PLOS | ONE

RESEARCH ARTICLE

Automatic Clustering Using Multi-objective Particle Swarm and Simulated Annealing

Ahmad Abubaker[1,2]*, Adam Baharum[1], Mahmoud Alrefaei[3]

1 School of Mathematical Sciences, University Sains Malaysia, 11800 USM Penang, Malaysia, **2** Department of Mathematics & Statistics, Al-Imam Muhammad Ibn Saud Islamic University, P.O. Box 90950, 11623 Riyadh, Saudi Arabia, **3** Departments of Mathematics & Statistics, Jordan University of Science and Technology, Irbid 22110, Jordan

* ahm.abubaker@gmail.com

Abstract

This paper puts forward a new automatic clustering algorithm based on Multi-Objective Particle Swarm Optimization and Simulated Annealing, "MOPSOSA". The proposed algorithm is capable of automatic clustering which is appropriate for partitioning datasets to a suitable number of clusters. MOPSOSA combines the features of the multi-objective based particle swarm optimization (PSO) and the Multi-Objective Simulated Annealing (MOSA). Three cluster validity indices were optimized simultaneously to establish the suitable number of clusters and the appropriate clustering for a dataset. The first cluster validity index is centred on Euclidean distance, the second on the point symmetry distance, and the last cluster validity index is based on short distance. A number of algorithms have been compared with the MOPSOSA algorithm in resolving clustering problems by determining the actual number of clusters and optimal clustering. Computational experiments were carried out to study fourteen artificial and five real life datasets.

Citation: Abubaker A, Baharum A, Alrefaei M (2015) Automatic Clustering Using Multi-objective Particle Swarm and Simulated Annealing. PLoS ONE 10(7): e0130995. doi:10.1371/journal.pone.0130995

Editor: Yong Deng, Southwest University, CHINA

Received: December 7, 2014

Accepted: May 27, 2015

Published: July 1, 2015

Data Availability Statement: All relevant data are within the paper and its Supporting Information files.

Funding: The authors have no support or funding to report.

Competing Interests: The authors have declared that no competing interests exist.

Introduction

Data clustering is an important task in the field of unsupervised datasets. The clustering technique distributes the dataset into clusters of similar features [1]. To solve a clustering problem, the number of clusters that fits a dataset must be determined, and the objects for these clusters must be assigned appropriately. The number of clusters may or may not be known, thereby making it difficult to find the best solution to the clustering problem. As such, the clustering problem can be viewed as an optimization problem. This challenge has led to the proposal of many automatic clustering algorithms in previous literature; these algorithms estimate the appropriate number of clusters and appropriately partition a dataset into these clusters without the need to know the actual number of clusters [2–8]. Most of these algorithms rely exclusively on one internal evaluation function (validity index). The validity index has an objective function to evaluate the various characteristics of clusters, which illustrates the clustering quality and accuracy of the clustering solutions [9]. Nevertheless, the single evaluation function is often ineligible to determine the appropriate clusters for a dataset, thus giving an inferior

solution [10]. Accordingly, the clustering problem is structured as a multi-objective optimization problem wherein different validity indices can be applied and evaluated simultaneously.

Several automatic multi-objective clustering algorithms are proposed in literature to solve the clustering problem. Evolution appeared in this area after Handl and Knowles [3] proposed an evolutionary approach called multi-objective clustering with automatic K determination (MOCK). For some of the automatic multi-objective clustering algorithms related to MOCK, can refer to [11–13]. A multi-objective clustering technique inspired by MOCK named VAMOSA, which is based on simulated annealing as the underlying optimization strategy and the point symmetry-based distance, was proposed by Saha and Bandyopadhyay [5].

How to deal with various shapes of datasets (hyper spheres, linear, spiral, convex, and non-convex), overlapping datasets, datasets with a small or large number of clusters, and datasets that have objects with small or large dimensions without providing the proper clustering or knowing the cluster number is a challenge. Saha and Bandyopadhyay [8] developed two multi-objective clustering techniques (GenClustMOO and GenClustPESA2) by using a simulated annealing-based multi-objective optimization technique and the concept of multiple centers to each cluster that can deal with different types of cluster structures. GenClustMOO and GenClustPESA2 were compared with MOCK [3], VGAPS [4], K-means (KM) [14], and single-linkage clustering technique (SL) [15] using numerous artificial and real-life datasets of diverse complexities. However, these algorithms did not give the desired high accuracy in clustering datasets.

The current study proposes an automatic clustering algorithm, namely, hybrid multi-objective particle swarm optimization with simulated annealing (MOPSOSA), which deals with different sizes, shapes, and dimensions of datasets and an unknown number of clusters. The Numerical results of the proposed algorithm are shown to perform better than those of the GenClustMOO [8] and GenClustPESA2 [8] methods in terms of clustering accuracy (see the Results and Discussions Section). In order to deal with any dataset and qualification to determine appropriate clusters and obtain good solutions with high accuracy, combinatorial particle swarm optimization II [7]is developed to deal with three different cluster validity indices, simultaneously. The first cluster validity index is the Davies-Bouldin index (*DB*-index) [16], which is based on Euclidean distance; the second one is symmetry-based cluster validity index (*Sym*-index) [4], which is based on point symmetry distance; and the last one is a connectivity-based cluster validity index (*Conn*-index) [17], which is based on short distance. If no change exists in a particle position or when it is moved to a bad position, then the MOPSOSA algorithm uses MOSA [18] to improve the searching particle. The MOPSOSA algorithm also utilizes KM method [14] to improve the selection of the initial particle position because of its significance in the overall performance of the search process. It creates a large number of Pareto optimal solutions through a trade-off between the three different validity indices. Therefore, the idea of sharing fitness [19] is incorporated in the proposed algorithm to maintain diversity in the repository that contains Pareto optimal solutions. Pareto optimal solutions are important for decision makers to choose from. Furthermore, to comply with the decision-maker requirements, the proposed algorithm utilizes a semi-supervised method [20] to provide a single best solution from the Pareto set. The performance of MOPSOSA is compared with the performances of three automatic multi-objective clustering techniques, namely, GenClustMOO [8], GenClustPESA2 [8], and MOCK [3], and with those of three single-objective clustering techniques, namely, VGAPS [4], KM [14], and SL [15], using 14 artificial and 5 real-life datasets.

The reminder of this paper is structured as follows; Section 2 describes the multi-objective clustering problem; Section 3 illustrates the proposed MOPSOSA algorithm in details; Section 4 presents the datasets used in the numerical experiments, the evaluation of clustering quality, and the setting of the parameters for the MOPSOSA algorithm; Section 5 includes discussion of the results; Finally, concluding remarks are given in Section 6.

Clustering Problem

The clustering problem is defined as follows: Consider the dataset $P = \{p_1,p_2,\ldots,p_n\}$, where $p_i = (p_{i1},p_{i2},\ldots,p_{id})$ is a feature vector of d-dimensions and also referred to as the object, p_{ij} is the feature value of object i at dimension j, and n is the number of objects in P. The clustering of P is the partitioning of P into k clusters $\{C_1,C_2,\ldots,C_k\}$ with the following properties:

$$\bigcup_{i=1}^{k} C_i = P \tag{1}$$

$$C_i \cap C_j = \phi,\ i \neq j,\ i = 1,2,\ldots,k,\ j = 1,2,\ldots,k \tag{2}$$

$$C_i \neq \phi,\ i = 1,2,\ldots,k \tag{3}$$

The clustering optimization problem with one objective function for the clustering problem can be formed as follows: $\min_{C\in\Theta}/\max\ f(C)$ such that Eqs (1) to (3) are satisfied, where f is the validity index function, Θ is the feasible solutions set that contains all possible clustering for the dataset P of n objects into k clusters, $C = \{C_1,C_2,\ldots,C_k\}$ and k = 2,3,. . .,n−1.

The multi-objective clustering problem for S different validity indices is defined as follows:

$$\min_{C\in\Theta}\ F(C) = [f_1(C),f_2(C),\ldots,f_S(C)]. \tag{4}$$

where $F(C)$ is a vector of S validity indices. Note that there may be no solution that minimizes all the functions $f_i(C)$. Therefore, the aim is to identify the set of all non-dominant solutions.

Definition: Consider C and C^* as two solutions in the feasible solutions set Θ, the solution C is said to be dominated by the solution C^* if and only if $f_i(C^*) \leq f_i(C)$, $\forall\ i = 1,\ldots,S$ and $f_i(C^*) < f_i(C)$ for at least one i. Otherwise, C is said to be non-dominated by C^*.

The Pareto optimal set is a set that includes all non-dominated solutions in the feasible solutions set Θ.

The Proposed MOPSOSA Algorithm

Simulated annealing requires more calculation time than does particle swarm optimization [21]. The former requires low variations of temperature parameters to obtain a global solution [22]. Some of the particles may become stagnant and remain unchanged, especially when the objective functions of the best personal position and the best global position are similar [21]. As such, the particle cannot jump out, which in turn causes convergence toward the local solution and the loss of its capability to search for the optimal Pareto set. This phenomenon is a disadvantage in comparison with simulated annealing, which can jump away from a local solution. The proposed MOPSOSA algorithm, as previously mentioned, is a hybrid algorithm that merges the advantages of fast calculation and convergence in particle swarm optimization with the capability to evade local solutions in simulated annealing.

The clustering solution X_i is described using label-based integer encoding [23]. Each particle position is a clustering solution. The particle position X_i^t and velocity V_i^t are presented as vectors with n components $X_i^t = (X_{i1}^t, X_{i2}^t, \ldots, X_{in}^t)$ and $V_i^t = (V_{i1}^t, V_{i2}^t, \ldots, V_{in}^t)$ at time t, $i = 1,\ldots,m$, where n is the number of data objects, and m is the number of particles (swarm size). The position component $X_{ij}^t \in \{1,\ldots,K_i^t\}$ represents the cluster number of j^{th} object in i^{th} particle, and $V_{ij}^t \in \{0,\ldots,K_i^t\}$ represents the motion of j^{th} object in i^{th} particle, where $K_i^t \in \{K_{\min},\ldots,K_{\max}\}$ is the number of clusters related to particle i at time t (where $K_{\min}$ and $K_{\max}$ are the minimum and maximum number of clusters, respectively; the default value of $K_{\min}$ is 2; and $K_{\max}$ is $\sqrt{n}+1$ unless it is manually specified) [24]. The best previous position of i^{th}

particle at iteration t is represented as $XP_i^t = (XP_{i1}^t, XP_{i2}^t, \ldots, XP_{in}^t)$. The leader position chosen from the repository of Pareto sets for i^{th} particle at iteration t is represented by $XG_i^t = (GP_{i1}^t, GP_{i2}^t, \ldots, GP_{in}^t)$.

