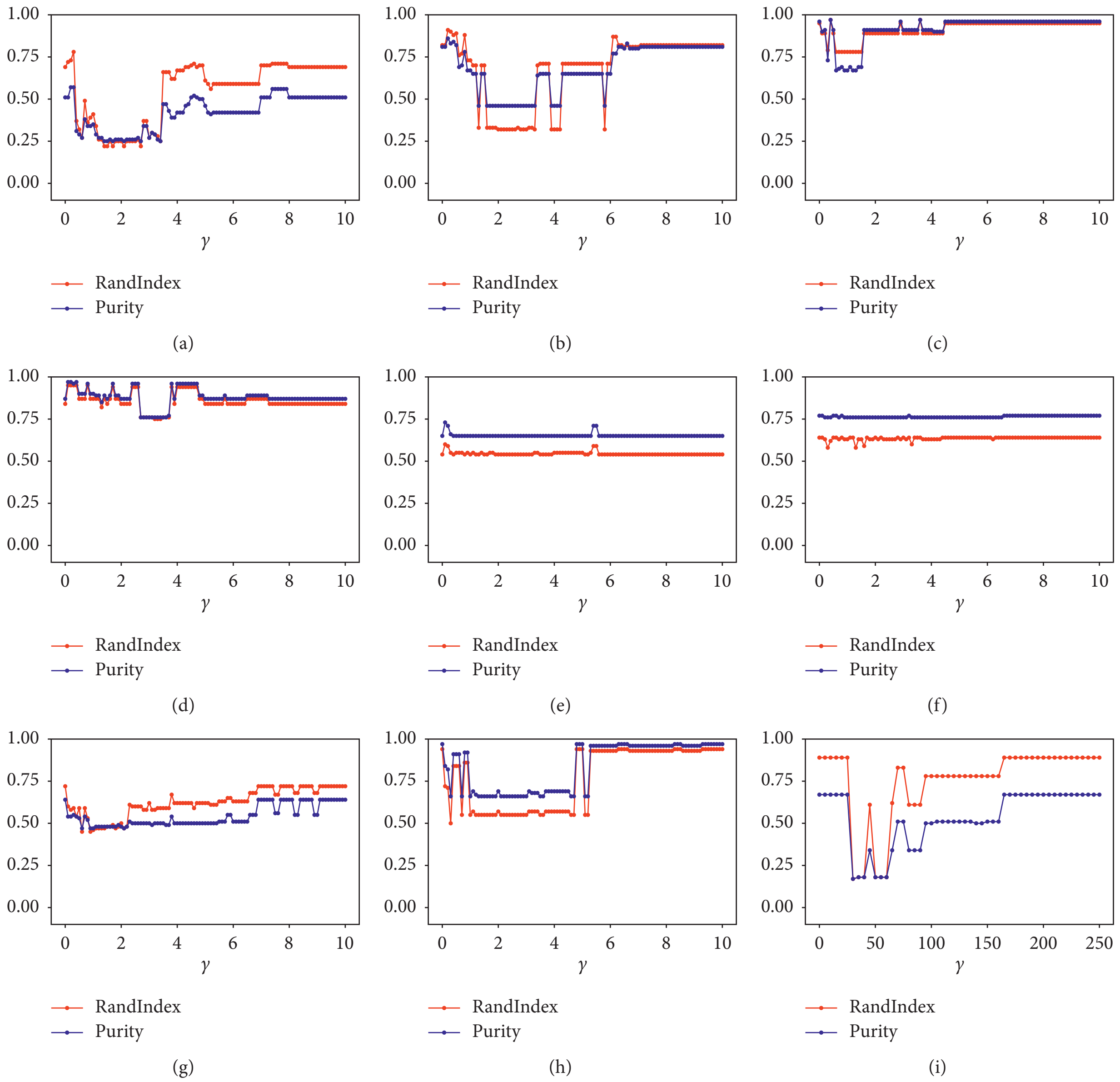
