SN Computer Science (2022) 3:311
https://doi.org/10.1007/s42979-022-01208-8

ORIGINAL RESEARCH

A New Particle Swarm Optimization Algorithm for Optimizing Big Data Clustering

Seyed Emadedin Hashemi[1] · **Madjid Tavana**[2,3] · **Maryam Bakhshi**[1]

Received: 31 July 2021 / Accepted: 5 April 2022 / Published online: 30 May 2022

Abstract

Clustering is an ideal tool for working with big data and searching for structures in the data set. Clustering aims at maximizing the similarity between the data within a cluster and minimizing the similarity between the data between different clusters. This study presents a new and improved Particle Swarm Optimization (PSO) algorithm using pattern reduction and reducing the clustering calculation time with Multistart Pattern Reduction-Enhanced PSO (MPREPSO). This method adds two pattern reduction operators and multistart operators into the PSO algorithms. The goal of the pattern reduction operator is to reduce the computational time from the compression of static patterns. The purpose of the multistart operator is to avoid falling into the local optimal by enforcing diversity in the population. Two pattern reduction and multistart operators are combined with the PSO algorithm to evaluate the performance of this method.

Keywords Big data · Clustering · Particle swarm optimization · Pattern reduction · Meta-heuristic algorithm

Introduction

One of the most critical indicators in the information world is data processing. Nowadays, big data analysis is getting more and more attention from researchers. Big data is defined as a data set whose size exceeds a typical database or computer [1]. One of the best ways to process and work with data is clustering. Clustering is an unsupervised technique that is part of pattern recognition, data mining, and machine learning. The goal of clustering is to group unlabeled data (called clusters) so that the data in one cluster are more similar and different from the data in other clusters [2]. Therefore, the similarity between the data in each group is the largest, and the similarity between the data in different groups is the smallest. The ability to cluster into the data space and identify its structure makes this method ideal for handling the vast data world [3]. The process of dividing different data into detached groups and grouping similar data into the same group is called clustering based on predefined similarity standards. Bioinformatics and pattern recognition can be used in various domains, such as image processing and web mining [4]. Different standards of similarity can be considered between the data in the cluster. For example, in document clustering, the similarity of the data is proportional to the number of common words in the two documents; In the clustering of the customers' shopping carts, it is determined based on the similarity of the purchases. Therefore, calculating the similarity between two data sets is very important in clustering, because it will change the quality of the final result. The distance represented by the heterogeneity allows movement in the data space and creates clusters. By calculating the distance between the two data, we can understand the closeness and position in the

This article is part of the topical collection "Advanced Computing and Data Sciences" guest edited by Mayank Singh, Vipin Tyagi and P.K. Gupta.

✉ Seyed Emadedin Hashemi
emad.hashemi88@gmail.com

Madjid Tavana
tavana@lasalle.edu
http://tavana.us

Maryam Bakhshi
maryambakhshi990@gmail.com

[1] Department of Industrial Engineering, Islamic Azad University Arak Branch, Arak, Iran

[2] Business Systems and Analytics Department, Distinguished Chair of Business Analytics, La Salle University, Philadelphia, USA

[3] Business Information Systems Department, Faculty of Business Administration and Economics, University of Paderborn, Paderborn, Germany

cluster. Because of the many applications of clustering in most fields, it seems necessary to improve the clustering algorithm [5]. Due to its complexity, the clustering problem can be considered an NP-hard problem [6]. Therefore, the clustering algorithm usually takes a long time to find an approximate solution. Many researchers focus on finding better solutions or using techniques such as data sampling to speed up clustering algorithms [7]. Several studies aim to improve the speed of clustering algorithms by reducing the number of comparisons between patterns and centroids of large-scale, high-dimensional data sets. These studies can be divided into three categories: dimensionality reduction [8], Centroid reduction [9], and Pattern reduction [10]. One of the applications of particle swarm optimization is to solve the clustering problem by using multiple search directions with social behavior to improve the quality of clustering results [11]. One significant advantage of particle swarm optimization (PSO) is that it can be used to avoid the problem of convergence to the nearest local state [12]. PSO-based clustering algorithms have been used successfully in many big data problems [13]], But the computation time problem will certainly affect the performance of any system. Speed of clustering, we study clustering using an improved particle clustering algorithm, and at the same time, we try to maintain or even improve the quality of clustering results.

Literature Review

In today's world, due to the existence of multiple communication networks, data generation is not a one-time process, but it gets generated regularly and dynamically [14]. The new digital world is overwhelmed with huge volumes of data [15], and exposure to large volumes of data in various fields has increased significantly. So big data is a big challenge for research based on data analysis [16]. As a result, these types of data analysis are particularly important. Topics of this subject of data analysis in machine learning are defined as clustering. Clustering is a technique that extracts natural groupings hidden in data to simplify them into meaningful and understandable information. The resulting groups gather similar objects based on their features to form clusters. The applications of clustering techniques have been used in various areas, including web analysis, business, marketing, education, data science, and medical diagnosis, among others. Data clustering is the process of partitioning the elements of a dataset (n) into several clusters (k clusters) in a d-dimensional Euclidean space based on similarity samples such as data samples belonging to the same cluster more associations. Still, there needs to be more separation between different cluster data items. At the same time, such a clustering must be computationally efficient and sustainable; However, when data is generated on agile development networks, reclustering may need to be done more frequently and in as little time as possible [17]. Hence researchers need to find clustering solutions so that big data clustering gets completed in a reasonable time and is sustainable as well over the dynamic data. While the idea of clustering was first introduced in 1935, it is currently attracting the attention of many researchers [18] who are producing great advances. Since clustering is present in various applications, various methods have been proposed to exploit it. These methods can be divided into two main distinct groups:

- *Hierarchical clustering* is the sequential formation of groups whose members are most similar to each other or the sequential separation of groups whose members are most similar. There are no problems in this type of clustering due to initialization and local minima.
- *Segmentation clustering* involves segregating input data into a specified number of clusters. Such methods usually seek separations that optimize the competency function locally. The algorithm is executed several times at different starting points to improve the clustering quality, and the best condition obtained from all execution times is selected as the clustering output [19].

Hierarchical clustering generates a tree-like hierarchical structure for partitioning the data. In contrast, segmentation clustering divides the big data into non-overlapping clusters such that each data item belongs to only one cluster. Segmentation clustering is mainly used for data partitioning [20].

Given the wide range of applications of big data clustering, there is a need for clustering methods that can group large volumes of data with appropriate accuracy and speed. Clustering problems are computationally Np-Hard, so clustering algorithms usually take a very long time to arrive at an approximate solution [6]. If the data size is too big, arranging the data through traditional deterministic techniques into clusters takes huge processing and exponential time. It is practically not much useful, as such computing may normally be carried out in a mobile computing environment, where efficiency and the sustainability of this computing are necessary. Several studies have focused on reducing computational time. Recently, some heuristic and evolutionary algorithms have been used for data clustering. Researchers have developed many evolutionary techniques for clustering problems like Genetic Algorithm, Harmony Search, Gravitational search algorithm, Bee Colony optimization, etc. The application of clustering algorithms is expanding due to the rapid growth of data volumes. With the growth of data volumes, there are problems associated with their storage, processing, and transmission in recent years. Existing algorithms are increasingly difficult to cope with the growing flow of datasets [21]. In most algorithms, we

initially need to specify the number of clusters as an input. However, in some applications, sometimes it is not known in advance. Therefore, the number of clusters during the execution time must be specified. Clustering algorithms are used in such areas as supply chain [22, 23], Natural Language Processing [24, 25], Internet of Things (IOT) [26], medicine [27, 28], and technical sciences [29].

Until now, numerous meta-heuristic optimization algorithms have been proposed by researchers based on inspiration from nature. Particle swarm optimization was introduced by [30]. They originally intended to use existing social models and social relations to create a kind of computational intelligence that meets individual capabilities. Their work led to developing a robust optimization algorithm, called the Particle Swarm Optimization Algorithm or PSO, adapted from the collective performance of groups of animals such as birds and fish [31, 32]. Particle Swarm Optimization (PSO) is one of the nature-inspired meta-heuristic optimization algorithms that have huge attention in clustering due to its self-orientation towards an optimal solution and its flexibility to use it with other techniques [33]. PSO aims to simulate the flocking behavior of birds based on the nearest neighbor principle that regulates their velocity and position concerning their nearest neighbors, devoid of any central coordination; governed by the flocking rules viz.-flock centering, collision avoidance, and velocity matching [34]. PSO algorithm is widely used in cluster analysis. However, it is a stochastic technique vulnerable to premature convergence to sub-optimal clustering solutions. PSO-based clustering algorithms also require tuning of the learning coefficient values to find better solutions. The latter drawbacks can be evaded by setting a proper balance between the exploitation and exploration behaviors of particles while searching the feature space. Moreover, particles must consider the magnitude of movement in each dimension and search for the optimal solution in the most populated regions in the feature space.