The flowchart in Fig 1 illustrates the general process of the MOPSOSA algorithm. The process of the algorithm is described in the following 11 steps:

Step 1: The algorithm parameters, such as swarm size m, number of iterations *Iter*, maximum and minimum numbers of clusters, velocity parameters, initial cooling temperature T_0, and $t = 0$, are initialized.

Step 2: The initial particle position X_i^t using KM method [14], initial velocity $V_i^t = 0$, and initial $XP_i^t = X_i^t$, $i = 1,\ldots,m$ are generated.

Step 3: The objective functions $f_1(X_i^t), \ldots, f_S(X_i^t)$, $i = 1,\ldots,m$, where S is the number of objective functions, are computed. The repository of Pareto sets is filled with all non-dominated XP_i^t, $i = 1,\ldots,m$ based on a fitness-sharing basis.

Step 4: The leader XG_i^t from the repository of Pareto sets nearest to current X_i^t is selected. The clusters in XP_i^t and XG_i^t are renumbered on the basis of their similarity to the clusters in X_i^t, $i = 1,\ldots,m$.

Step 5: The new $Vnew_i$ and $Xnew_i$, $i = 1,\ldots,m$, are computed using XG_i^t, XP_i^t, X_i^t, and V_i^t.

Step 6: The validity of $Xnew_i$, $i = 1,\ldots,m$ is checked, and the correction process is applied if it is not valid.

Step 7: The objective functions $f_1(Xnew_i),\ldots,f_s(Xnew_i)$ and $f_1(X_i^t), \ldots, f_s(X_i^t)$, $i = 1,\ldots,m$ are computed.

Step 8: A dominance check for $Xnew_i$, $i = 1,\ldots,m$ is performed, that is, if $Xnew_i$ is non-dominated by X_i^t, then $X_i^{t+1} = Xnew_i$ and $V_i^{t+1} = Vnew_i$; otherwise, the MOSA technique is applied and $X_i^{t+1} = X_i^{MOSA}$ and $V_i^{t+1} = V_i^{MOSA}$, $i = 1,\ldots,m$, where X_i^{MOSA} and V_i^{MOSA} are the position and velocity particles respectively obtained by applying the MOSA technique. The MOSA is discussed in details in section MOSA Technique below. Upon completion of the generation of new positions for all particles, the cooling temperature T_{t+1} is updated.

Step 9: The new XP_i^{t+1}, $i = 1,\ldots,m$ is identified.

Step 10: The Pareto set repository is updated.

Step 11: $t = t + 1$ is set; if $t \geq Iter$, then the algorithm is stopped and the Pareto set repository contains the Pareto solutions; otherwise, go to step 4.

The following sections will elucidate the steps of the MOPSOSA algorithm.

Particles swarm initialization

Initial particles are generally considered one of the success factors in particle swarm optimization that affect the quality of the solution and the speed of convergence. Hence, the MOPSOSA algorithm employs KM method as a means to improve the generation of the initial swarm of particles. Fig 2 depicts a flowchart for the generation of m particles. Starting with $i = 1$ and $W = \min\{K_{max}-K_{min}+1,m\}$, if $W = m$, then m particles will be generated by KM method with the number of clusters $K_i = K_{min}+i-1$, $i = 1,\ldots,m$. If $W = K_{max}-K_{min}+1$, then the first W particles will be generated by KM with the number of clusters $K_i = K_{min}+i-1$, $i = 1,\ldots,W$, and the other particle will be generated by KM with the number of clusters K_i, $i = W+1,\ldots m$ selected

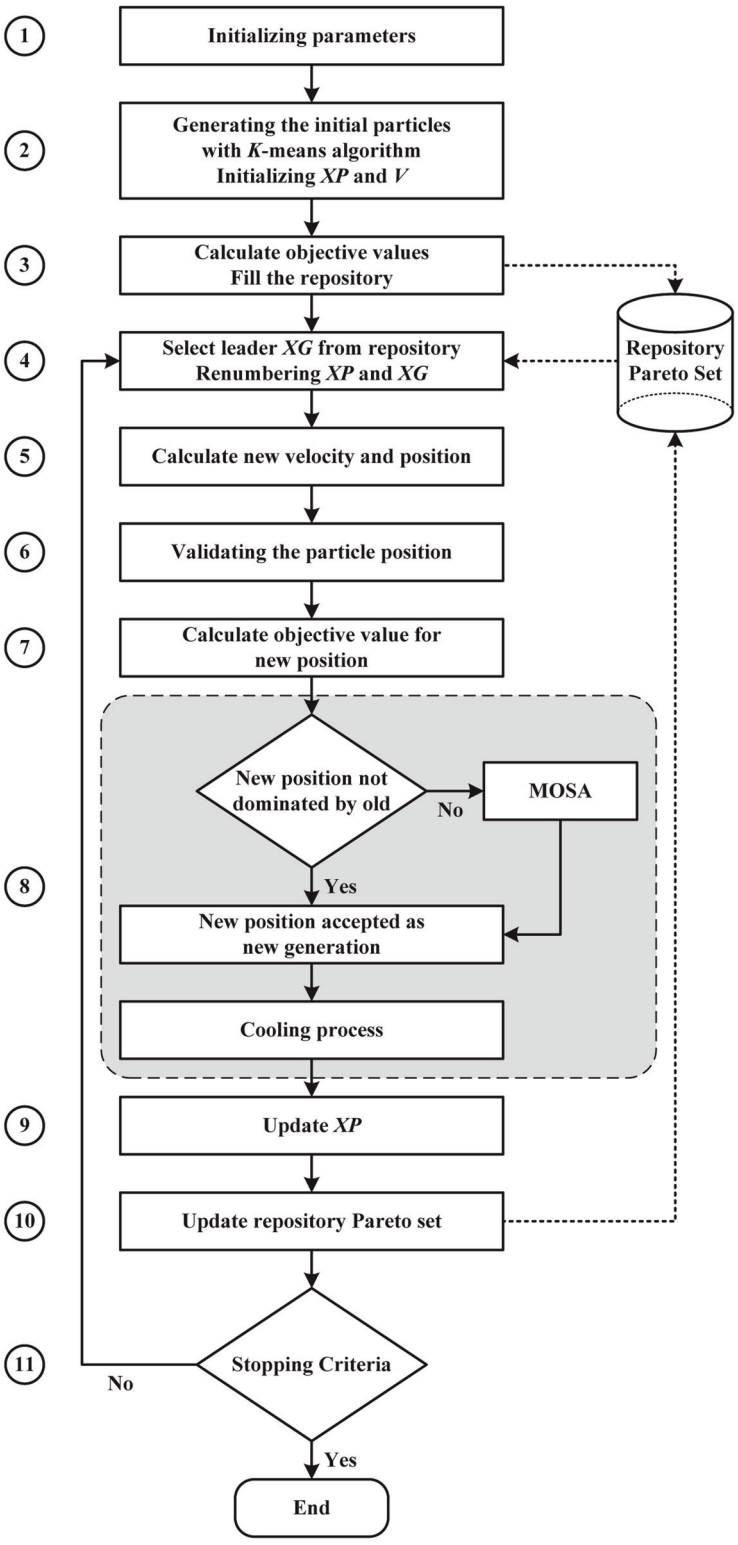

Fig 1. Flowchart for the proposed MOPSOSA algorithm.

doi:10.1371/journal.pone.0130995.g001

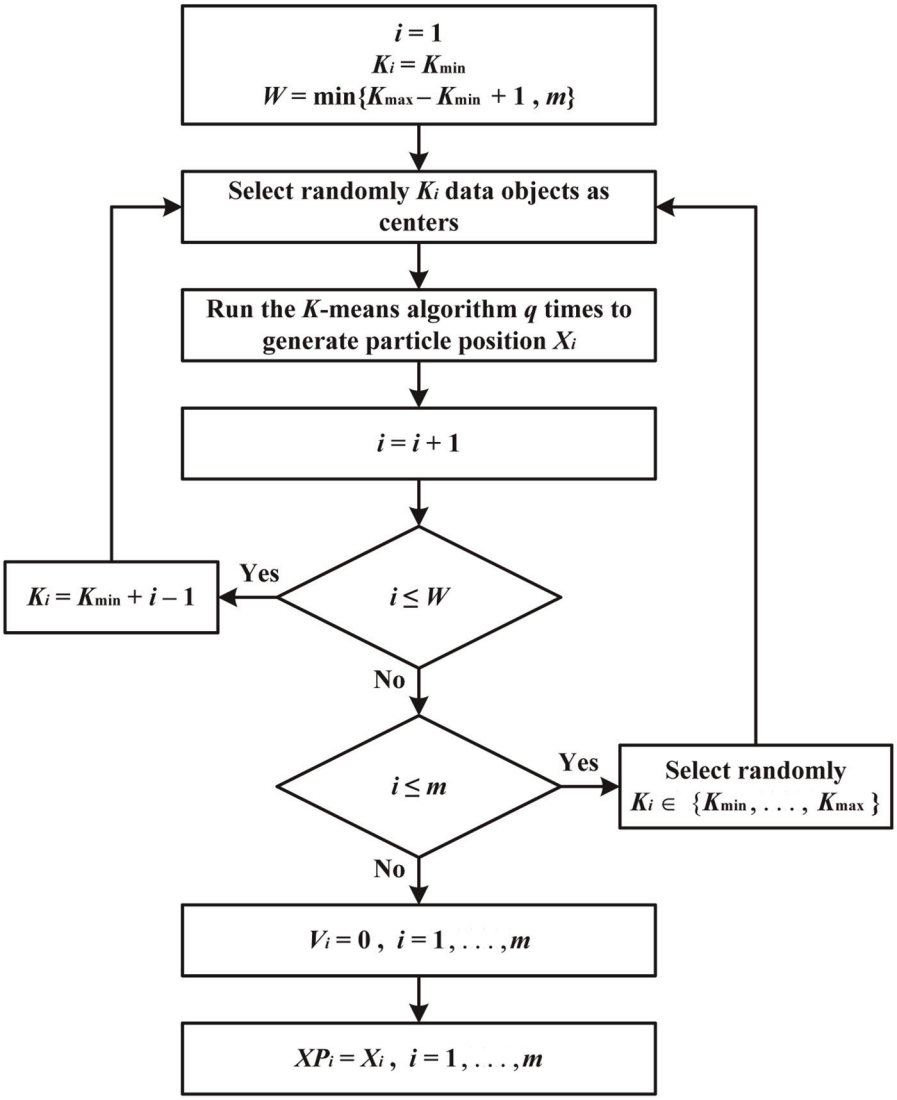

Fig 2. Flowchart for initializing particle swarm.