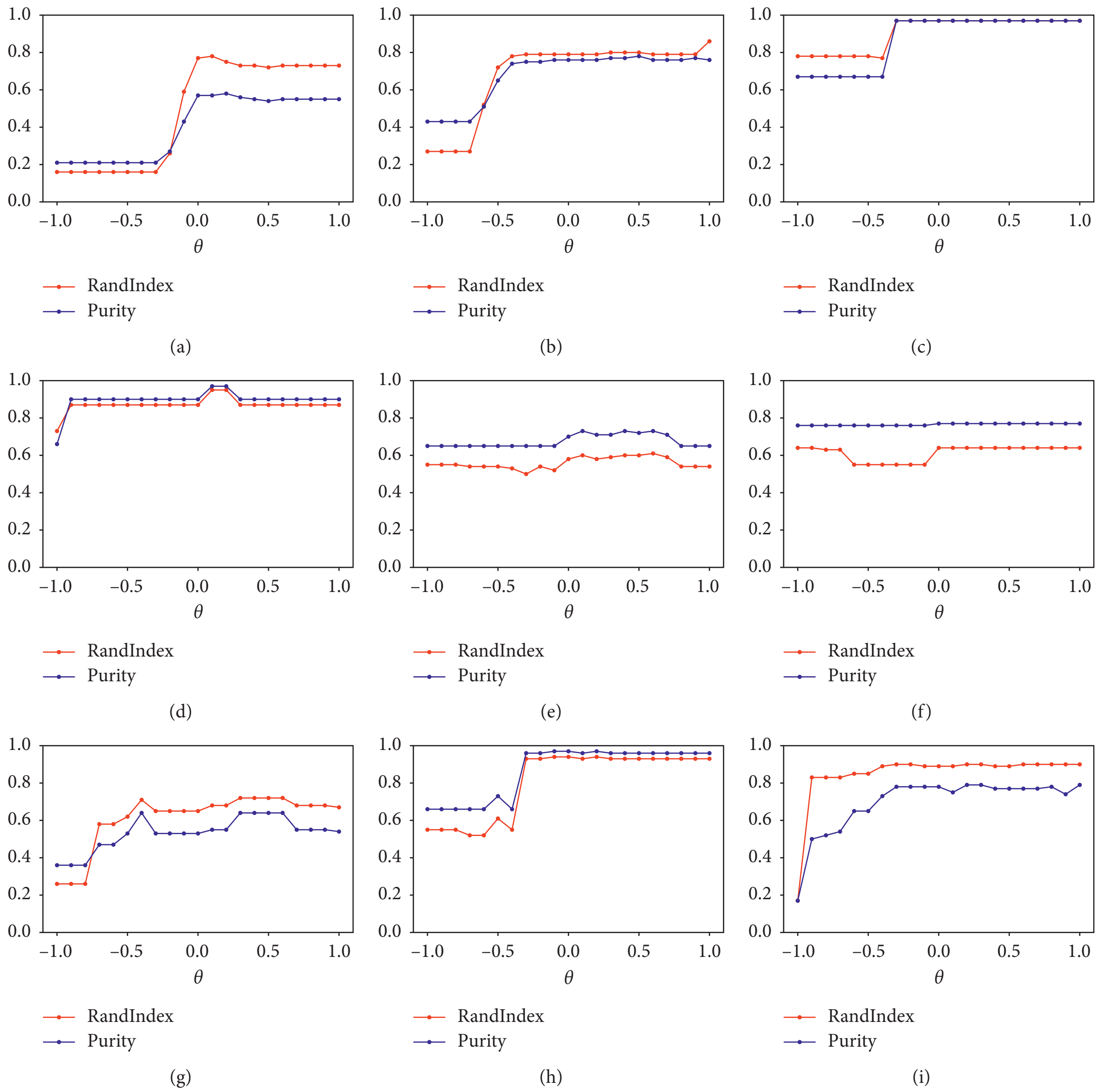