Alswaitti et al. [35] proposed a novel approach for data clustering based on particle swarms. In this proposal, the balance between exploitation and exploration processes is considered using a combination of (i) kernel density estimation technique associated with new bandwidth estimation method to address the premature convergence and (ii) estimated multidimensional gravitational learning coefficients. Sustainable grouping of Big data into various clusters is an open research problem that aims to provide computationally efficient solutions and maintainable solutions over dynamic data. Sharma and Chhabra [17] proposed a sustainable clustering algorithm by hybridizing PSO with a mutation operator to cluster the data generated from different networks. The data generated by such networks are usually dynamic and heterogeneous, and the number of clusters is not fixed/known in advance. Hence, the proposed algorithm is further extended for automatically generating and re-adjusting the clusters over mobile network devices, facilitating the generation of sustainable clusters. The results show that the proposed algorithm is efficient in creating well-separated, compact, and sustainable clusters. Sharma and Chhabra [36] proposed a new dimension called family memory has been introduced to enhance PSO-based clustering's performance wherein the data were clustered and the particles. Earlier studies on PSO-based clustering had used uniform distribution to create the particles' coordinates. Rengasamy and Murugesan [37] proposed a novel Feature Linkage Weight with PSO (FLWPSO) by combining the feature linkage-based weight reduction and Particle Swarm Optimization to build a novel cluster called Feature Linkage Weight with Particle Swarm Optimization (FLWPSO). The proposed clustering method can identify the centroids in a user-defined number of clusters. Malarvizhi and Amshakala [38] proposed a new hybrid algorithm that utilizes Cuckoo Search (CS), PSO, and k-means. The proposed algorithm employed PSO and k-means to produce new nests in standard CS to obtain better results. It also benefits from Mantegna levy distribution to obtain higher convergence speed and local search. Evaluation results show that the proposed algorithm is an efficient method for data clustering and produces more optimized results than standard PSO. Tarkhaneh et al. [39] focuses on the research and analysis of the container scheduling algorithm and an intelligent PSO algorithm based on the Kubernetes container cloud system for big data applications. The PSO algorithm was used by Liu et al. [40] to cluster the data to increase the quality of clustering results by using multiple searches with social behavior. Omran et al. [41] proposed a new parallel batch clustering algorithm based on the k-means algorithm. The proposed algorithm splits a dataset into equal partitions and reduces the exponential growth of computations. The goal is to preserve the dataset's characteristics while increasing the clustering speed. The centers of the clusters are calculated for each partition, which is merged and also clustered later. Alguliyev et al. [42] proposed a method for clustering n data objects using k-means and gravitational search algorithms. A hybrid method for improving k-means performance in data clustering using the colonial competition algorithm was proposed by Hatamlou et al. [43]. The K-mean algorithm is one of the most popular and easily implemented splitting clustering. Unfortunately, the original version has limitations, such as dependence on initial values and the optimal local response convergence. By combining this algorithm with the PSO algorithm, these limitations have been eliminated, and significant results where the convergence rate has been much higher than the previous samples have been obtained. An

important advantage of PSO is that it can avoid convergence to the nearest local optimum. However, several studies have shown that the PSO algorithm suffers from premature convergence to local optima [44], which occurs due to the lack of diversity of PSO and the inability of particles to explore the entire feature space. A set of parameters (learning coefficients) must be tuned in the PSO algorithm, which may affect its performance [45]. The values of these parameters are often set manually despite their effect on controlling the exploitation and exploration processes [46]. Consequently, many attempts have been proposed to enhance the performance of the PSO clustering algorithm to solve the premature convergence and the tuning of learning coefficients [47, 48].

According to the issues mentioned above, this paper proposes a new approach for data clustering based on the PSO algorithm. This study aims to investigate a clustering method using a Multistart Pattern Reduction-Enhanced PSO (MPREPSO). Because clustering algorithms usually take a very long time to arrive at an approximate solution, the computational speed of clustering algorithms is low. Therefore, the ultimate goal of this research is to accelerate the speed of the clustering algorithm.

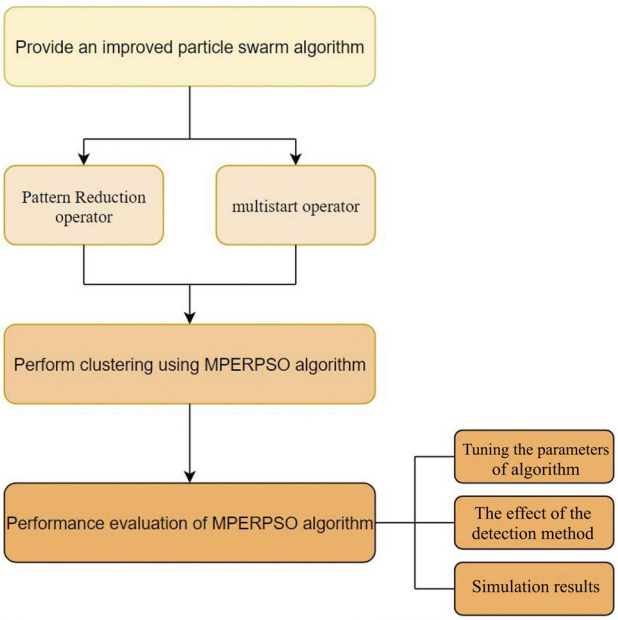

Fig. 1 A schematic view of the proposed approach

Methodology

In addition to introducing the capabilities of the PSO algorithm, this study also exposed the shortcomings of clustering. In this regard, a PSO-based algorithm is introduced that covers the disadvantages of PSO and includes advantages over other PSO-based algorithms. The content process of this paper is illustrated in the following Fig. 1:

PSO Algorithm

The PSO method is an optimization method that can deal with problems whose answer is a point or surface in n-dimensional space. In such an atmosphere, hypotheses are put forward, and an initial velocity is assigned to the particles and communication channels between the particles are considered. The particles move in the response space, and the results are calculated based on the "competency function" after each time interval. Over time, particles move toward particles that have higher suitability. The main advantage of this method over other optimization strategies is that a large number of condensing particles makes the method flexible in the face of the problem of local optimal response. Figure 2 shows an example of the movement of particles in the search space. Part 1 shows the initial position of the particles in the two-dimensional search space, and with the repetition of the algorithm, the particles eventually converge as part 6.

Each particle has a position that determines the particle's position in the search space. As the particle moves, the position of the particle changes over time. The formula for updating the position of a particle is given in Eq. (1).

$$\begin{aligned} p_i(t+1) &= p_i(t) + v_i(t+1) \\ p_i(t) &\sim U(p_{\min}, p_{\max}) \end{aligned} \tag{1}$$

In this equation, $p_i(t)$ denotes the position of the i-th particle in time t. Also, each particle needs a velocity to move in space, $v_i(t)$ represents the velocity of the i particle at time t. By adding speed to the position of each particle, a new position for the particle can be considered. A competency function evaluates the quality of a particle's position in the search space. Particles have the ability to remember the best position they have been in during their lifetime. The best individual experience of a particle or the best position met by a particle is indicated by pbest. Particles can also be aware of the best position the whole group meets; this position is denoted by gbest. The velocity vector in the optimization process reflects the particle's empirical knowledge and the particle community's reflection. Each particle considers the following two components to move in the search space:

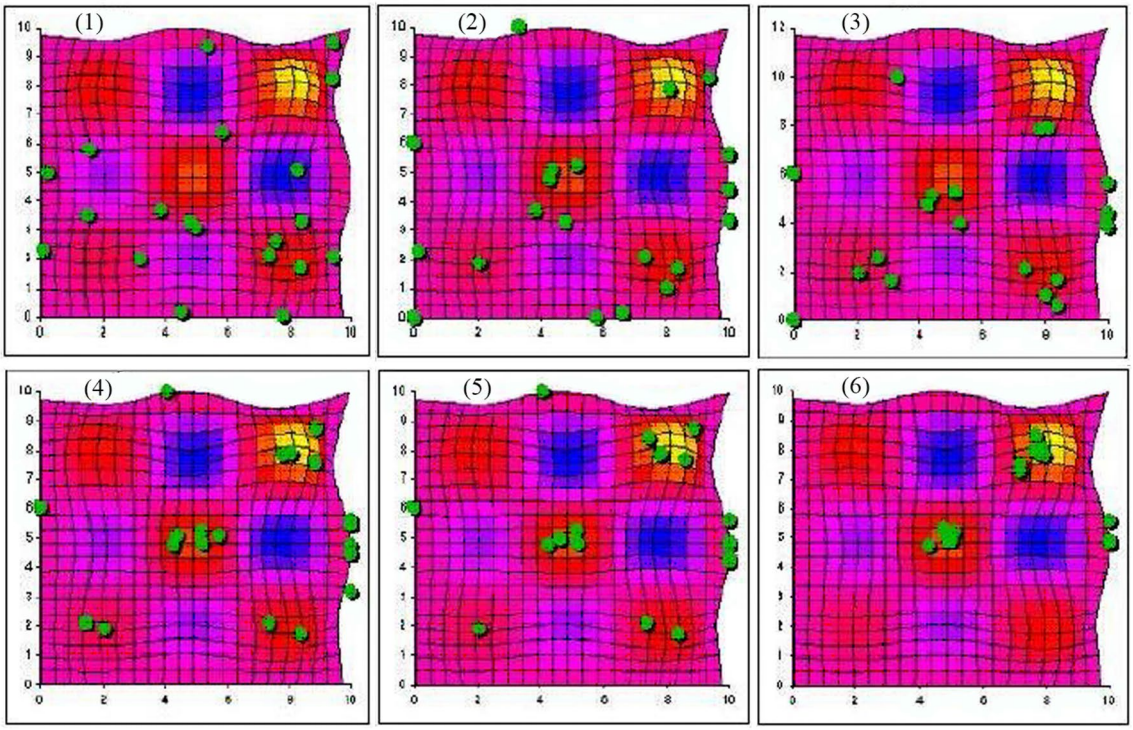

Fig. 2 The process of moving a particle in a group

Cognitive component: $pbest_i(t) - p_i(t)$ is the best solution that a particle alone can obtain.