doi:10.1371/journal.pone.0130995.g002

randomly between K_{min} and K_{max}. For each particle, the initial velocities are selected to be zero $V_i = 0$, $i = 1,\ldots,m$, and the initial XP_i is equal to the current position X_i for all $i = 1,\ldots,m$.

Objective functions

The proposed algorithm uses three types of cluster validity indices as objective functions to achieve optimization. These validity indices, *DB*-index, *Sym*-index, and *Conn*-index, apply three different distances, namely, Euclidean distance, point symmetric distance, and short distance, respectively. Each validity index indicates a different aspect of good solutions in clustering problems. These validity indices are described below.

DB-index. This index was developed by Davies—Bouldin [16] which is a function of the ratio of the sum of within-cluster objects (intra-cluster distance) and between cluster separation (inter-cluster distance). The within i^{th} cluster C_i, $S_{i,q}$ is calculated using Eq (5). The

distance between clusters C_i and C_j is denoted by $d_{ij,t}$, which is computed using Eq (6).

$$S_{i,q} = \left(\frac{1}{n_i} \sum_{p \in C_i} \|p - c_i\|_2^q \right)^{\frac{1}{q}} \tag{5}$$

$$d_{ij,t} = \|c_i - c_j\|_t \tag{6}$$

where $n_i = |C_i|$ is the number of objects in cluster C_i, c_i is the cluster center of cluster C_i and is defined as $c_i = \frac{1}{n_i} \sum_{p \in C_i} p$, and q and t are positive integer numbers. DB is defined as:

$$DB = \frac{1}{k} \sum_{i=1}^{k} R_{i,qt} \tag{7}$$

where $R_{i,qt} = \max_{j, j \neq i} \left\{ \frac{S_{i,q} + S_{j,q}}{d_{ij,t}} \right\}$. A small value of DB means a good clustering result.

Sym-index. The recently developed point symmetry distance $d_{ps}(p,c)$ is employed in this cluster validity index *Sym*, which measures the overall average symmetry in connection with the cluster centers [4]. It is defined as follows. Let p be a point, and the reflected symmetrical point of p with respect to a specific center c is $2c - p$ and is denoted by p^*. Let *knear* unique nearest neighbors to p^* be at the Euclidean distances of d_i, $i = 1,\ldots,knear$. The point symmetric distance is defined as:

$$d_{ps}(p,c) = d_{sym}(p,c) \times d_e(p,c) = \frac{\sum_{i=1}^{knear} d_i}{knear} \times d_e(p,c) \tag{8}$$

where $d_e(p,c)$ is the Euclidean distance between the point p and the center c and $d_{sym}(p,c)$ is a symmetric measure of p with respect to c, which is defined as $\sum_{i=1}^{knear} d_i / knear$. In this study, $knear = 2$. The cluster validity function is defined as

$$Sym = \left(\frac{1}{k} \times \frac{1}{\varepsilon_k} \times D_k \right) \tag{9}$$

where $\varepsilon_k = \sum_{i=1}^{k} E_i$, $E_i = \sum_{j=1}^{n_i} d_{ps}^*(p_j^i, c_i)$, p_j^i is the j^{th} object of cluster i, and $D_k = \max_{i,j=1}^{k} \|c_i - c_j\|$ is the maximum Euclidean distance between the two centers among all cluster pairs. Eq (8) is used with some constraint to compute $d_{ps}^*(p_j^i, c_i)$. The *knear* nearest neighbors of p_j^* and p_j^i should belong to the i^{th} cluster, where p_j^* is the reflected point of the point p_j^i with respect to c_i. A large value for *Sym*-index means that the actual number of clusters and proper partitioning are obtained.

***Conn*-index.** The third cluster validity index used in this study is proposed by Saha and Bandyopadhyay [17], it depends on the notion of cluster connectedness. To compute *Conn*-index, the the relative neighborhood graph [25] structuring for the dataset has to be conducted first. Subsequently, the short distance between two points x and y is denoted by $d_{short}(x,y)$ and is defined as follows:

$$d_{short}(x,y) = \min_{i=1}^{npath} \max_{j=1}^{ned_i} w(ed_j^i) \tag{10}$$

where *npath* is the number of all paths between x and y in the RNG structuring; ned_i is the number of edges along i^{th} path, $i = 1,\ldots,npath$; ed_j^i is j^{th} edge in i^{th} path, $j = 1,\ldots,ned_i$ and $i = 1,\ldots,$

npath; and $w(ed_j^i)$ is the edge weight of the edge ed_j^i. The edge weight $w(ed_j^i)$ is equal to the Euclidean distance between a and b, $d_e(a,b)$, where a and b are the end points of the edge ed_j^i.

The cluster validity index *Conn* developed by Saha and Bandyopadhyay [17] is defined as follows:

$$Conn = \frac{\sum_{i=1}^{k}\sum_{j=1}^{n_i} d_{short}(p_j^i, m_i)}{n\left(\min_{i,j=1,\ i\neq j}^{k} d_{short}(m_j, m_i)\right)} \tag{11}$$

where m_i is the medoid of the i^{th} cluster that is equal to the point with the minimum average distance to all points in the i^{th} cluster $m_i = p^i_{\text{minindex}}$, and $\text{minindex} = \arg\min_{t=1}^{n_i}\left(\sum_{j=1}^{n_i} d_e(p_t^i, p_j^i)/n_i\right)$. The minimum value of *Conn*-index means the clusters interconnected internally and separately from each other.

After the particles have been moved to a new position, the three objective functions are computed for each particle in the swarm. The objective functions for a particle position X are $\{DB(X), 1/Sym(X), Conn(X)\}$. The three objectives are minimized simultaneously using MOPSOSA algorithm.

XP updating

The previous best position of i^{th} particle at iteration t is updated by non-dominant criteria. XP_i^t is compared with the new position X_i^{t+1}. Three cases of this comparison are considered.

- If XP_i^t is dominated by X_i^{t+1}, then $XP_i^{t+1} = X_i^{t+1}$.
- If X_i^{t+1} is dominated by XP_i^t, then $XP_i^{t+1} = XP_i^t$.
- If XP_i^t and X_i^{t+1} are non-dominated, then one of them will be chosen randomly as XP_i^{t+1}.

This update occurs on each particle.

Repository updating

The repository is utilized as a guide by MOPSOSA algorithm for the swarm toward the Pareto front. The non-dominated particle positions are stored in the repository. To preserve the diversity of non-dominated solutions in the repository, sharing fitness [19] is a good method to control the acceptance of new entries into the repository when it is full. Fitness sharing was used by Lechuga and Rowe [26] in multi-objective particle swarm optimization. In each iteration, the new non-dominated solutions are added into the external repository and elimination of the dominated solutions. In case the non-dominated solutions are increased than the size of the repository, the fitness sharing is calculated for all non-dominated solutions. The solutions that have largest values of fitness sharing are selected to fill the repository.

Cluster re-numbering

The re-numbering process is designed to eliminate the redundant particles that represent the same solution. The proposed MOPSOSA algorithm employs the re-numbering procedure designed by Masoud et al. [7]. This procedure uses a similarity function to measure the degree of similarity between the clusters of two input solutions X_i^t and XP_i^t (or XG_i^t). The two clusters

that are most similar are matched. Any cluster in XP_i^t (or XG_i^t) not matched to any cluster X_i^t will use the unused number in the clustering numbering. MOPSOSA algorithm uses the similarity function known as Jaccard coefficient [27], which is defined as follows:

$$Sim(C_j, \widehat{C_k}) = \frac{n_{11}}{n_{11} + n_{10} + n_{01}} \tag{12}$$

where C_j is j^{th} cluster in X_i^t, $\widehat{C_k}$ is k^{th} cluster in XP_i^t, n_{11} is the number of objects that exist in both C_j and $\widehat{C_k}$, n_{10} is the number of objects that exist in C_j but does not exist in $\widehat{C_k}$, and n_{01} is the number of objects that do not exist in C_j but exist in $\widehat{C_k}$.

Velocity computation

MOPSOSA algorithm employs the expressions and operators modified by Masoud et al. [7]. The new velocity for particle i at iteration t is calculated as follows:

$$V_i^{t+1} = (W \otimes V_i^t) \oplus ((R_1 \otimes (XP_i^t \ominus X_i^t)) \oplus (R_2 \otimes (XG_i^t \ominus X_i^t))) \tag{13}$$

where W, R_1, and R_2 are the vectors of n components with values 0 or 1 that are generated randomly with a probability of w, r_1, and r_2, respectively. The operations $\otimes$, $\oplus$, and $\ominus$ are the multiplication, merging, and difference, respectively.