Figure 5: Parameter analysis for the threshold parameter γ. (a) BTissue. (b) Ecoil. (c) Iris. (d) Wine. (e) PID. (f) BTSC. (g) Glass. (h) Cancer. (i) SControl.

performance on the dataset. Although the synthetic datasets are easy to be clustered intuitively, not all the clustering algorithms achieve remarkable performance in this study. Overall, CHC, DBSCAN, and LGC algorithms obtain more competitive advantages than the other algorithms. The GHC algorithm obtains good clustering results on all synthetic datasets while DBSCAN and LGC algorithms only achieve the worse performances on a few datasets.

5.4. Experiments on Real-Life Datasets. In order to investigate the performance, the proposed GHC algorithm and other competitive approaches are adopted to solve the clustering problems on the real-world datasets tabulated in 3. For each real-life dataset, the well-tuned parameters of all algorithms also are tabulated in Table 2. The performance results are shown in Table 4. In Table 4, the first row denotes the algorithms while the first column denotes the real-life datasets used in the experiments. The digits in other fields of this table denote the evaluation results for the GHC algorithm and other existing algorithms on each dataset. GHC algorithm obtains the best values of all evaluation criteria on the datasets such as BTissue, Iris, and Wine. On the other datasets, the evaluation results of the GHC algorithm are the best or close to the best for the four evaluation criteria. In the overall view, the GHC algorithm outperforms other competitive algorithms on these real-world datasets.

5.5. Discussions. In this subsection, we mainly discuss the role and impact of parameters to the performance of the GHC algorithm. There are three tunable parameters γ, θ, and



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 6: Parameter analysis for the threshold parameter θ. (a) BTissue. (b) Ecoil. (c) Iris. (d) Wine. (e) PID. (f) BTSC. (g) Glass. (h) Cancer. (i) SControl.

k, which are required to determine for GHC. The parameter γ determines the masses of data points, which affects the force of gravity straightly and controls the structure of the gravitational graph with the gravity forces varying. The second parameter θ controls the number of subgraphs that the gravitational graph can be partitioned into. The third parameter k determines the sparsity and connectivity of the gravitational graph. In the previous subsection, it can be noticed that the GHC algorithm performs better than the state-of-the-art algorithms for all synthetic datasets, though these parameters are set to fixed values $(\gamma = 0.2, \theta = 0.1, k = 6)$ which maybe are not the optimal values.

To illustrate the impact of the parameters γ, θ, and k, we conduct a series of experiments on real-world datasets to analyze the influence of each parameter to the clustering performance of the GHC algorithm. The prior knowledge of the real-world datasets can be used to search the optimal parameters with the best values of evaluation metrics. Figure 5 shows that the values of the evaluation metrics Rand Index and Purity change when the parameter γ varies in a given interval. The parameter γ varies from 0 to 10 with increment of 0.1 for all datasets except the SControl dataset for which it varies from 0 to 250 with increment of 5. It can be noticed that the performance values fluctuate within an interval as the parameter γ is increased. The evaluation result on each dataset will converge to a fixed value when γ is beyond the interval. Because γ determines the mass of each data point by equation (3), the gravitational force between two data points will be significantly different when γ is set to different values. In essence, the different distribution of data points' masses affects the gravitational forces between them