Social component: $gbest_i(t) - p_i(t)$ is the best solution that is recognized by the whole group.

There are two main models that calculate their velocity vectors for the standard PSO algorithm based on both cognitive and social components. These two models are called lbest PSO and gbest PSO, and the difference between them is in the neighborhood that is considered for each particle.

Gbest PSO Model

In this algorithm, the neighbor of one particle is the total group of particles, and the formula for updating the best position of each particle is calculated by Eq. (2) (f is a competency function.)

$$pb_i(t+1) = \begin{cases} pb_i(t) \text{if} f(p_i(t+1)) \geq f(pb_i(t)) \\ p_i(t+1) \text{if} f(p_i(t+1)) < f(pb_i(t)) \end{cases} \tag{2}$$

The vector $gb(t)$ is calculated as Eq. (3).

$$\begin{aligned} &gb(t) = pb_i \\ &s.t: \\ &f(pb_i) = \min\{f(pb_0(t)), f(pb_1(t)), \ldots, f(pb_s(t)) \end{aligned} \tag{3}$$

Therefore, the velocity of the i particle is updated based on 4.

$$\begin{aligned} V_{ij}(t+1) = {} & wV_{ij}(t) + C_1 r_{1j}(t)\left(pb_{ij}(t) - p_{ij}(t)\right) \\ & + C_2 r_{2j}(t)\left(gb_j(t) - p_{ij}(t)\right) \\ & j \in 1, \ldots, d \\ & r_{1j}(t), r_{2j}(t) \approx U(0,1)) \end{aligned} \tag{4}$$

w denotes the weight of inertia, and C_1,C_2 are the constant values. This weight actually affects a percentage of the previous particle velocity in calculating the new velocity. The higher the value, the higher the public search, and the lower the weight, the higher the local search. The velocity and position of each particle can be calculated by Eqs. (1), (4), respectively. They are updated, and this operation is repeated until the maximum number of repetitions is reached or the updated speed is close to zero. The performance of each particle is estimated by the competency function.

Lbest PSO Model

A particle can only communicate with a number of particles in its neighborhood to update its speed. N_i represents the set

of neighboring particles of a particle. The topology used to communicate between particles is the ring topology.

$$\begin{aligned} N_i &= \left\{pb_{i-l}(t), pb_{i-l+1}(t), \ldots, pb_{i-1}(t), pb_i(t), pb_{i+1}(t), \ldots, pb_{i+l-1}(t), pb_{i+l}(t)\right\} \\ gb(t+1) &\in \left\{N_i | f(gb(t+1) = \min\{f(pb_i(t))\} \forall pb_i \in N_i\right\} \\ V_{ij}(t+1) &= wV_{ij}(t) + C_1 r_{1j}(t)\left(pb_{ij}(t) - p_{ij}(t)\right) + C_2 r_{2j}(t)\left(gb_j(t) - p_{ij}(t)\right) \end{aligned} \tag{5}$$

Neighbors of a particle practically determine the social behaviors of a particle, so the topology of neighbors is an important and useful issue. If $l = s$, lbest acts like gbest, the neighbors of a particle are all the particles in the group.

Topology or Social Network Structure

A variety of topologies can be considered for the neighbors of a particle. The most common topologies used are ring and star. In star topology, a particle is considered the center, and all the particles in the group are attached. All other particles are attached only to the center. In a ring topology, particles are in a ring, and each particle has many neighbors on the left and several neighbors on the right. Other types of network topology include the following:

- BUS TOPOLOGY
- MESH TOPOLOGY
- TREE TOPOLOGY
- HYBRID TOPOLOGY

The choice of topology greatly affects finding the best solution for the group. Using the gbest model, the speed is greatly increased. Using a ring topology slows down the response. This is because the best solution is based on many neighbors that do not affect the entire group of particles. This slowing down of the response enables the particles to search more areas of space and prevents premature convergence.

PSO Problems

The particle optimization algorithm has several weaknesses. For example, there is a possibility of particles in local optimizations in this algorithm. Although PSO is faster than all evolutionary algorithms, it usually cannot compensate for the quality of the solution by increasing iterations. One reason is that in this algorithm, the particles converge to a specific point between the best general position and the best personal position. Many changes have been made to the PSO to address this weakness. One of these changes is the improvement in the weight of inertia or w. Another disadvantage of this method is its dependence on the problem. This dependence is usually the result of changes in the algorithm parameters. In general, one parameter cannot be applied to all problems. Many parameters affect the PSO algorithm, including speed control, the weight of inertia, number of particles, number of neighbors, and speed coefficients. Hence, we seek to improve the particle swarm algorithm for clustering.

Clustering Using Particle Swarm Optimization

The algorithm below the pseudo-clustering code using the PSO algorithm shows that the k center of the cluster is directly encrypted as the position of a particle:

1. Create an initial population of particles, each of which contains k randomly generates centroids.
2.For each particle i
3. For each pattern x
4. Calculate the distance between x and all the centroids.
5. Assign x to the closest cluster
6. End
7. Calculate the fitness value.
8. Update the personal best pb_i of each particle and the global best gb.
9. Change the velocity and position.
10. End
11. If the stop criterion is satisfied, then stop and output the best particle ;otherwise, go to step 2

For example, $p_i = \left\{c_{i1}, c_{i2}, \ldots, c_{in}\right\}$ represents the i-th particle, so that c_{ij} represents the j-th center of the i-th particle. The suitability of each particle is obtained using the Eq. (6):

$$f(P_i, M_i) = w_1 d_{\max}(P_i, M_i) + w_2[Z_{\max} - d_{\min}(P_i)] \tag{6}$$

$$d_{\max}(P_i, M_i) = \max_{j=1,\ldots,k}\left[\sum_{\forall x \in \pi_{ij}} \frac{d(x, c_{ij})}{\left|c_{ij}\right|}\right] \tag{7}$$

$$d_{\min}\underset{\forall a,b,a\neq b}{(p_i)} = \min\{d(c_{ia}, c_{ib})\} \tag{8}$$

where $d_{\max}$ represents the maximum mean square of the distance within the cluster and $d_{\min}$ represents the minimum distance between all centers. The M_i matrix represents the assignment of patterns to the encoded centers of the i particle and $Z_{\max}$ is the largest property value in the data set.

Proposed Improved Particle Swarm Optimization Algorithm

As mentioned above, to evaluate the method considered in this study, pattern reduction and multi-start operators must be combined with different versions of the PSO algorithm. Therefore, in this section, we review several versions of PSO.

- *Comparative PSO Algorithm (CPSO)*
  In PSO, the position and velocity of each particle were randomly assigned in the early stages, and the particles were distributed in the d-dimensional search space. One particle will be set as GBest. This particle shows the best performance of all particles. After several iterations, the particles will be affected by PBest and GBest and will slowly accumulate towards GBest. If PBest does not change, particles will increasingly accumulate around GBest. However, the available particles form clusters around GBest, which predicts the search process. A fact that leads to useless computational results in subsequent iterations is called early convergence. Due to this problem of early convergence in the PSO algorithm, the CPSO algorithm was introduced to improve the PSO algorithm. GBest represents the best particle solution that affects all particles in the updated iteration process. But in the CPSO algorithm, it is assumed that each particle has a small share in the search to achieve the best answer in the search space [58].
- *Evolutionary PSO Algorithm (EPSO)*
  The main idea of the EPSO algorithm is based on the overall design of the PSO combined with a selection method in which the best particle in each step is selected by a racing method [59].
- *local best PSO algorithm (LPSO)*
  This algorithm is another improvement in the PSO algorithm that aims to prevent premature convergence. A ring topology is used for this purpose. In a ring topology, the particles are in a ring, and each particle has many neighbors on the left and several neighbors on the right. Since each particle is connected to only two other particles in a ring topology, the best global position for each particle is different, which will reduce the convergence speed of the algorithm. The low convergence rate enables particles to search larger areas of space and prevents premature convergence. A shrinkage factor is also used in the particle velocity update equation in this algorithm. The purpose of the shrinkage factor is to strike a balance between the local and global search of the algorithm. This factor has a similar function to inertial weight [60].

The Proposed Algorithm

In this research, an efficient method (called MPREPSO) based on the concept of pattern reduction, has been investigated to reduce the computation time of PSO-based clustering algorithms. This method adds two pattern reduction operators and a multistart operator to PSO-based algorithms. The purpose of the pattern reduction operator is to reduce the computational time by compressing repetitive patterns, and the purpose of the multi-starter operation is to prevent trapping in the local optimization by applying diversity to the population.