- **Difference operator$\ominus$**

The difference operation calculates the difference between X_i^t and XP_i^t (or XG^t). Let $\lambda P_i^t = (\lambda p_{i1}^t, \ldots, \lambda p_{in}^t) = XP_i^t \ominus X_i^t$, and $\lambda G_i^t = (\lambda g_{i1}^t, \ldots, \lambda g_{in}^t) = XG^t \ominus X_i^t$ be defined as follows:

$$\lambda p_{ij}^t = \begin{cases} XP_{ij}^t & \text{if } X_{ij}^t \neq XP_{ij}^t \\ 0 & \text{otherwise} \end{cases} \tag{14}$$

$$\lambda g_{ij}^t = \begin{cases} XG_j^t & \text{if } X_{ij}^t \neq XG_j^t \\ 0 & \text{otherwise} \end{cases} \tag{15}$$

- **Multiplication operator $\otimes$**

The multiplication operator is defined as follows: let $A = (a_1,\ldots,a_n)$ and $B = (b_1,\ldots,b_n)$ are two vectors of n components, then $A \otimes B = (a_1 b_1,\ldots,a_n b_n)$.

- **Merging operator $\oplus$**

The merging operator is defined as follows: let $A = (a_1,\ldots,a_n)$ and $B = (b_1,\ldots,b_n)$ be two vectors of n components, then $C = A \oplus B = (c_1,c_2,\ldots,c_n)$, where

$$c_i = \begin{cases} a_i & \text{if } a_i \neq 0 \text{ and } b_i = 0 \\ b_i & \text{if } a_i = 0 \text{ and } b_i \neq 0 \\ a_i \text{ or } b_i \text{ randomly} & \text{if } a_i \neq 0 \text{ and } b_i \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

Position computation

MOPSOSA algorithm employs the definition to generate new positions, as proposed by Masoud et al. [7]. The new position is generated from the velocity as follows:

$$X_{ij}^{t+1} = \begin{cases} V_{ij}^{t} & \text{if } V_{ij}^{t+1} \neq 0 \\ r & \text{otherwise} \end{cases} \tag{17}$$

where r is an integer random number in $[1, K_i^t + 1]$ and $K_i^t + 1 < K_{\max}$. This property enables the particle to add new clusters. The previous operators and the differences in cluster number of X_i^t, XP_i^t, and XG^t lead to the addition or removal of some of the clusters in the output of the new position X_i^{t+1}. Sometimes an empty cluster may exist, which leads to invalid particle position. Such an instance can be avoided by exposing the particle to reset the numbering clusters. The re-numbering process works by encoding the largest cluster number to the smallest unused one.

MOSA technique

MOSA method [18] is applied in the MOPSOSA algorithm at iteration t for particle i in case X_i^t dominates the new position $Xnew_i$. Fig 3 presents the flowchart for the MOSA technique applied in MOPSOSA. The procedure for the MOSA technique is explained in eight steps below.

1. Step 1: Let *PSX* and *PSV* be two empty sets, *niter* is a maximum number of iteration, and $q = 0$.
2. Step 2: Evaluate $EXP_q = \prod_{j=1}^{s} \exp(-[f_j(Xnew_i) - f_j(X_i^t)]^+ / T_t)$, where the cooling temperature T_t is updated in step 8 of MOPSOSA algorithm. Generate uniform random number $u \in (0,1)$, if $u < EXP_q$, go to step 7. Otherwise, proceed to the next step.
3. Step 3: Add $Xnew_i$ to *PSX* and $Vnew_i$ to *PSV*, then *PSX* and *PSV* are updated to include only non-dominant solutions.
4. Step 4: If $q \geq niter$, then choose a solution randomly from *PSX* as the new particle position $Xnew_i$ and the corresponding velocity $Vnew_i$ from *PSV*, and proceed to step 7. Otherwise, $q = q+1$, and generate the new velocity $Vnew_i$ and position $Xnew_i$ from the old position X_i^t.
5. Step 5: Calculate the objective function $f_1(Xnew_i), \ldots, f_s(Xnew_i)$, and $f_1(X_i^t), \ldots, f_S(X_i^t)$.
6. Step 6: Perform a dominance check for $Xnew_i$, if $Xnew_i$ is non-dominated by X_i^t, then proceed to step 7. Otherwise go to step 2.
7. Step 7: The new position and velocity $Xnew_i$ and $Vnew_i$ are accepted as the new generation of X_i^{MOSA} and V_i^{MOSA}, respectively, $X_i^{MOSA} = Xnew_i$ and $V_i^{MOSA} = Vnew_i$.
8. Step 8: Check the validity for X_i^{MOSA}, and apply the re-numbering process if it is invalid. Return X_i^{MOSA} and V_i^{MOSA}.

Selection of the best solution

In general, a Pareto set containing several non-dominated solutions is provided on the final run of multi-objective problems [28]. Each non-dominated solution introduces a pattern of clustering for the given dataset. The semi-supervised method proposed by Saha and

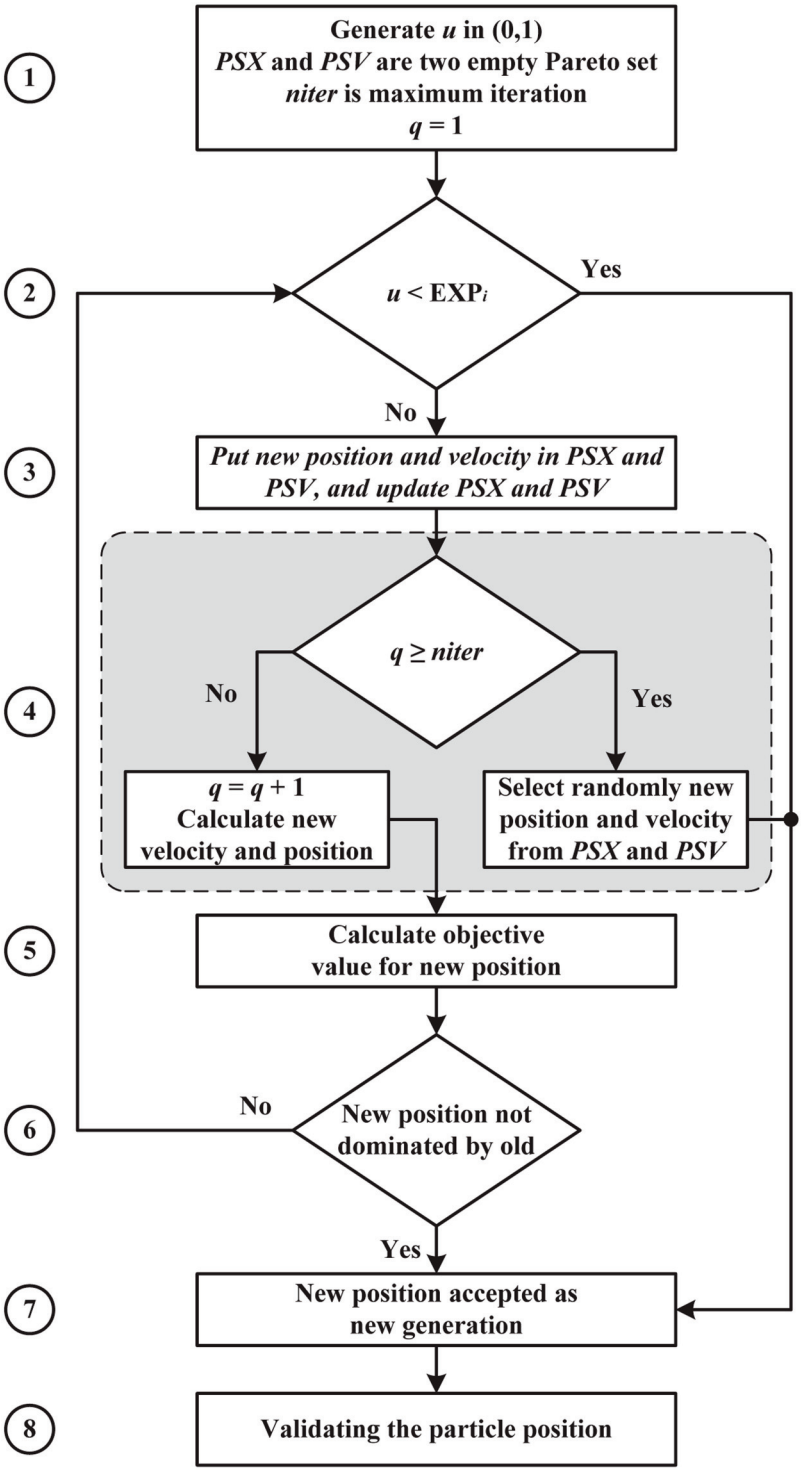

Fig 3. Flowchart for the MOSA technique applied in MOPSOSA.

doi:10.1371/journal.pone.0130995.g003

Bandyopadhyay [20] is utilized in the MOPSOSA algorithm to select the best solution from the Pareto optimal set. This semi-supervised approach can only be applied when the cluster labels of some points in the dataset are known. The misclassification value is computed by using the Minkowski score *MS* [29]. Let *T* be the actual solution and *C* be the selected solution; hence, *MS* is defined as follows:

$$MS(T, C) = \sqrt{\frac{n_{01} + n_{10}}{n_{11} + n_{10}}} \tag{18}$$

The low values of *MS* are "better" with the optimal value for *MS* set as 0.

Experimental Study

This section presents the datasets used for the experiment, the measurement of the accuracy solution, and parameters settings of the proposed algorithm.

Experimental datasets

The MOPSOSA algorithm is examined on 14 artificial and 5 real-life datasets (S1 File). Table 1 displays the types of datasets, the number of points (objects), the dimensions (features), and the number of clusters. Further details on these datasets are provided below.

- **Artificial datasets**

1. Sph_5_2 [2] dataset (Appendix A in S1 File): This dataset consists of 250-point 2D distributed over five overlapping spherically shaped clusters. Each cluster contains 50 points. Fig 4a illustrates this dataset.

Table 1. Description of the artificial and real-life datasets.