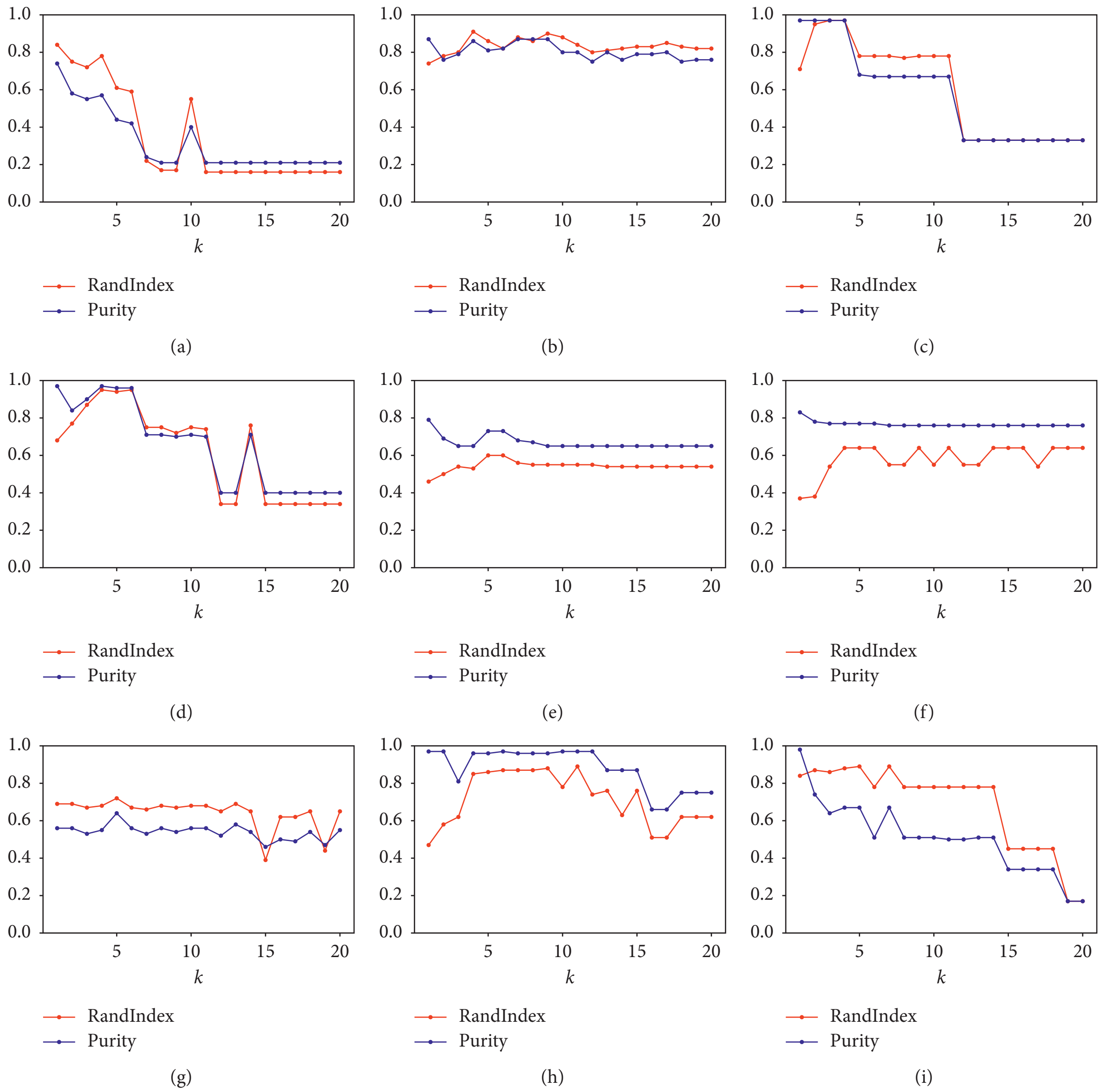


Figure 7: Parameter analysis for the threshold parameter k. (a) BTissue. (b) Ecoil. (c) Iris. (d) Wine. (e) PID. (f) BTSC. (g) Glass. (h) Cancer. (i) SControl.

and leads to different clustering performance. Figure 6 shows that the values of the evaluation metrics Rand Index and Purity with the best clustering performance change as the parameter θ specifies different values, which changes from −1 to 1 with increment 0.1. From Figure 6, it can be noticed that the evaluation values increase on the general trend as the threshold θ is increasing in most of the real-world datasets. The reason is that the data points of different clusters are divided into the same cluster in the second stage of the GHC algorithm when the parameter θ is set to a lower value. In contrast, the performance of the GHC algorithm is better when the value of parameter θ is set to a high value because the data points of different clusters can be partitioned into a cluster correctly. Figure 7 shows that the values of the evaluation metrics Rand Index and Purity with the best clustering performance change as the parameter k specifies different values. In general, the performance metric Purity decreases as the parameter k is increasing. From Figure 7, there is a single peak at which the value of Rand Index reaches the maximum when k changes from 1 to 20. From the above analysis, the GHC algorithm can achieve good performance when the three parameters are set to the suitable values for each dataset.