Pattern Reduction Operator

The pattern reduction operator is divided into detection operator and compression operator, each of which is described below:

•*Detection Operator*

The purpose of this operator is to detect fixed patterns. Fixed patterns are patterns whose cluster is unlikely to change. Fixed patterns can be identified in two ways:

1. Patterns that are at a certain distance from the center of the cluster.

2. Patterns that remain in a cluster after a certain number of repetitions.

In this research, a simple and fast method to determine patterns that are located at a certain distance from the center of their cluster is presented. Assuming that α percent of the patterns in each cluster is constant. To determine the value of α (α percent) of patterns that are close to the center of a cluster, we calculate the average distance of the patterns from the center μ and the standard deviation σ, and using Eq. (9), we obtain the value of γ. Now the pattern is fixed or related to the α percent whose distance to the center is less than the value of γ.

$$\gamma = \mu \pm b\sigma \tag{9}$$

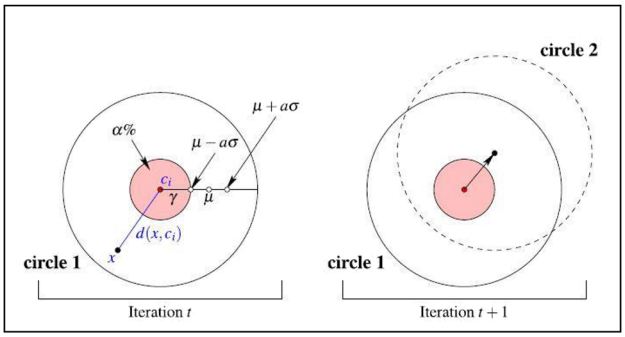

Fig. 3 Explain a simple example of how a detection operator works

In Eq. (9), $b \geq 0$ represents the σ (standard deviation) required to obtain the distance. The distance is a threshold for filtering dynamic patterns and a parameter for balancing accuracy and speed. This improves the convergence of the particle swarm algorithm.

As shown in Fig. 3, the center of circle 1 represents the center of cluster c_i in repetition t, and all patterns within circle 1 belong to this cluster. When the values of μ and standard deviation (σ) are calculated, the MPREPSO algorithm can easily differentiate the α percent of the patterns considered as constants. The right-hand side of Fig. 3 shows that when the center of a cluster moves in iteration t + 1, patterns that are fixed in iteration t are unlikely to move toward other clusters. Thus, the patterns within the colored circle in t-iteration do not change the cluster to which they belong in t + 1 iteration.

In the second method of detection, the number of iterations that a pattern remains in a cluster is counted. The more iterations that the pattern remains in a cluster, the more likely it is that the pattern will be fixed. The number of iterations that a pattern needs to stay in a cluster depends on convergence speed. If the number of iterations is considered large, the accuracy will be high, but the calculation time will be increased, and if the number of iterations is considered a small value, the convergence speed will be increased [5].

Compression operators: After identifying the fixed patterns, we use the compression operator as follows. Suppose R represents a set of fixed patterns belonging to a particular cluster (Fig. 4).

1. Tamar The patterns belonging to the set R are compressed into a pattern named r.

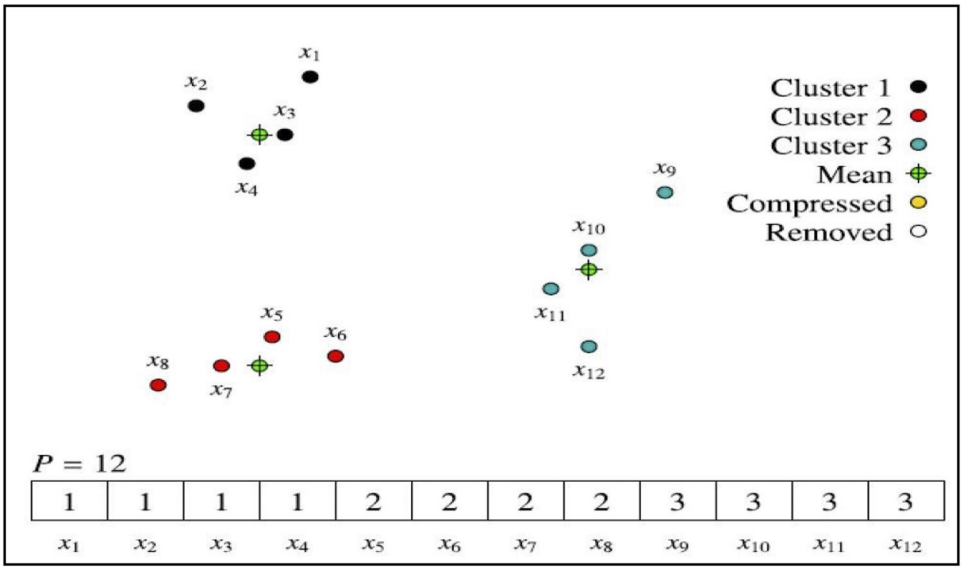

Fig. 4 At the end of t – 1 iteration, 12 patterns are assigned to 3 clusters

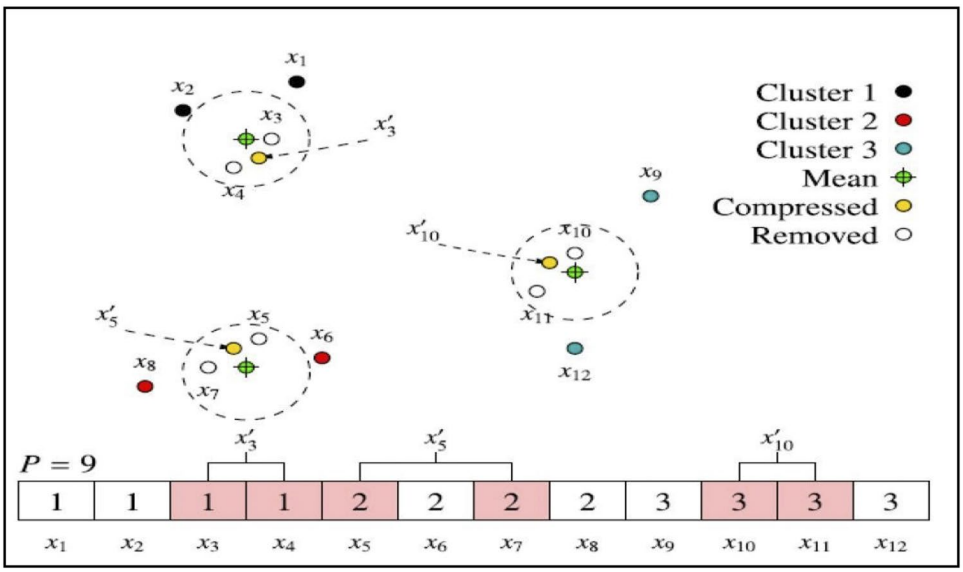

Fig. 5 In repetition t, patterns close to their means are compressed and removed themselves

2. By adding r to set X and removing all the patterns of set R from set X, the calculations related to the patterns in R will be removed. Set X is defined as follows:

$$X = (X \cup \{r\}) \backslash R \tag{10}$$

As shown in Fig. 5, the number of patterns is reduced from 12 to 9, so by compressing and eliminating duplicate patterns, the computation time is significantly reduced.

Multistart Operator

The multi-starter operator is used to vary the MPREPSO algorithm to prevent premature convergence, which may reduce the quality of the final result. The steps to do this

1. Remove particles that have a lower fit than the average fit of all particles.
2. Generation of new particles using the random cloning method.
3. Combine the tera selected in step 1 and the particles created in step 2 to create a new population for the algorithm.

Using this method, the multi-starter operator generates particles that find a better solution than the current search path. For this reason, the multi-starter operator can prevent MPREPSO from being trapped in local optimization.

Multistart operator steps

Step 1: Assume the particles remaining after the pattern reduction operator are $p_1 = \{c_{11}, c_{12}, c_{13}\}$, $p_2 = \{c_{21}, c_{22}, c_{23}\}$, $p_3 = \{c_{31}, c_{32}, c_{33}\}$ and $p_4 = \{c_{41}, c_{42}, c_{43}\}$, each of which is encoded as 3 centers. Assuming that p_3 and p_4 particles have a lower fit than the average fit of the particles, we exclude them from the population.

Step 2: The first center of the new particles can take one of the two values of c_{11} or c_{21} , the second center of the new particles is one of the two values of c_{12} or c_{22} , and the third center of the new particles is one of the two values of c_{13} or c_{23}.

Step 3: The particles produced in step 2, (p_3, p_4) together with the particles remaining from step 1, (p_1, p_2) form a new population.

iN summary, the steps of the multi-starter operator are as shown in Figure 6.