Dataset	# Points	Dimension	# Clusters
Sph_5_2	250	2	5
Sph_4_3	400	3	4
Sph_6_2	300	2	6
Sph_10_2	500	2	10
Sph_9_2	900	2	9
Pat1	557	2	3
Pat2	417	2	2
Long1	1000	2	2
Sizes5	1000	2	4
Spiral	1000	2	2
Square1	1000	2	4
Square4	1000	2	4
Twenty	1000	2	20
Fourty	1000	2	40
Iris	150	4	3
Cancer	683	9	2
Newthyroid	215	5	3
LiverDisorder	345	6	2
Glass	214	9	6

doi:10.1371/journal.pone.0130995.t001

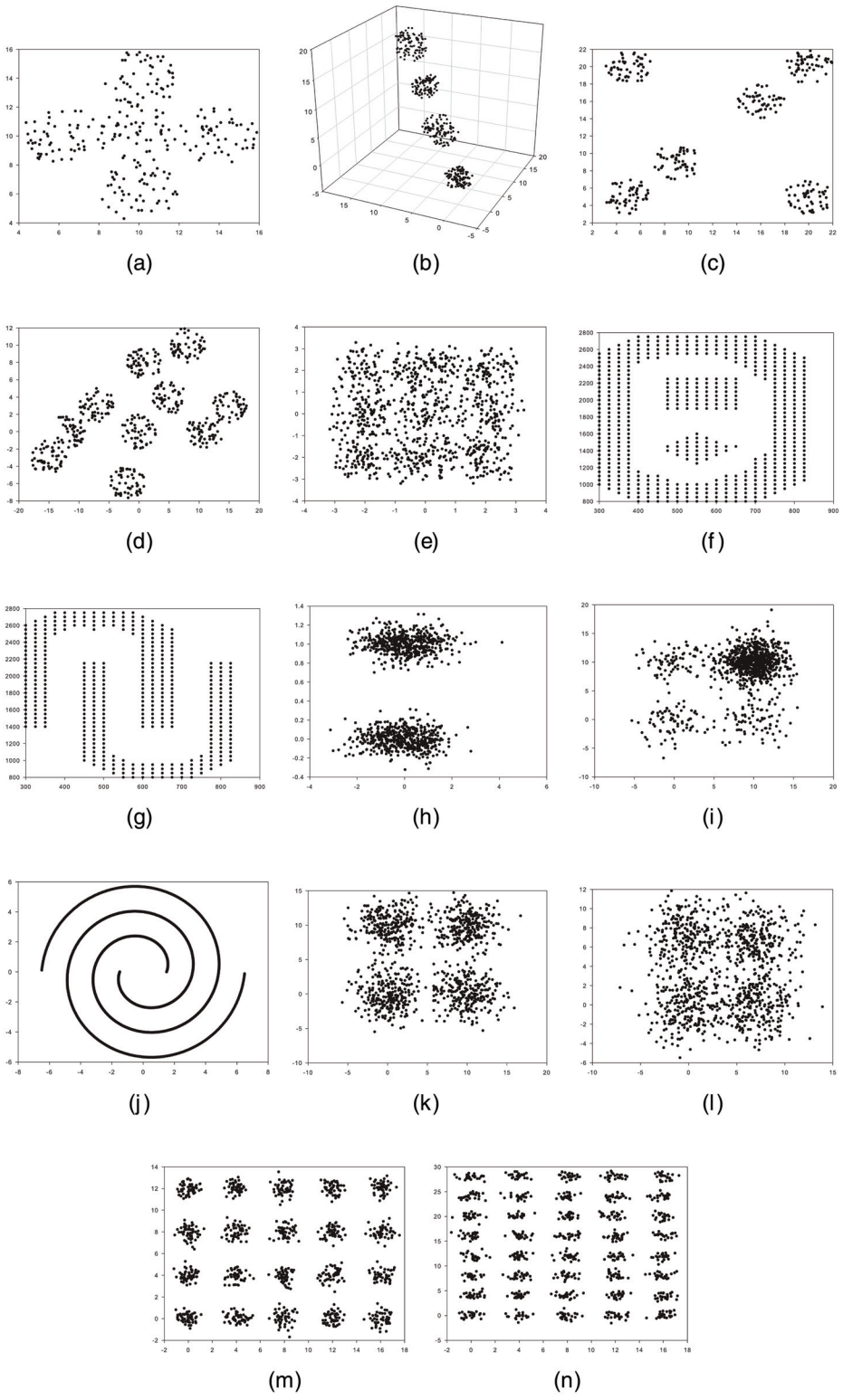

Fig 4. Graphs of the artificial datasets. (a) Sph_5_2. (b) Sph_4_3. (c) Sph_6_2. (d) Sph_10_2. (e) Sph_9_2. (f) Pat1. (g) Pat2. (h) Long1. (i) Sizes5. (j) Spiral. (k) Square1. (l) Square4. (m) Twenty. (n) Fourty.

doi:10.1371/journal.pone.0130995.g004

2. Sph_4_3 [2] dataset (Appendix B in S1 File): This dataset demonstrated in Fig 4b comprises 400-point 3D distributed over four disjointed hyper spherically shaped clusters. Each cluster contains 100 points.
3. Sph_6_2 [2] dataset (Appendix C in S1 File): This dataset involves 300-point 2D distributed over six different clusters. Each cluster embodies 50 points. This dataset is depicted in Fig 4c.
4. Sph_10_2 [30] dataset (Appendix D in S1 File): This dataset accommodates 500-point 2D distributed over 10 different clusters, of which some are overlapping. Each cluster holds 50 points. This dataset is shown in Fig 4d.
5. Sph_9_2 [30] dataset (Appendix E in S1 File): Specified in Fig 4e, this dataset embodies 900-point 2D distributed over nine highly overlapping clusters, in which each cluster incorporates 100 points.
6. Pat1 [31] dataset (Appendix F in S1 File): This dataset involves 557-point 2D distributed over three different clusters; one of these clusters is non-convex. This dataset is signified in Fig 4f.
7. Pat2 [31] dataset (Appendix G in S1 File): This dataset contains 417-point 2D distributed over two nonlinear, non-symmetric, and non-overlapping clusters. Fig 4g shows this dataset.
8. Long1 [3] dataset (Appendix H in S1 File): This dataset shown in Fig 4h encloses 1000-point 2D distributed over two long-shaped clusters.
9. Sizes5 [3] dataset (Appendix I in S1 File): This dataset comprises 1000-point 2D distributed over four square-shaped clusters, one of which contains more points than the others. Fig 4i displays this dataset.
10. Spiral [3] dataset (Appendix J in S1 File): This dataset exhibited in Fig 4j consists of 1000-point 2D distributed over two spiral-shaped clusters.
11. Square1 [3] dataset (Appendix K in S1 File): This dataset includes 1000-point 2D distributed over four semi-overlapping square-shaped clusters. Each cluster contains 250 points. This dataset is shown in Fig 4k.
12. Square4 [3] dataset (Appendix L in S1 File): Specified in Fig 4l, this dataset comprises 1000-point 2D distributed over four overlapping square-shaped clusters, each containing 250 points.
13. Twenty [3] dataset (Appendix M in S1 File): This dataset incorporates 1000-point 2D distributed over 20 small clusters. Each cluster contains 50 points. This dataset is shown in Fig 4m.
14. Fourty [3] dataset (Appendix N in S1 File): This dataset exhibited in Fig 4n consists of 1000-point 2D distributed over 40 small clusters. Each cluster contains 25 points.

- **Real-life datasets**

1. Iris [32] dataset (Appendix O in S1 File): This dataset comprises 150 four-feature samples distributed over three clusters each containing 50 observations. These samples are obtained from different categories of the iris flower (i.e., Setosa, Versicolor, and Virginica). Each sample has four feature values: sepal length, sepal width, petal length, and petal width. Two clusters of the iris flower (Versicolor and Virginica) are highly overlapping.

2. Cancer [32] dataset (Appendix P in S1 File): This dataset consists of 683 samples with nine laboratory tests distributed over two clusters. Procured from Wisconsin Breast Cancer, these samples consist of two categories, malignant and benign, which are known to be linearly separable.
3. Newthyroid [32] dataset (Appendix Q in S1 File): This dataset incorporates 215 instances with five laboratory tests distributed over three clusters. These samples are labeled as "Thyroid gland data," which embody three categories (i.e., normal, hypo, and hyper).
4. LiverDisorder [32] dataset (Appendix R in S1 File): This dataset represents 345 instances with six laboratory tests distributed over two clusters. The task is to determine whether a person suffers from alcoholism.
5. Glass [32] dataset (Appendix S in S1 File): This dataset involves 214 samples with nine features distributed over six clusters. The field of criminological investigations has motivated the study on classifying the types of glass. At the scene of the crime, a glass left can provide evidence if it is correctly identified. In this dataset, the 10th feature (ID number) has been removed.

Evaluating the clustering quality

An external criterion of the clustering quality for evaluating the results is presented in this section. The F-measure [33] is selected to compute the final solution obtained from the MOPSOSA, GenClustMOO, GenClustPESA2, MOCK, VGAPS, KM, and SL clustering algorithms. Let T and C be the two clustering solutions, $T = \{T_1, \ldots, T_{k_T}\}$ be the truth solution, and $C = \{C_1, \ldots, C_{k_C}\}$ be the solution to be measured, where k_T and K_C are number of clusters for the solutions T and C respectively. The F-measure of classes T_i and cluster C_j are defined as follows:

$$F\left(T_i, C_j\right) = \frac{2*P(T_i, C_j)*R(T_i, C_j)}{P(T_i, C_j) + R(T_i, C_j)} \tag{19}$$

where $P(T_i,C_j) = n_{ij}/|C_j|$ and $R(T_i,C_j) = n_{ij}/|T_i|$ Meanwhile, the F-measure of solutions T and C are construed below:

$$F(T, C) = \sum_{i=1}^{k_T} \frac{|T_i|}{n} \max_{c_j \in C} \left\{F\left(T_i, C_j\right)\right\} \tag{20}$$

where n is the number of the dataset. Higher values of $F(T,C)$ are better values, and the optimal value of $F(T,C)$ is 1.