6. Conclusions

In this paper, we propose a novel gravity-based clustering approach that sufficiently utilizes the vector properties of gravitational force. To some extent, the data gravitational force can be considered as a similarity measure which takes not only density but also distance into account. To illustrate the performance of GHC algorithm, the experiments with all well-tuned parameters have been conducted on synthetic datasets and real-life datasets compared with the other famous clustering algorithms. The experiments' results show that the GHC algorithm is robust and achieves competitive performance. Of course, it also can be noticed that the time complexity of the GHC algorithm is high. The problem can be improved in the future. Meanwhile, the GHC algorithm can be applied in more application fields.

Data Availability

The data used to support the findings of this study have been deposited in the UCI Machine Learning Repository (http://archive.ics.uci.edu/ml) and the figshare database (https://doi.org/10.6084/m9.figshare.8187623.v1).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 61672136 and 11872323).

References

[1] X. Zhang, H. Gao, G. Li et al., "Multi-view clustering based on graph-regularized nonnegative matrix factorization for object recognition," *Information Sciences*, vol. 432, pp. 463–478, 2018.

[2] H.-J. Xing and M.-H. Ha, "Further improvements in feature-weighted fuzzy C-means," *Information Sciences*, vol. 267, pp. 1–15, 2014.

[3] S. K. Choy, S. Y. Lam, K. W. Yu, W. Y. Lee, and K. T. Leung, "Fuzzy model-based clustering and its application in image segmentation," *Pattern Recognition*, vol. 68, pp. 141–157, 2017.

[4] Y. Lei, Y. Zhou, and J. Shi, "Overlapping communities detection of social network based on hybrid C-means clustering algorithm," *Sustainable Cities and Society*, vol. 47, Article ID 101436, 2019.

[5] C. He, Y. Tang, H. Liu, X. Fei, H. Li, and S. Liu, "A robust multi-view clustering method for community detection combining link and content information," *Physica A: Statistical Mechanics and its Applications*, vol. 514, pp. 396–411, 2019.

[6] H. Chen, Y. Zhang, and I. Gutman, "A kernel-based clustering method for gene selection with gene expression data," *Journal of Biomedical Informatics*, vol. 62, pp. 12–20, 2016.

[7] R. Jothi, S. K. Mohanty, and A. Ojha, "Functional grouping of similar genes using eigenanalysis on minimum spanning tree based neighborhood graph," *Computers in Biology and Medicine*, vol. 71, pp. 135–148, 2016.

[8] R. Janani and S. Vijayarani, "Text document clustering using spectral clustering algorithm with particle swarm optimization," *Expert Systems with Applications*, vol. 134, pp. 192–200, 2019.

[9] C. Luo, Y. Li, and S. M. Chung, "Text document clustering based on neighbors," *Data & Knowledge Engineering*, vol. 68, no. 11, pp. 1271–1288, 2009.

[10] A. K. Jain, "Data clustering: 50 years beyond *K*-means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651–666, 2010.

[11] H.-S. Park and C.-H. Jun, "A simple and fast algorithm for K-medoids clustering," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3336–3341, 2009.

[12] C. Jinyin, H. Huihao, C. Jungan, Y. Shanqing, and S. Zhaoxia, "Fast density clustering algorithm for numerical data and categorical data," *Mathematical Problems in Engineering*, vol. 2017, Article ID 6393652, 15 pages, 2017.

[13] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery & Data Mining (KDD-96)*, Portland, OR, USA, August 1996.

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

[15] B. Lorbeer, A. Kosareva, B. Deva, D. Softić, P. Ruppel, and A. Küpper, "Variations on the clustering algorithm BIRCH," *Big Data Research*, vol. 11, pp. 44–53, 2018.

[16] G. Karypis, E.-H. Han, and V. Kumar, "Chameleon: hierarchical clustering using dynamic modeling," *Computer*, vol. 32, no. 8, pp. 68–75, 1999.

[17] J. Handl and J. Knowles, "An evolutionary approach to multiobjective clustering," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 1, pp. 56–76, 2007.

[18] H. Peng, P. Shi, J. Wang, A. Riscos-Núñez, and M. J. Pérez-Jiménez, "Multiobjective fuzzy clustering approach based on tissue-like membrane systems," *Knowledge-Based Systems*, vol. 125, pp. 74–82, 2017.

[19] I. Saha and U. Maulik, "Incremental learning based multiobjective fuzzy clustering for categorical data," *Information Sciences*, vol. 267, pp. 35–57, 2014.