Steps of clustering using the studied algorithm

Pseudo-code algorithm MPREPSO
1.Randomly select m subset of patterns, S_i denoted for $i = 1, 2, ..., m$ from X by sampling.
2. Create an initial population of m particles, each of which encodes the k centroids obtained by applying k-means to the corresponding S_i.
3. For each particle i
4. For each pattern $x \in X$
5. Calculate the distance between x to all the centroids,denoted $C_i j$ for $j = 1, 2, ..., k$
6. Assign x to the closest cluster.
7. End
8. Calculate the fitness value using Eq. (6).
9. For each cluster
10. Detect the set of static patterns R.
11. Compress the set of static patterns R into a single patterns r and remove R;i.e... , $X = (X \cup \{r\}) \backslash R$
12. End
13. End
14. Update the personal best pb_i and the global best gb using Eqs. (1),(4).
15. Change the velocities and positions.
16. Perform the multistart operator.
17. If the stop criterion is satisfied, then stop and output the best particle; otherwise . go to step 3.

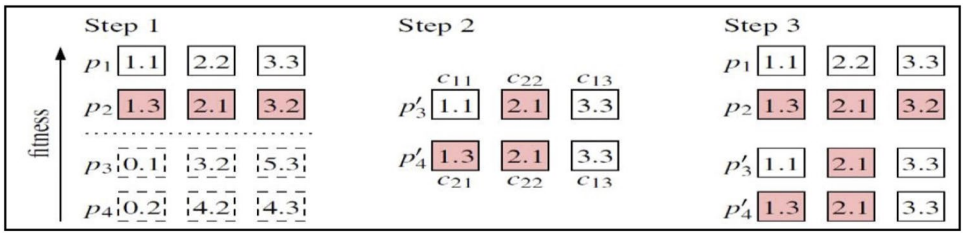

Fig. 6 steps of the multi-starter operator

According to the pseudo-code of the MPREPSO algorithm, the clustering steps using the MPREPSO algorithm are as follows:

In step 1, we select m subsets of the patterns as random sampling and denote them by S_i , $i = 1, 2, \ldots, m$. In step 2, we create the initial population containing m particles. Each particle in this population represents the center k, obtained by applying the K-mean algorithm to the corresponding S_i. Then in steps 3–7, using an iterative process, the distance of each pattern to all centers is calculated, and each pattern is assigned to the nearest center. In steps 9–13, the pattern reduction operator is applied to each cluster. In step 14, the best individual position for each particle and the best global position are updated. In step 15, the speed and position of each particle is updated, and then in step 16, the multi-starter operator is applied. Finally, if the stop condition is met, the algorithm is stopped and the best particle is considered as the output; otherwise, the algorithm will return to step 3.

The flowchart of the proposed algorithm is shown in Fig. 7.

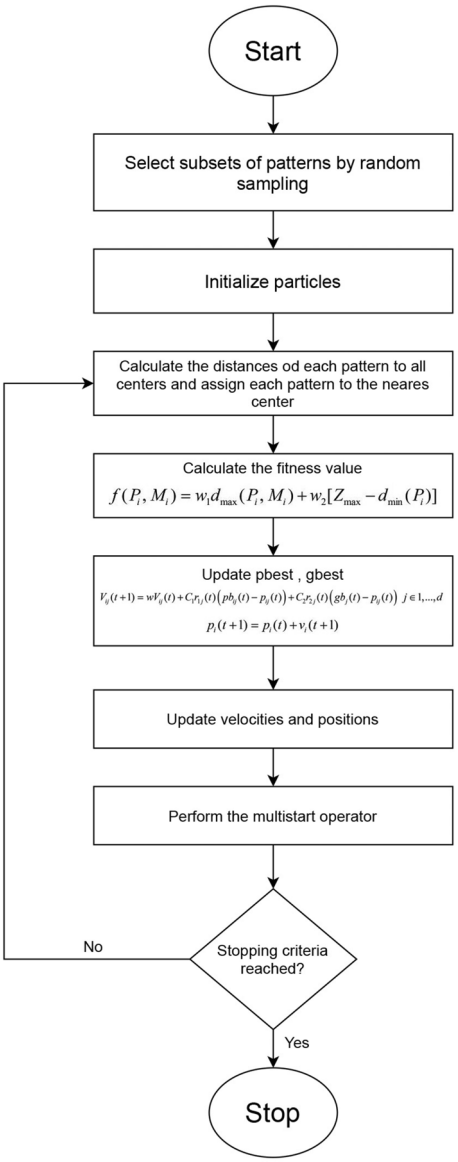

Fig. 7 Flowchart of MPREPSO algorithm

Algorithm Complexity

The proposed algorithm relies on two kinds of detection different kinds of detection operators to find computations that are essentially redundant. More precisely, the first detection operator considers patterns within a predefined radius γ to their centroids static. This operator uses a simple approach to determining the top α% of patterns for each cluster that can be considered as static by using the standard deviations σ, mean μ, and confidence intervals of patterns in that cluster so that no sorting is required. This reduces the time complexity. Note that we are assuming that the distances of all the patterns in a cluster to their centroid are normally distributed. Accordingly, to find the top 20% of patterns that are close to the centroid of a cluster, the detection operator only

Table 1 Standard data set for evaluating clustering algorithms

Data sets	Number of dimensions	number of samples
Iris	4	150
Wine	13	178
Brest cancer	9	286
Car evaluation	6	1728
Statlog	13	270
Yeast	8	1484

needs to compute the average distance (μ) of the patterns to their centroid and the standard deviation σ. A pattern will be in the top 20% (i.e.,$\frac{100-60}{2}\% = 20\%$) if its distance to the centroid is smaller than $\gamma = \mu - \sigma$. The second detection operator checks to see if a static pattern can be eliminated by counting the number of iterations that pattern stays in the same cluster. Intuitively, the higher the number of iterations a pattern remains in the same cluster, the more likely the pattern is static. How many iterations a pattern needs to stay in the same group depends on the convergence speed or the quality of the end result. If we set the number of iterations to a large value, the accuracy rate will be high, but the computation time will increase. Conversely, the computation time will decrease if we set the number of iterations to a small value.

Calculations and Evaluation

This section combines two pattern reduction and multi-starter operators with PSO and uses them to cluster standard datasets to compare these algorithms' quality improvement and execution time. At the beginning of this section, we introduce the data set that is intended for clustering. In the following, we express the simulation results using two factors, quality and execution time.

Introducing the Data Set

Standard data sets are commonly used to evaluate the efficiency of data clustering. This dataset can be found in the UCI dataset [61]. The data sets we use in this study are shown in Table 1.

Evaluation of the MPREPSO Algorithm Performance

Combine the two pattern reduction operators and the multi-starter into four PSO-based clustering algorithms: Standard PSO (SPSO), Comparative PSO (CPSO), Local best PSO (LPSO), and Evolutionary PSO (EPSO), and use the obtained algorithms to cluster the six datasets according to Table 1.

We use the following two methods to identify fixed patterns:

1. Patterns that are at a certain distance from the center of the cluster.

2. Patterns that remain in a cluster after a certain number of repetitions.

Property For simplicity in common, if we use the first detection method, we use PR_1, and if we use both detection methods, we use PR_2, and also if we use both detection methods and the multi-starter operator, we use PR_2 or We use the same MPREPSO algorithm. Also, to compare the

Table 2 Tuning the parameters

Algorithm	Parameter settings
SPSO	$\omega = 0,\ \ a_1 = a_2 = 1.49,\ \ m = 20$
LPSO	$\omega = [0.4\ \ 0.9],\ \ a_1 = a_2 = 1.49,\ \ m = 20$
CPSO	$\chi = 0.72984,\ \ a_1 = a_2 = 2,\ \ m = 50$
EPSO	$\omega_1 = \omega_2 = \omega_3 = \omega_4 = 0.5,\ \ \rho = 2,\ \ m = 20$

Table 3 Comparison of the results of the studied algorithm using both detection methods

Dataset	Δ_{AR}		Δ_T	
	PR_1	PR_2	PR_1	PR_2
Iris	1.133758	1.170701	– 6.101720	– 8.320974
Wine	8.680628	10.366492	– 5.759124	– 6.375079
Brest cancer	0.807229	0.807229	– 40.371306	– 6.827217
Car evaluation	0.005362	0.005362	– 43.422772	– 6.249234
Statlog	– 2.585612	– 2.759712	– 13.220179	– 9.847898
Yeast	0.001334	– 0.001779	– 29.046196	– 8.316151

MPREPSO algorithm on different versions of PSO, we use two criteria of quality and execution time. The quality of the clustering results, which is the same as the accuracy, is obtained using Eq. (11).

$$AR = \frac{\sum_{i=1}^{n} A_i}{n} \times 100 \tag{11}$$

A_i is replaced by one of the two values 0 or 1. $A_i = 1$ indicates that the pattern x_i is properly clustered, and $A_i = 2$ indicates that x_i is assigned to the wrong cluster.

In the first detection method, if the distance of a pattern to the center of its cluster is less than γ in Eq. (4), or according to the second detection method, if a pattern remains in the same cluster after 2 repetitions, you consider that pattern constant and also to compare the algorithm. We use Eq. (12).

$$\Delta_\beta = \frac{\beta_\phi - \beta_\Psi}{\beta_\Psi} \tag{12}$$

Δ_β represents an increase in β_ϕ (new algorithm) compared to β_ψ (original algorithm). $\beta = AR$ indicates the quality of the clustering results and $\beta = T$ indicates the calculation time. In this study, a larger Δ_{AR} value of m means a greater increase in the quality of the cluster, while a smaller value of Δ_T means a greater increase in the calculation time. Note that the quality of the clustering results is measured in AR for the data set.