Parameter settings

Table 2 presents the parameter settings employed in the proposed MOPSOSA algorithm. The performance of this algorithm is compared with three multi-objective automatic and three single-objective clustering algorithms (i.e., GenClustMOO, GenClustPESA2, MOCK, VGAPS, KM, and SL). These algorithms and the proposed algorithm are executed on all the above mentioned datasets. Employing semi-supervised method [20], the GenClustMOO and GenClustPESA2 algorithms select the best solutions from the final Pareto set. Additional details on the standard parameters employed in these algorithms can be acquired in Saha and Bandyopadhyay [8]. In the MOCK algorithm, GAP statistics [34] is used to select the best solution. The

Table 2. Parameter settings used in MOPSOSA algorithm.

Description	Parameters	Value
Swarm size	k	50
Number of iteration	*Iter*	100
Probability value to generate W	w	0.95
Probability value to generate R_1	r_1	0.90
Probability value to generate R_2	r_2	0.90
Minimum number of clusters	K_{min}	2
Maximum number of clusters	K_{max}	$\sqrt{n}+1$
Initial temperature	T_0	100

doi:10.1371/journal.pone.0130995.t002

source code of the standard parameters used in MOCK is available in [3]. VGAPS, KM, and SL clustering algorithms provide a single solution. In VGAPS, population size is equal to 100, the number of generation is equivalent to 60, and mutation and crossover probabilities are computed adaptively. The total computations implemented in the proposed algorithm, GenClustMOO, GenClustPESA2, MOCK, and VGAPS, as well as the number of iterations in KM and SL, are all equal. Each algorithm is implemented 30 times.

Results and Discussions

For each algorithm, the average value of F-measure is calculated for the final best solution to compare and exhibit the performance of the proposed algorithm with that of other algorithms. More information about the results of the cluster number and F-measure values of GenClustMOO, GenClustPESA2, MOCK, VGAPS, KM, and SL on the specified datasets can be acquired from Saha and Bandyopadhyay [8]. Table 3 displays the best value of F-measure and the number of clusters for the datasets automatically obtained with MOPSOSA, GenClustMOO, GenClustPESA2, MOCK, and VGAPS automatic clustering techniques. KM and SL are implemented with the actual number of clusters on all datasets.

Discussion of the artificial datasets results

1. Sph_5_2: Table 4 displays that the maximum F-measure value for this dataset was obtained with the MOPSOSA algorithm even though existence five overlapping spherical clusters. However, MOPSOSA, GenClustMOO, GenClustPESA2, and VGAPS established the actual number of clusters as illustrated in Table 3. Fig 5a shows the clustering of this dataset after the MOPSOSA algorithm was applied.
2. Sph_4_3: The actual number for this dataset was detected with the MOPSOSA, GenClustMOO, GenClustPESA2, MOCK, and VGAPS clustering algorithms. All seven algorithms also achieved an F-measure value of 1, providing 100% accuracy for the clustering of this dataset (refer to Tables 3 and 4). Fig 5b exhibits the graph of clusters Sph_4_3 after the MOPSOSA algorithm was employed.
3. Sph_6_2: The F-measure value for this dataset was determined to be 1 for the seven algorithms (Table 4), signifying the accurate performance of all algorithms. Moreover, all algorithms attained the real number of clusters as demonstrated in Table 3. Fig 5c depicts the graph of the clusters for this dataset with the application of the MOPSOSA algorithm.

Table 3. F-measure value and the number of clusters for different datasets obtained by MOPSOSA compared with those acquired by GenClustMOO, GenClustPESA2, MOCK, and VGAPS algorithms.

		MOPSOSA		GenClustMOO [8]		GenClustPESA2 [8]		MOCK [3]		VGAPS [4]	
Dataset	**# Clusters**	**k**	**FM**	**k**	**FM**	**k**	**FM**	**k**	**FM**	**k**	**FM**
Sph_5_2	5	5	0.98	5	0.97	5	0.94	6	0.91	5	0.55
Sph_4_3	4	4	1.00	4	1.00	4	1.00	4	1.00	4	1.00
Sph_6_2	6	6	1.00	6	1.00	6	1.00	6	1.00	6	1.00
Sph_10_2	10	10	0.99	10	0.99	12	0.94	6	0.72	7	0.76
Sph_9_2	9	9	0.92	9	0.69	8	0.66	9	0.73	9	0.49
Pat1	3	3	1.00	3	0.95	3	0.95	10	0.55	4	0.42
Pat2	2	2	1.00	2	1.00	2	1.00	11	0.55	4	0.59
Long1	2	2	1.00	2	1.00	2	1.00	2	1.00	3	0.50
Sizes5	4	4	0.98	4	0.97	3	0.88	2	0.80	5	0.82
Spiral	2	2	1.00	2	1.00	2	1.00	3	0.95	6	0.38
Square1	4	4	0.99	4	0.99	4	0.99	4	0.99	4	0.99
Square4	4	4	0.94	4	0.92	4	0.88	4	0.90	2	0.93
Twenty	20	20	1.00	20	1.00	24	0.95	20	1.00	20	0.48
Fourty	40	40	1.00	40	1.00	40	0.98	40	1.00	2	0.10
Iris	3	3	0.92	3	0.79	3	0.93	2	0.78	3	0.76
Cancer	2	2	0.98	2	0.97	2	0.98	2	0.82	2	0.95
Newthyroid	3	3	0.89	3	0.86	9	0.69	2	0.74	5	0.66
Liver Disorder	2	2	0.69	2	0.67	5	0.60	2	0.67	2	0.70
Glass	6	6	0.57	6	0.49	5	0.53	5	0.53	5	0.53

doi:10.1371/journal.pone.0130995.t003

Table 4. Averages and standard deviations for the F-measure values on the different datasets obtained from MOPSOSA, GenClustMOO, GenClustPESA2, MOCK, VGAPS, KM, and SL algorithms.

	F-measure that obtained from						
Dataset	**MOPSOSA**	**GenClustMOO [8]**	**GenClustPESA2 [8]**	**MOCK [3]**	**VGAPS [4]**	**KM [14]**	**SL [15]**
Sph_5_2	**0.982 ± 0.006**	0.957 ± 0.021	0.936 ± 0.012	0.902 ± 0.011	0.541 ± 0.011	0.938 ± 0.015	0.661 ± 0.012
Sph_4_3	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**
Sph_6_2	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**
Sph_10_2	**0.991 ± 0.002**	0.981 ± 0.011	0.931 ± 0.021	0.717 ± 0.013	0.752 ± 0.011	0.891 ± 0.014	0.841 ± 0.011
Sph_9_2	**0.921 ± 0.001**	0.681 ± 0.012	0.652 ± 0.018	0.717 ± 0.009	0.481 ± 0.012	0.683 ± 0.013	0.250 ± 0.014
Pat1	**0.989 ± 0.012**	0.946 ± 0.013	0.946 ± 0.009	0.547 ± 0.011	0.418 ± 0.014	0.618 ± 0.008	0.882 ± 0.011
Pat2	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**	0.545 ± 0.013	0.582 ± 0.021	0.754 ± 0.013	**1.000 ± 0.000**
Long1	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**	0.487 ± 0.021	0.500 ± 0.011	**1.000 ± 0.000**
Sizes5	**0.977 ± 0.001**	0.968 ± 0.001	0.883 ± 0.011	0.791 ± 0.012	0.816 ± 0.013	0.226 ± 0.021	0.181 ± 0.011
Spiral	**1.000 ± 0.000**	**1.000 ± 0.000**	**1.000 ± 0.000**	0.948 ± 0.011	0.373 ± 0.016	0.509 ± 0.011	0.504 ± 0.015
Square1	**0.999 ± 0.011**	**0.999 ± 0.013**	**0.999 ± 0.014**	**0.999 ± 0.012**	**0.999 ± 0.014**	0.732 ± 0.021	0.368 ± 0.006
Square4	**0.935 ± 0.001**	0.918 ± 0.014	0.878 ± 0.011	0.895 ± 0.011	0.925 ± 0.013	0.715 ± 0.015	0.368 ± 0.016
Twenty	**1.000 ± 0.000**	**1.000 ± 0.000**	0.948 ± 0.015	**1.000 ± 0.000**	0.479 ± 0.022	0.809 ± 0.003	0.947 ± 0.009
Fourty	**1.000 ± 0.000**	**1.000 ± 0.000**	0.979 ± 0.015	**1.000 ± 0.000**	0.950 ± 0.006	0.798 ± 0.018	0.909 ± 0.023
Iris	**0.937 ± 0.011**	0.788 ± 0.011	0.926 ± 0.015	0.775 ± 0.022	0.754 ± 0.013	0.887 ± 0.001	0.764 ± 0.009
Cancer	**0.981 ± 0.003**	0.969 ± 0.009	0.979 ± 0.014	0.918 ± 0.014	0.953 ± 0.012	0.961 ± 0.013	0.688 ± 0.008
Newthyroid	**0.885 ± 0.010**	0.863 ± 0.016	0.687 ± 0.015	0.739 ± 0.014	0.659 ± 0.011	0.677 ± 0.013	0.648 ± 0.009
Liver Disorder	**0.770 ± 0.010**	0.673 ± 0.002	0.603 ± 0.015	0.671 ± 0.012	0.705 ± 0.009	0.655 ± 0.013	0.672 ± 0.006
Glass	**0.568 ± 0.002**	0.494 ± 0.012	0.534 ± 0.012	0.534 ± 0.006	0.534 ± 0.008	0.492 ± 0.014	0.422 ± 0.007

The best F-measure for each dataset is marked in bold. Each algorithm is implemented on 30 independent runs.

doi:10.1371/journal.pone.0130995.t004

4. Sph_10_2: Table 3 reveals that only the MOPSOSA and GenClustMOO clustering algorithms achieved the desired number of clusters for this dataset. However, a maximum F-measure value was obtained with MOPSOSA (refer to Table 4) despite some overlap in these datasets. Fig 5d shows the graph for the clustering of Sph_10_2 with the post-application of the MOPSOSA algorithm.