[20] J. Xu, G. Wang, and W. Deng, "DenPEHC: density peak based efficient hierarchical clustering," *Information Sciences*, vol. 373, pp. 200–218, 2016.

[21] L. Aliahmadipour and E. Eslami, "GHFHC: generalized hesitant fuzzy hierarchical clustering algorithm," *International Journal of Intelligent Systems*, vol. 31, no. 9, pp. 855–871, 2016.

[22] F. Ros and S. Guillaume, "Munec: a mutual neighbor-based clustering algorithm," *Information Sciences*, vol. 486, pp. 148–170, 2019.

[23] W. E. Wright, "Gravitational clustering," *Pattern Recognition*, vol. 9, no. 3, pp. 151–166, 1977.

[24] H. C. Yung, "Segmentation of color images based on the gravitational clustering concept," *Optical Engineering*, vol. 37, no. 3, pp. 989–1000, 1998.

[25] Z. Wang, Z. Yu, C. L. P. Chen et al., "Clustering by local gravitation," *IEEE Transactions on Cybernetics*, vol. 48, no. 5, pp. 1383–1396, 2018.

[26] A. Bahrololoum, H. Nezamabadi-Pour, and S. Saryazdi, "A data clustering approach based on universal gravity rule," *Engineering Applications of Artificial Intelligence*, vol. 45, pp. 415–428, 2015.

[27] M. Alswaitti, M. K. Ishak, and N. A. M. Isa, "Optimized gravitational-based data clustering algorithm," *Engineering Applications of Artificial Intelligence*, vol. 73, pp. 126–148, 2018.

[28] J. Gomez, O. Nasraoui, and E. Leon, "RAIN: data clustering using randomized interactions between data points," in *Proceedings of the 2004 International Conference on Machine Learning and Applications, 2004. Proceedings*, pp. 250–255, IEEE, Louisville, KY, USA, 2004.

[29] S. Kundu, "Gravitational clustering: a new approach based on the spatial distribution of the points," *Pattern Recognition*, vol. 32, no. 7, pp. 1149–1160, 1999.

[30] M. A. Sanchez, O. Castillo, J. R. Castro, and P. Melin, "Fuzzy granular gravitational clustering algorithm for multivariate data," *Information Sciences*, vol. 279, pp. 498–511, 2014.

[31] A. Hatamlou, "Black hole: a new heuristic optimization approach for data clustering," *Information Sciences*, vol. 222, pp. 175–184, 2013.

[32] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, and G. S. A., "GSA: a gravitational search algorithm," *Information Sciences*, vol. 179, no. 13, pp. 2232–2248, 2009.

[33] A. Shukla and S. N. Singh, "Cluster based wind-hydrothermal unit commitment using GSA algorithm," in *Proceedings of the 2014 IEEE PES General Meeting—Conference Exposition*, pp. 1–5, National Harbor, MD, USA, July 2014.

[34] M. Yin, Y. Hu, F. Yang, X. Li, and W. Gu, "A novel hybrid K-harmonic means and gravitational search algorithm approach for clustering," *Expert Systems with Applications*, vol. 38, no. 8, pp. 9319–9324, 2011.

[35] K.-C. Wong, C. Peng, Y. Li, and T.-M. Chan, "Herd Clustering: a synergistic data clustering approach using collective intelligence," *Applied Soft Computing*, vol. 23, pp. 61–75, 2014.

[36] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison," in *Proceedings of the 26th Annual International Conference on Machine Learning-ICML '09*, pp. 1073–1080, 2009.

[37] D. Arthur and S. Vassilvitskii, "*K*-means++: the advantages of careful seeding," in *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA '07*, pp. 1027–1035, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2007.

[38] D. R. DeFord and S. D. Pauls, "Spectral clustering methods for multiplex networks," *Physica A: Statistical Mechanics and Its Applications*, vol. 533, p. 121949, 2019.

[39] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: an efficient data clustering method for very large databases," *SIGMOD Record (ACM Special Interest Group on Management of Data)*, vol. 25, no. 2, pp. 103–114, 1996.