Table 4 The effect of γ on AR quality in the first detection method

Dataset	Δ_{AR}				
	$\mu - 2\alpha$	$\mu - \alpha$	μ	$\mu + \alpha$	$\mu + 2\alpha$
Iris	2.369251	0.005364	2.018341	1.753680	– 4.637502
Wine	– 0.328742	– 0.258963	0.753095	– 0.1859605	– 0.265832
Brest cancer	– 1.359201	– 0.519684	0.125695	0.001366	0.428695
Car evaluation	16.963852	12.369254	5.369752	1.0253214	2.968526
Statlog	5.9632501	3.001263	5.018569	3.623551	2.320453
Yeast	– 3.325968	0.025369	5.652901	– 5.635820	3.321045

Table 5 The effect of γ on the results of calculating the execution time T in the first detection method

Dataset	Δ_T				
	$\mu - 2\alpha$	$\mu - \alpha$	μ	$\mu + \alpha$	$\mu + 2\alpha$
Iris	– 20.325901	– 21.658214	– 22.102536	– 24.825304	– 25.302539
Wine	– 28.352732	– 28.383252	– 28.754329	– 29.239531	– 30.690536
Brest cancer	– 28.463250	– 29.012593	– 29.453980	– 21.325991	– 21.697531
Car evaluation	– 22.342682	– 22.432613	– 22.463583	– 22.832451	– 21.342912
Statlog	– 25.804362	– 25.834623	– 26.653781	– 26.163428	– 26.243861
Yeast	– 23.853426	– 23.927326	– 23.674973	– 24.042965	– 24.682964

Table 6 Improving the quality of MPREPSO (MPR2) algorithm for different versions of PSO

Dataset	SPSO		CPSO		LPSO		EPSO	
	PR2	MPR2	PR2	MPR2	PR2	MPR2	PR2	MPR2
DSD-1	0.643725	0.753036	0.528143	0.586304	0.737374	0.602020	– 2.354552	0.973675
DSD-2	1.720991	1.910039	2.923933	3.196660	1.942997	2.000000	– 1.365289	0.695230
DSD-3	0.544269	0.558778	0.827699	0.832179	0.718213	0.718213	– 1.523452	0.068936
DSD-4	0.001335	0.0001335	0.001335	0.001335	0.000445	0.000445	– 5.753641	4.586243
DSD-5	– 2.920301	– 1.177444	– 2.278960	– 1.778960	– 2.365410	-2.747724	– 6.651450	10.56032
DSD-6	0.013989	0.014635	0.006739	0.008985	0.007640	0.011236	– 6.953620	1.963245
Average	0.000668	0.341789	0.334817	0.474417	0.173543	0.097365	– 4.100334	3.141226

Table 7 Improved runtime of MPREPSO (MPR2) algorithm for different versions of PSO

Dataset	SPSO		CPSO		LPSO		EPSO	
	PR2	MPR2	PR2	MPR2	PR2	MPR2	PR2	MPR2
DSD-1	– 17.174694	– 19.612074	– 21.319771	– 20.913227	– 25.568914	– 24.658932	– 25.586932	– 23.653241
DSD-2	– 19.019418	– 19.327061	– 17.088976	– 17.727575	– 21.625638	– 21.369570	– 25.652370	– 21.364215
DSD-3	– 19.806740	– 19.900067	– 20.898718	– 22.041473	– 25.543269	– 23.326921	– 22.397265	– 18.562536
DSD-4	– 19.674709	– 19.675221	– 19.908042	– 21.990662	– 21.257439	– 19.289634	– 23.869345	– 20.933565
DSD-5	– 18.397532	– 20.003629	– 18.350316	– 18.813792	– 18.126983	– 16.932567	– 19.149632	– 17.961255
DSD-6	– 18.333081	– 19.142404	– 17.656295	– 19.180497	– 21.682397	– 20.658932	– 20.923672	– 17.251836
Average	– 18.734362	– 19.610076	– 19.204416	– 22.111221	– 22.300773	– 21.039426	– 22.929869	– 19.954441

Tuning the Parameters of Algorithms

All simulations have been performed 30 times, and the other characteristics of the algorithm parameters are summarized in Table 2. Parameter m is the population size, χ is the contraction factor of the LPSO algorithm, and ρ is the probability of EPSO algorithm connections. The MPREPSO algorithm uses 2% of input patterns for each particle to create the initial solution. The maximum ($V_{\max}$) speed of PSO-based algorithms is 1.11.

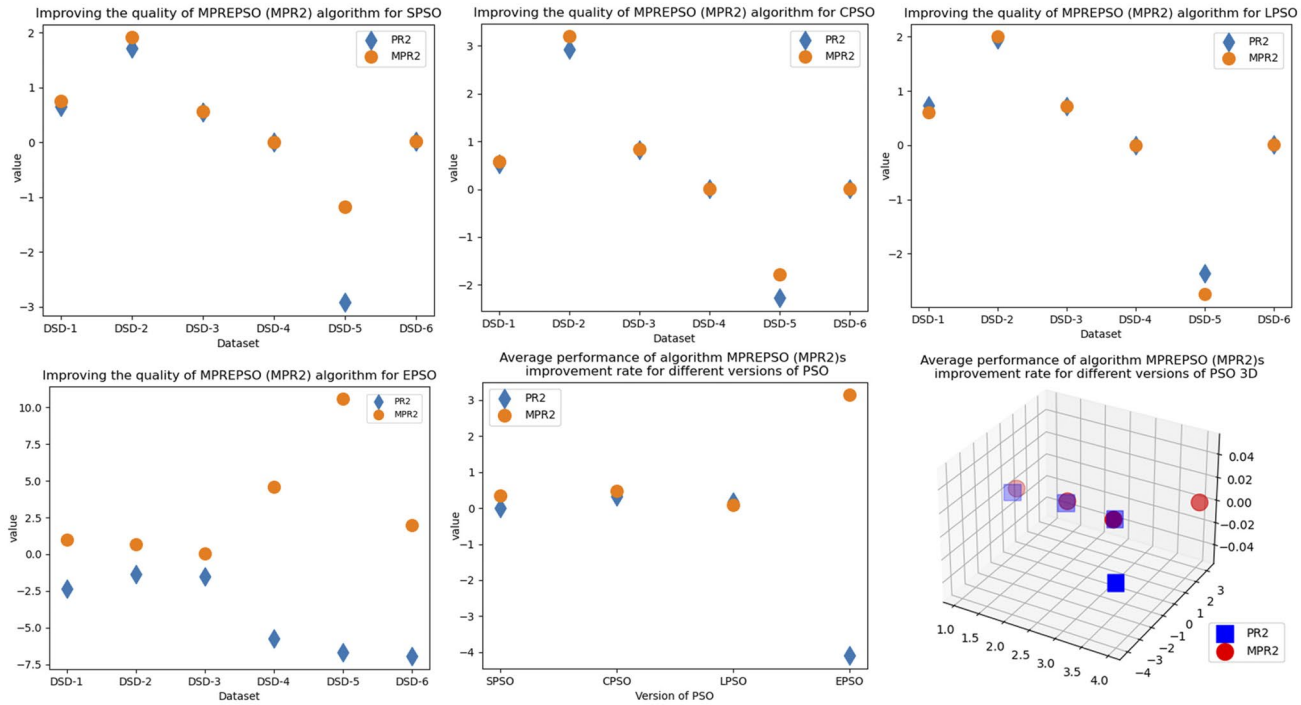

Fig. 8 Comparison of the quality of the proposed algorithm MPREPSO (MPR2) with other algorithms