5. Sph_9_2: For this dataset, Table 3 shows that MOPSOSA, GenClustMOO, MOCK, and VGAPS, except GenClustPESA2, were identified to be highly efficient in detecting the actual number of clusters. Despite the existence overlaps in all clusters for this dataset, MOPSOSA obtained a maximum F-measure value, demonstrating the accuracy of its performance (refer to Table 4). Fig 5e illustrates the dataset clustering with the MOPSOSA algorithm.

6. Pat1: Table 4 demonstrates that only MOPSOSA achieved the maximum F-measure value for this dataset, indicating a high accuracy clustering for well-separated clusters and for clusters of various shapes. Nevertheless, MOPSOSA, GenClustMOO, and GenClustPESA2 clustering algorithms attained the real number of clusters (Table 3), whereas MOCK was observed inappropriate for this dataset. The three clusters are clearly depicted in Fig 5f after the algorithm was applied on this dataset.

7. Pat2: Tables 3 and 4 show that the MOPSOSA, GenClustMOO, and GenClustPESA2 clustering algorithms obtained the real number of clusters for this dataset with the F-measure value as 1, signifying the high clustering accuracy of these algorithms in clustering these nonlinear and non-spherically dataset. Fig 5g reveals the graph of the two clusters in Pat2 with the application of the MOPSOSA algorithm.

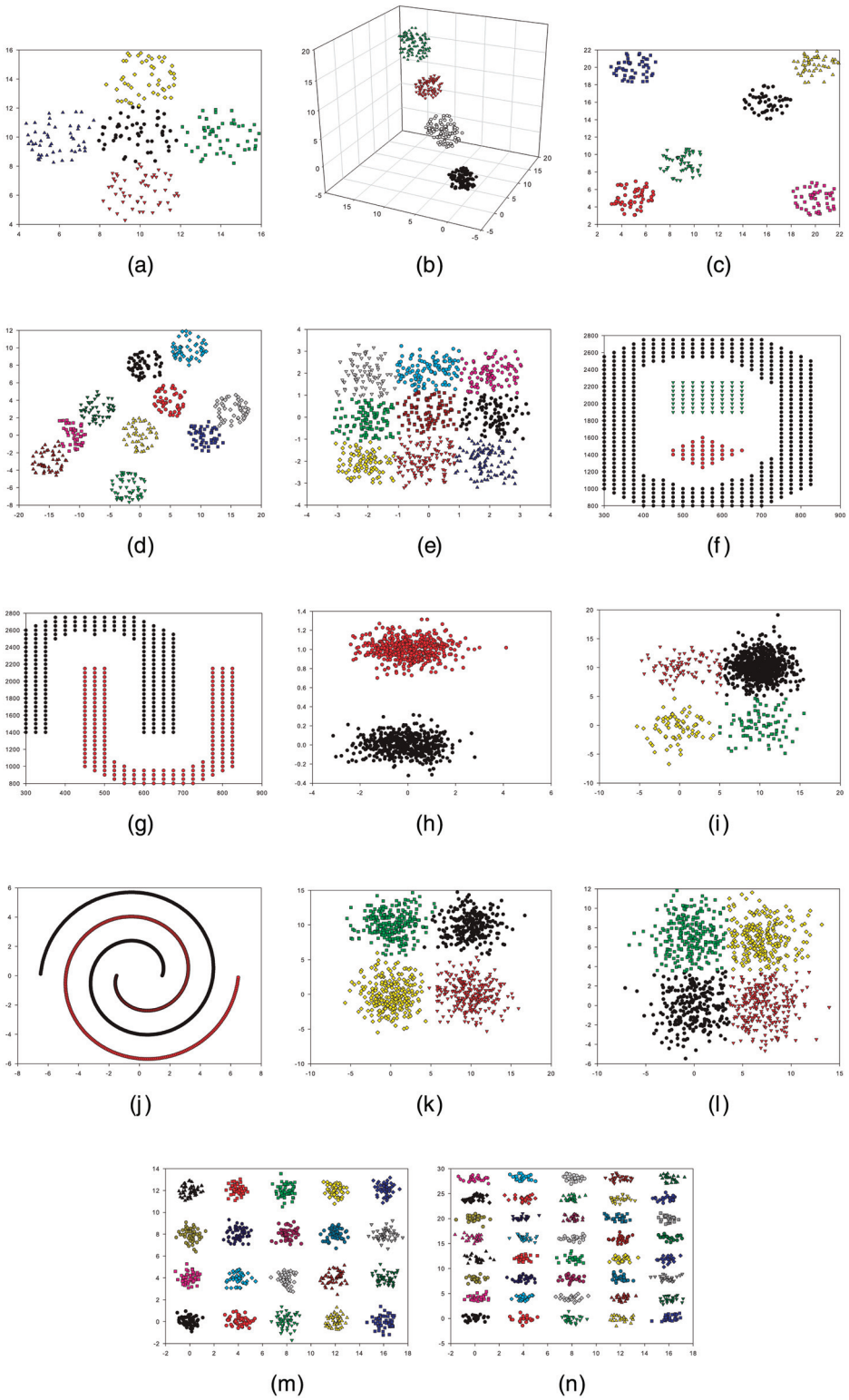

Fig 5. Graphs of the artificial datasets after applying the MOPSOSA algorithm. (a) Sph_5_2. (b) Sph_4_3. (c) Sph_6_2. (d) Sph_10_2. (e) Sph_9_2. (f) Pat1. (g) Pat2. (h) Long1. (i) Sizes5. (j) Spiral. (k) Square1. (l) Square4. (m) Twenty. (n) Fourty.

doi:10.1371/journal.pone.0130995.g005

8. Long1: For this dataset, MOPSOSA, GenClustMOO, GenClustPESA2, MOCK, and SL acquired the F-measure value of 1. Meanwhile, MOPSOSA, GenClustMOO, GenClustPESA2, and MOCK automatically resolved the proper cluster numbers for this dataset (refer to Tables 3 and 4). Fig 5h presents the clustering of this dataset into two correct clusters with the application of the MOPSOSA algorithm.

9. Sizes5: Table 4 reveals the maximum F-measure value obtained with the MOPSOSA algorithm for this dataset, which indicates that the proposed algorithm is qualified to clustering a dataset with different sizes of clusters. Regardless, Table 3 specifies that both MOPSOSA and GenClustMOO identified the actual number of clusters. Fig 5i shows the result of clustering on this dataset with the application of the MOPSOSA algorithm.

10. Spiral: Table 4 indicates that an F-measure value of 1 was acquired by MOPSOSA, GenClustMOO, and GenClustPESA2 for this dataset, indicating 100% accurate clustering on the spiral shapes. MOPSOSA, GenClustMOO, and GenClustPESA2 clustering algorithms also determined the real number of clusters as shown in Table 3. Fig 5j is a clear graphic illustration of the two spirals for this dataset with the application of the MOPSOSA algorithm.

11. Square1: For this dataset, all five automatic clustering algorithms (MOPSOSA, GenClustMOO, GenClustPESA2, MOCK, and VGAPS) detected the appropriate number of clusters (refer to Tables 3 and 4) and obtained the maximum F-measure value, thereby indicating their high accuracy in clustering this dataset. Fig 5k illustrates the result of clustering Square1 into four clusters by applying the MOPSOSA algorithm.

12. Square4: Table 3 exhibits that, for this dataset, MOPSOSA, GenClustMOO, GenClustPESA2, and MOCK, except VGAPS, established the actual number of clusters, with the maximum F-measure value obtained via MOPSOSA (see Table 4). The proposed algorithm was capable to clustering this data with high accuracy even though there are four overlapping clusters. The graph for the clustering of this dataset using the MOPSOSA algorithm is depicted in Fig 5l.

13. Twenty: For this dataset, MOPSOSA, GenClustMOO, MOCK, and VGAPS determined the real number of clusters (see Table 3), except GenClustPESA2. However, MOPSOSA, GenClustMOO, and MOCK obtained an F-measure value of 1, demonstrating an extremely high clustering accuracy even for several clusters (refer to Table 4). The clusters for this dataset after the application of MOPSOSA algorithm is graphically shown in Fig 5m.

14. Fourty: Table 3 reveals that for this dataset, only three automatic clustering algorithms (MOPSOSA, GenClustMOO, and MOCK) identified the desired cluster number. All these algorithms also obtained the F-measure value of 1, demonstrating an exceedingly high clustering accuracy despite the large number of clusters (refer to Table 4). Fig 5n depicts the graph for clustering this dataset after the application of the MOPSOSA algorithm.

Discussion of the real-life datasets results

1. Iris: Table 4 shows that for this dataset, the maximum F-measure value was obtained with the proposed algorithm MOPSOSA. However, with the exception of MOCK, all four automatic clustering algorithms (MOPSOSA, GenClustMOO, GenClustPESA2, and VGAPS) resolved the proper number of clusters, as evidenced in Table 3.

2. Cancer: The maximum F-measure value for this dataset was obtained with the proposed MOPSOSA algorithm (see Table 4). Nevertheless, all five automatic clustering algorithms

(MOPSOSA, GenClustMOO, GenClustPESA2, MOCK, and VGAPS) identified the correct number of clusters, as illustrated in Table 3.

3. Newthyroid: Table 4 reveals that the maximum F-measure value for this dataset was attained with the MOPSOSA algorithm. However, Table 3 specifies that only two automatic clustering algorithms (MOPSOSA and GenClustMOO) determined the actual number of clusters.

4. Liver Disorder: For this dataset, MOPSOSA, GenClustMOO, MOCK, and VGAPS, except GenClustPESA2, identified the actual number of clusters (refer to Table 3). Meanwhile, the maximum F-measure was achieved with the proposed algorithm MOPSOSA (refer to Table 4).

5. Glass: Table 4 demonstrates that the maximum F-measure value for this dataset was obtained with the MOPSOSA algorithm. Only MOPSOSA and GenClustMOO automatic clustering algorithms were determined to be capable of achieving the desired number of clusters (see Table 3).

Summary of results

The above results signify that the proposed MOPSOSA algorithm achieves accurate results in all datasets. Moreover, the proposed algorithm can automatically establish the correct cluster numbers for all datasets used in the experiment. The algorithm is also proven capable of dealing with various shapes of datasets (hyper spheres, linear, and spiral), overlapping datasets, datasets that have well-separated clusters with any convex and non-convex shapes, and datasets that contain several clusters. With most datasets having dimensions from 2 to 9, objects from 150 to 1000, and number of clusters from 2 to 40, the MOPSOSA algorithm displays superiority over the three multi-objective automatic and three single-objective clustering algorithms. The results also show that the GenClustMOO algorithm can automatically identify the actual cluster numbers, but with a lower quality of clustering accuracy than the proposed algorithm. In general, MOCK can detect the number of clusters for hyper spheres and linear, but it is unsuccessful for non-convex well-separated and overlapping clusters. The results also prove that the VGAPS algorithm is not suitable for non-convex well-separated clusters or for datasets with numerous clusters.

The main factors that led to the accuracy of the proposed algorithm in solving the clustering problem are attributed to the power and speed of the search characterized by the particle swarm, with the guarantee of not becoming stagnant into local solutions via the MOSA algorithm. The development of particle swarm to address more than one validity index can cluster any dataset. The generation of the initial swarm of particles can be improved with KM method. Meanwhile, the repository for preserving the diversity of clustering solutions can be updated by adopting the sharing fitness, and the redundant particles can be eliminated with the renumbering process.

Conclusion

This research proposed a new automatic multi-objective clustering algorithm MOPSOSA based on a hybrid multi-objective particle swarm algorithm and multi-objective simulated annealing. A multi-objective particle swarm optimization was also developed from a combinatorial particle swarm optimization. The proposed algorithm was proven capable of automatically clustering the dataset into the appropriate number of clusters. With the simultaneous optimization of three objective functions, the Pareto optimal set was obtained from the

proposed algorithm. The first objective function considered the compactness of the clustering based on Euclidean distance. The second function regarded the total symmetry of the clusters, and the third considered the connectedness of the clusters. The proposed algorithm was performed on 19 real-life and artificial datasets, and its performance was compared with that of three multi-objective automatic and three single-objective clustering techniques. MOPSOSA obtained better accuracy in its results compared to that of other algorithms. The results also demonstrated that the proposed algorithm can be used for datasets of various shapes and for overlapping and non-convex datasets.

Supporting Information

S1 File. Experimental datasets. 250 points of the artificial datasets Sph_5_2 (Appendix A). 400 points of the artificial datasets Sph_4_3 (Appendix B). 300 points of the artificial datasets Sph_6_2 (Appendix C). 500 points of the artificial datasets Sph_10_2 (Appendix D). 900 points of the artificial datasets Sph_9_2 (Appendix E). 557 points of the artificial datasets Pat1 (Appendix F). 417 points of the artificial datasets Pat2 (Appendix G). 1000 points of the artificial datasets Long1 (Appendix H). 1000 points of the artificial datasets Sizes5 (Appendix I). 1000 points of the artificial datasets Spiral (Appendix J). 1000 points of the artificial datasets Square1 (Appendix K). 1000 points of the artificial datasets Square4 (Appendix L). 1000 points of the artificial datasets Twenty (Appendix M). 1000 points of the artificial datasets Fourty (Appendix N). 150 samples of the real-life datasets Iris (Appendix O). 683 samples of the real-life datasets Cancer (Appendix P). 215 instances of the real-life datasets Newthyroid (Appendix Q). 345 instances of the real-life datasets LiverDisorder (Appendix R). 214 samples of the real-life datasets Glass (Appendix S).
(PDF)

Author Contributions

Conceived and designed the experiments: AA. Performed the experiments: AA. Analyzed the data: AA. Contributed reagents/materials/analysis tools: AA. Wrote the paper: AA AB MA.

References

1. Jain AK, Dubes RC. Algorithms for clustering data. Upper Saddle River: Prentice hall Englewood Cliffs; 1988.
2. Bandyopadhyay S, Maulik U. Genetic clustering for automatic evolution of clusters and application to image classification. Pattern recognition. 2002; 35(6):1197–208.
3. Handl J, Knowles J. An evolutionary approach to multiobjective clustering. IEEE Transactions on Evolutionary Computation. 2007; 11(1):56–76.
4. Bandyopadhyay S, Saha S. A point symmetry-based clustering technique for automatic evolution of clusters. IEEE Transactions on Knowledge and Data Engineering. 2008; 20(11):1441–57.
5. Saha S, Bandyopadhyay S. A symmetry based multiobjective clustering technique for automatic evolution of clusters. Pattern recognition. 2010; 43(3):738–51.
6. Liu Y, Wu X, Shen Y. Automatic clustering using genetic algorithms. Applied mathematics and computation. 2011; 218(4):1267–79.
7. Masoud H, Jalili S, Hasheminejad SMH. Dynamic clustering using combinatorial particle swarm optimization. Applied intelligence. 2013; 38(3):289–314.
8. Saha S, Bandyopadhyay S. A generalized automatic clustering algorithm in a multiobjective framework. Applied Soft Computing. 2013; 13(1):89–108.
9. Halkidi M, Vazirgiannis M, editors. Clustering validity assessment: Finding the optimal partitioning of a data set. Proceedings of the 2001 IEEE International Conference on Data Mining (ICDM 01); 2001; California, USA: Proceedings of the 2001 IEEE International Conference on Data Mining (ICDM 01).

10. Suresh K, Kundu D, Ghosh S, Das S, Abraham A, Han SY. Multi-objective differential evolution for automatic clustering with application to micro-array data analysis. Sensors. 2009; 9(5):3981–4004. doi: 10.3390/s90503981 PMID: 22412346
11. Liu Y, Özyer T, Alhajj R, Barker K. Integrating Multi-Objective Genetic Algorithm and Validity Analysis for Locating and Ranking Alternative Clustering. Informatica (Slovenia). 2005; 29(1):33–40.
12. Matake N, Hiroyasu T, Miki M, Senda T, editors. Multiobjective clustering with automatic k-determination for large-scale data. Proceedings of the 9th annual conference on Genetic and evolutionary computation; 2007; London, England: In Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2007).
13. Bandyopadhyay S, Saha S. GAPS: A clustering method using a new point symmetry-based distance measure. Pattern recognition. 2007; 40(12):3430–51.
14. MacQueen J, editor Some methods for classification and analysis of multivariate observations. Proceedings of the fifth Berkeley symposium on mathematical statistics and probability; 1967; Oakland, CA, USA.: In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability.
15. Everitt BS, Landau S, Leese M. Cluster Analysis: Hodder Arnold, London 2001.
16. Davies DL, Bouldin DW. A cluster separation measure. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1979; 1(2):224–7. PMID: 21868852
17. Saha S, Bandyopadhyay S. Some connectivity based cluster validity indices. Applied Soft Computing. 2012; 12(5):1555–65.
18. Ulungu B, Teghem J, Fortemps P. Heuristic for multi-objective combinatorial optimization problems by simulated annealing. In: Gu J, Chen G, Wei Q, Wang S, editors. MCDM: Theory and Applications: Sci-Tech; 1995. p. 229–38.
19. Goldberg DE, Richardson J, editors. Genetic algorithms with sharing for multimodal function optimization. Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms; 1987; Cambridge, MA, USA Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms.
20. Saha S, Bandyopadhyay S. A new multiobjective simulated annealing based clustering technique using symmetry. Pattern Recognition Letters. 2009; 30(15):1392–403.
21. Shieh H-L, Kuo C-C, Chiang C-M. Modified particle swarm optimization algorithm with simulated annealing behavior and its numerical verification. Applied Mathematics and Computation. 2011; 218 (8):4365–83.
22. Mitra D, Romeo F, Sangiovanni-Vincentelli A. Convergence and Finite-Time Behavior of Simulated Annealing. Advances in Applied Probability. 1986; 18(3):747–71.
23. Hruschka ER, Campello RJGB, Freitas AA, De Carvalho APLF. A survey of evolutionary algorithms for clustering. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews. 2009; 39(2):133–55.
24. Yang SL, Li YS, Hu XX, PAN RY. Optimization Study on k Value of K-means Algorithm. Systems Engineering-Theory & Practice. 2006; 2:97–101. doi: 10.1002/jps.24311 PMID: 25546650
25. Toussaint GT. The relative neighbourhood graph of a finite planar set. Pattern recognition. 1980; 12 (4):261–8.
26. Salazar-Lechuga M, Rowe JE, editors. Particle swarm optimization and fitness sharing to solve multi-objective optimization problems. Congress on Evolutionary Computation (CEC'2005); 2005; Edinburgh, Scotland, UK: Congress on Evolutionary Computation (CEC'2005). doi: 10.1016/j.biosystems.2015.05.002 PMID: 25982071
27. Jaccard P. Lois de distribution florale dans la zone alpine. Bull Soc Vaudoise Sci Nat. 1902; 38:69–130.
28. Deb K. Multi-objective optimization using evolutionary algorithms. New York: Wiley; 2001.
29. Jardine N, Sibson R. Mathematical taxonomy. New York: Wiley; 1971.
30. Bandyopadhyay S, Pal SK. Classification and learning using genetic algorithms: applications in bioinformatics and web intelligence: Springer Science & Business Media; 2007.
31. Pal SK, Mitra S. Fuzzy versions of Kohonen's net and MLP-based classification: performance evaluation for certain nonconvex decision regions. Information Sciences. 1994; 76(3):297–337.
32. Lichman M. UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California School of Information and Computer Science. 2013.
33. Fung BC, Wang K, Ester M, editors. Hierarchical document clustering using frequent itemsets. SDM; 2003; San Francisco, CA: In Proceedings of the 3rd SIAM International Conference on Data Mining.
34. Tibshirani R, Walther G, Hastie T. Estimating the number of clusters in a data set via the gap statistic. Journal of the Royal Statistical Society: Series B 2001; 63(2):411–23.