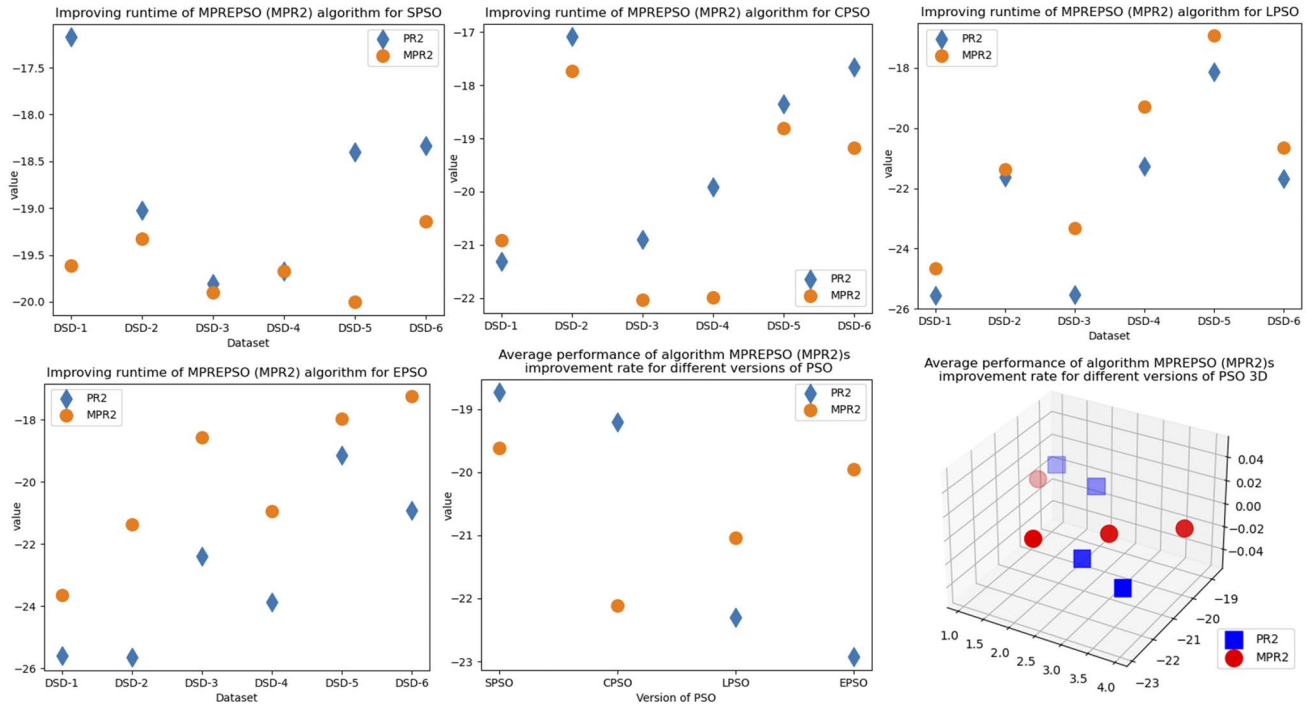

Fig. 9 Comparison of runtime of the proposed MPREPSO algorithm (MPR2) with other algorithms

The Effect of the Detection Method

To identify fixed patterns, two detection methods have been used to compress the fixed patterns together and reduce the calculations of clustering algorithms. Table 3 compares the results of these two methods for clustering the data set. Tables 4 and 5 show the simulation results using both the detector and multi-starter (MPR2) operators for different γ. These tables help us to better understand the effect of γ on the performance of the first detection operator. According to these results, the smaller the value of γ , the more patterns can be compressed and, as a result, the calculation time is reduced.

Results and Discussions

As the results in Table 7 show, the MPREPSO algorithm without a multi-starter operator (PR2) reduced the computation time of PSO-based algorithms from −20.325901% to −30.690536% but also reduced the quality of clustering results to an average of – 6.23. As shown in Table 6, the multistart operator effectively reduces quality of the pattern reduction operator. As a result, the MPREPSO algorithm (MPR2) can significantly improve the quality of clustering results. For example, compared to EPSO, MPREPSO can increase the quality of the final PSO result from – 4.1 to 3.14 (as much as 91.3%) on Average. In summary, MPREPSO can either provide better quality or even a better result than PSO-based algorithms alone. It can significantly reduce the computation time of PSO-based clustering algorithms. Tables 6 and 7 show the results of data clustering using the MPREPSO algorithm by applying SPSO, LPSO, CPSO, and EPSO in terms of quality (AR) and execution time (*T*).

These results are visualized in Figs. 8 and 9.

Conclusion

In this study, an improved particle swarm optimization algorithm based on the pattern reduction concept for clustering is used to reduce the time complexity of the PSO-based clustering algorithm. The algorithm uses two detection methods to determine if the pattern can be considered a fixed form. In addition, the multi-starter operator was used to improve the quality of the result. The simulation results show that many calculations during PSO convergence are essentially redundant and can be ignored. The simulation results show that the PSO-based algorithm can significantly reduce the algorithm computation time for clustering problems and achieve better results than the other algorithms introduced in this study. In theory, if M is the population size, N is the number of input patterns, K is the number of clusters, and L is the clustering problem, then the temporal complexity of MPREPSO can be basically reduced from O (MNKLT) to O (MNKL) Dimension, T is the number of iterations. For example, the MPREPSO algorithm can significantly reduce the computation time of the PSO-based algorithm for grouping data sets. Based on the observations, the limitation of the MPREPSO algorithm is that the number of clusters must be specified. Although various cluster algorithms can automatically determine the number of clusters, this is still a limitation of the MPREPSO algorithm.

Funding The authors did not receive support from any organization for the submitted work.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Cheng S, Zhang Q, Qin Q. Big data analytics with swarm intelligence. Ind Manag Data Syst. 2016;116(4):646–66. https://doi.org/10.1108/IMDS-06-2015-0222.
2. Verma H, Verma D, Tiwari PK. A population based hybrid FCM-PSO algorithm for clustering analysis and segmentation of brain image. Expert Syst Appl. 2021;167: 114121. https://doi.org/10.1016/j.eswa.2020.114121.
3. Zhang C, Ouyang D, Ning J. An artificial bee colony approach for clustering. Expert Syst Appl. 2010;37(7):4761–7. https://doi.org/10.1016/j.eswa.2009.11.003.
4. Kuo RJ, Wang MJ, Huang TW. An application of particle swarm optimization algorithm to clustering analysis. Soft Comput. 2011;15(3):533–42. https://doi.org/10.1007/s00500-009-0539-5.
5. Tsai C-W, Huang K-W, Yang C-S, Chiang M-C. A fast particle swarm optimization for clustering. Soft Comput. 2015;19(2):321–38. https://doi.org/10.1007/s00500-014-1255-3.
6. Kogan J. Introduction to clustering large and high-dimensional data. Cambridge: Cambridge University Press; 2007.
7. Bagirov AM, Ugon J, Webb D. Fast modified global k-means algorithm for incremental cluster construction. Pattern Recogn. 2011;44(4):866–76. https://doi.org/10.1016/j.patcog.2010.10.018.
8. Xu R, WunschII D. Survey of clustering algorithms. IEEE Trans Neural Netw. 2005;16(3):645–78. https://doi.org/10.1109/TNN.2005.845141.
9. Lai JZC, Huang T-J, Liaw Y-C. A fast -means clustering algorithm using cluster center displacement. Pattern Recogn.

2009;42(11):2551–6. https://doi.org/10.1016/j.patcog.2009.02.014.
10. Chiang M-C, Tsai C-W, Yang C-S. A time-efficient pattern reduction algorithm for k-means clustering. Inf Sci. 2011;181(4):716–31. https://doi.org/10.1016/j.ins.2010.10.008.
11. van der Merwe DW, Engelbrecht AP. Data clustering using particle swarm optimization. In: The 2003 Congress on Evolutionary Computation, 2003. CEC '03. (Vol. 1, pp. 215–220). IEEE. https://doi.org/10.1109/CEC.2003.1299577
12. Paterlini S, Krink T. Differential evolution and particle swarm optimization in partitional clustering. Comput Stat Data Anal. 2006;50(5):1220–47. https://doi.org/10.1016/j.csda.2004.12.004.
13. Parsopoulos KE, Vrahatis MN. Particle swarm optimization and intelligence: advances and applications: advances and applications. Chennai: IGI Global; 2010.
14. Su S, Zhao S. An optimal clustering mechanism based on Fuzzy-C means for wireless sensor networks. Sustain Comput Inform Syst. 2018;18:127–34. https://doi.org/10.1016/j.suscom.2017.08.001.
15. Ripan RC, Sarker IH, Hossain SMM, Anwar MM, Nowrozy R, Hoque MM, Furhad MH. A data-driven heart disease prediction model through K-means clustering-based anomaly detection. SN Comput Sci. 2021;2(2):112. https://doi.org/10.1007/s42979-021-00518-7.
16. Kaur A, Kaur R, Jagdev G. Analyzing and exploring the impact of big data analytics in sports sector. SN Comput Sci. 2021;2(3):184. https://doi.org/10.1007/s42979-021-00575-y.
17. Sharma M, Chhabra JK. Sustainable automatic data clustering using hybrid PSO algorithm with mutation. Sustain Comput Inform Syst. 2019;23:144–57. https://doi.org/10.1016/j.suscom.2019.07.009.
18. Su Z, Wang P, Shen J, Li Y, Zhang Y, Hu E. Automatic fuzzy partitioning approach using variable string length artificial bee colony (VABC) algorithm. Appl Soft Comput. 2012;12(11):3421–41. https://doi.org/10.1016/j.asoc.2012.06.019.
19. Mitra S, Banka H. Multi-objective evolutionary biclustering of gene expression data. Pattern Recogn. 2006;39(12):2464–77. https://doi.org/10.1016/j.patcog.2006.03.003.
20. Jain AK, Duin PW, Mao Jianchang. Statistical pattern recognition: a review. IEEE Trans Pattern Anal Mach Intell. 2000;22(1):4–37. https://doi.org/10.1109/34.824819.
21. Reddy CK. Data Clustering. In: Aggarwal CC, Reddy CK (eds). Chapman and Hall/CRC 2018. https://doi.org/10.1201/9781315373515
22. Kuo RJ, Potti Y, Zulvia FE. Application of metaheuristic based fuzzy K-modes algorithm to supplier clustering. Comput Ind Eng. 2018;120:298–307. https://doi.org/10.1016/j.cie.2018.04.050.
23. Baskar A. Clustering of Indian districts based on supply chain requirements. Mater Today Proc. 2021;46:9914–9. https://doi.org/10.1016/j.matpr.2021.02.292.
24. Allen TT, Sui Z, Parker NL. Timely decision analysis enabled by efficient social media modeling. Decis Anal. 2017;14(4):250–60. https://doi.org/10.1287/deca.2017.0360.
25. Rose RL, Puranik TG, Mavris DN. Natural language processing based method for clustering and analysis of aviation safety narratives. Aerospace. 2020;7(10):143. https://doi.org/10.3390/aerospace7100143.
26. Tang R, Fong S. Clustering big IoT data by metaheuristic optimized mini-batch and parallel partition-based DGC in Hadoop. Futur Gener Comput Syst. 2018;86:1395–412. https://doi.org/10.1016/j.future.2018.03.006.
27. Masoudi-Sobhanzadeh Y, Jafari B, Parvizpour S, Pourseif MM, Omidi Y. A novel multi-objective metaheuristic algorithm for protein-peptide docking and benchmarking on the LEADS-PEP dataset. Comput Biol Med. 2021;138: 104896. https://doi.org/10.1016/j.compbiomed.2021.104896.
28. Kraus JM, Kestler HA. A highly efficient multi-core algorithm for clustering extremely large datasets. BMC Bioinform. 2010;11(1):169. https://doi.org/10.1186/1471-2105-11-169.
29. Kuo RJ, Zheng YR, Nguyen TPQ. Metaheuristic-based possibilistic fuzzy k-modes algorithms for categorical data clustering. Inf Sci. 2021;557:1–15. https://doi.org/10.1016/j.ins.2020.12.051.
30. Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of ICNN'95—International Conference on Neural Networks (Vol. 4, pp. 1942–1948). 1995. IEEE. https://doi.org/10.1109/ICNN.1995.488968
31. Assareh E, Behrang MA, Assari MR, Ghanbarzadeh A. Application of PSO (particle swarm optimization) and GA (genetic algorithm) techniques on demand estimation of oil in Iran. Energy. 2010;35(12):5223–9. https://doi.org/10.1016/j.energy.2010.07.043.
32. Zhu Z, Zhou J, Ji Z, Shi Y-H. DNA sequence compression using adaptive particle swarm optimization-based memetic algorithm. IEEE Trans Evol Comput. 2011;15(5):643–58. https://doi.org/10.1109/TEVC.2011.2160399.
33. Alam S, Dobbie G, Koh YS, Riddle P, Ur Rehman S. Research on particle swarm optimization based clustering: a systematic review of literature and techniques. Swarm Evol Comput. 2014;17:1–13. https://doi.org/10.1016/j.swevo.2014.02.001.
34. Akbar S, Pardasani KR, Panda NR. PSO based neuro-fuzzy model for secondary structure prediction of protein. Neural Process Lett. 2021;53(6):4593–612. https://doi.org/10.1007/s11063-021-10615-6.
35. Alswaitti M, Albughdadi M, Isa NAM. Density-based particle swarm optimization algorithm for data clustering. Expert Syst Appl. 2018;91:170–86. https://doi.org/10.1016/j.eswa.2017.08.050.
36. Rengasamy S, Murugesan P. PSO based data clustering with a different perception. Swarm Evol Comput. 2021;64: 100895. https://doi.org/10.1016/j.swevo.2021.100895.
37. Malarvizhi K, Amshakala K. Data clustering using hybrid of feature linkage weight based feature reduction and particle Swarm optimization. Mater Today Proc. 2021. https://doi.org/10.1016/j.matpr.2021.01.514.
38. Tarkhaneh O, Isazadeh A, Khamnei HJ. A new hybrid strategy for data clustering using cuckoo search based on Mantegna levy distribution, PSO and k-means. Int J Comput Appl Technol. 2018;58(2):137–49. https://doi.org/10.1504/IJCAT.2018.094576.
39. Liu B, Li J, Lin W, Bai W, Li P, Gao Q. K-PSO: an improved PSO-based container scheduling algorithm for big data applications. Int J Netw Manag. 2021;31(2): e2092. https://doi.org/10.1002/nem.2092.
40. Omran MG, Engelbrecht AP, Salman A. Image classification using particle swarm optimization. In: Recent advances in simulated evolution and learning. Chennai: World Scientific; 2004. p. 347–65.
41. Alguliyev RM, Aliguliyev RM, Sukhostat LV. Parallel batch k-means for Big data clustering. Comput Ind Eng. 2021;152: 107023. https://doi.org/10.1016/j.cie.2020.107023.
42. Hatamlou A, Abdullah S, Nezamabadi-pour H. A combined approach for clustering based on K-means and gravitational search algorithms. Swarm Evol Comput. 2012;6:47–52. https://doi.org/10.1016/j.swevo.2012.02.003.
43. Niknam T, Taherian Fard E, Pourjafarian N, Rousta A. An efficient hybrid algorithm based on modified imperialist competitive algorithm and K-means for data clustering. Eng Appl Artif Intell. 2011;24(2):306–17. https://doi.org/10.1016/j.engappai.2010.10.001.
44. Rana S, Jasola S, Kumar R. A review on particle swarm optimization algorithms and their applications to data clustering. Artif Intell Rev. 2011;35(3):211–22. https://doi.org/10.1007/s10462-010-9191-9.

45. Silva Filho TM, Pimentel BA, Souza RMCR, Oliveira ALI. Hybrid methods for fuzzy clustering based on fuzzy c-means and improved particle swarm optimization. Expert Syst Appl. 2015;42(17–18):6315–28. https://doi.org/10.1016/j.eswa.2015.04.032.
46. Črepinšek M, Liu S-H, Mernik M. Exploration and exploitation in evolutionary algorithms. ACM Comput Surv. 2013;45(3):1–33. https://doi.org/10.1145/2480741.2480752.
47. Lee YL, El-Saleh AA, Ismail M. Gravity-based particle swarm optimization with hybrid cooperative swarm approach for global optimization. J Intell Fuzzy Syst. 2014;26(1):465–81. https://doi.org/10.3233/IFS-130872.
48. Pei S, Tong L. Gaussian kernel particle swarm optimization clustering algorithm. In: 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), 2016; 198–204. https://doi.org/10.1109/FSKD.2016.7603174
49. Elkan C. Using the triangle inequality to accelerate k-means. In: Proceedings of the 20th international conference on Machine Learning (ICML-03) 2003; pp. 147–153.
50. Lu Y, Lu S, Fotouhi F, Deng Y, Brown SJ. FGKA. In: Proceedings of the 2004 ACM symposium on Applied computing—SAC '04 (p. 622). New York, New York, USA: ACM Press 2004. https://doi.org/10.1145/967900.968029
51. Amiri B, Hossain L, Mosavi SE. Application of harmony search algorithm on clustering. In: Proceedings of the world congress on engineering and computer science (Vol. 1, pp. 20–22) 2010.
52. Maulik U, Bandyopadhyay S. Genetic algorithm-based clustering technique. Pattern Recogn. 2000;33(9):1455–65. https://doi.org/10.1016/S0031-3203(99)00137-5.
53. Bandyopadhyay S, Maulik U. Genetic clustering for automatic evolution of clusters and application to image classification. Pattern Recogn. 2002;35(6):1197–208. https://doi.org/10.1016/S0031-3203(01)00108-X.
54. Sung CS, Jin HW. A tabu-search-based heuristic for clustering. Pattern Recogn. 2000;33(5):849–58. https://doi.org/10.1016/S0031-3203(99)00090-4.
55. Shelokar P, Jayaraman V, Kulkarni B. An ant colony approach for clustering. Anal Chim Acta. 2004;509(2):187–95. https://doi.org/10.1016/j.aca.2003.12.032.
56. Fathian M, Amiri B. A honeybee-mating approach for cluster analysis. Int J Adv Manuf Technol. 2008;38(7–8):809–21. https://doi.org/10.1007/s00170-007-1132-7.
57. Niknam T, Olamaei J, Amiri B. A hybrid evolutionary algorithm based on ACO and SA for cluster analysis. J Appl Sci. 2008;8(15):2695–702. https://doi.org/10.3923/jas.2008.2695.2702.
58. Jarboui B, Cheikh M, Siarry P, Rebai A. Combinatorial particle swarm optimization (CPSO) for partitional clustering problem. Appl Math Comput. 2007;192(2):337–45. https://doi.org/10.1016/j.amc.2007.03.010.
59. Miranda V, Fonseca N. EPSO—best-of-two-worlds meta-heuristic applied to power system problems. In: Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600) (Vol. 2, pp. 1080–1085). IEEE 2002. https://doi.org/10.1109/CEC.2002.1004393.
60. Bratton D, Kennedy J. Defining a Standard for Particle Swarm Optimization. In: 2007 IEEE Swarm Intelligence Symposium (pp. 120–127). 2007; IEEE. https://doi.org/10.1109/SIS.2007.368035.
61. UCL. Dataset. 2002. Retrieved from https://archive.ics.uci.edu/ml/datasets.php.