symmetry

MDPI

Article

A New Meta-Heuristics Data Clustering Algorithm Based on Tabu Search and Adaptive Search Memory

Youseef Alotaibi

Department of Computer Science, College of Computer and Information Systems, Umm Al-Qura University, Makkah 21955, Saudi Arabia; yaotaibi@uqu.edu.sa

Abstract: Clustering is a popular data analysis and data mining problem. Symmetry can be considered as a pre-attentive feature, which can improve shapes and objects, as well as reconstruction and recognition. The symmetry-based clustering methods search for clusters that are symmetric with respect to their centers. Furthermore, the K-means (K-M) algorithm can be considered as one of the most common clustering methods. It can be operated more quickly in most conditions, as it is easily implemented. However, it is sensitively initialized and it can be easily trapped in local targets. The Tabu Search (TS) algorithm is a stochastic global optimization technique, while Adaptive Search Memory (ASM) is an important component of TS. ASM is a combination of different memory structures that save statistics about search space and gives TS needed heuristic data to explore search space economically. Thus, a new meta-heuristics algorithm called (MHTSASM) is proposed in this paper for data clustering, which is based on TS and K-M. It uses TS to make economic exploration for data with the help of ASM. It starts with a random initial solution. It obtains neighbors of the current solution called trial solutions and updates memory elements for each iteration. The intensification and diversification strategies are used to enhance the search process. The proposed MHTSASM algorithm performance is compared with multiple clustering techniques based on both optimization and meta-heuristics. The experimental results indicate the superiority of the MHTSASM algorithm compared with other multiple clustering algorithms.

Keywords: data clustering; Tabu Search; K-means; adaptive memory; meta-heuristics

check for updates

Citation: Alotaibi, Y. A New Meta-Heuristics Data Clustering Algorithm Based on Tabu Search and Adaptive Search Memory. *Symmetry* **2022**, *14*, 623. https://doi.org/10.3390/sym14030623

Academic Editor: Kuo-Hui Yeh

Received: 3 March 2022
Accepted: 18 March 2022
Published: 20 March 2022

1. Introduction

Clustering can be considered as a key problem in data analysis and data mining [1]. It tries to group similar data objects into the sets of disjoint classes or clusters [2,3]. Symmetry can be considered as a pre-attentive feature that can improve the shapes and objects as well as reconstruction and recognition. Furthermore, the symmetry-based clustering methods search for clusters that are symmetric with respect to their centers. As each cluster has the point symmetry property, an efficient point symmetry distance (PSD) was proposed by Su and Chou [4] to support partitioning the dataset into the clusters. They proposed a symmetry-based K-means (SBKM) algorithm that can give symmetry with minimum movement [5].

The literature shows that there are many techniques to solve clustering problems. They are basically categorized as the hierarchical clustering algorithm and the partitional clustering algorithm [6,7].

The hierarchical clustering algorithms mainly generate clusters hierarchies signified in a tree structure called a dendrogram. It can be successively proceeded throughout either splitting larger clusters or merging smaller clusters into larger ones. The clusters identify as the connected components in a tree. Hence, the output of hierarchical clustering algorithms is a tree of clusters [8–10].

However, partitional clustering algorithms attempt to treat a dataset as one cluster and try to decompose it into disjoint clusters. As partitional clustering does not need to make a tree structure, it can be considered to be easier than hierarchy clustering. The partitional algorithms are restricted with a criterion function while decomposing the dataset, and thus this function leads to minimizing dissimilarity in each cluster and maximizing similarity in data objects of each cluster [11].

The K-means (K-M) algorithm is considered to be one of the most common algorithms of partitional clustering [12–14]. However, it has some disadvantages, such as specifying the number of k clusters, as it is based on practical experience and knowledge. The main K-M problems are its sensitivity to initialization and becoming trapped in local optima [15,16]. Therefore, meta-heuristics algorithms are used to escape from these problems.

Meta-heuristics is an iterative generation processes that can guide a heuristic algorithm by different responsive concepts for exploring and exploiting search space [17]. It is a class of approximate methods that can attack hard combinatorial problems when the classical optimization methods fail. It allows for the creation of new hybrid methods by combining different concepts [18,19].

Tabu Search (TS) is considered to be one of the meta-heuristics algorithms based on memory objects to make economic exploitation and exploration for search space. Tabu list (TL) is one of the main memory elements of TS. It is a list that contains recently visited solutions by TS. It is used by TS in order to avoid re-entering pre-visited regions [20,21].

This paper aims to propose a new meta-heuristics Tabu Search with an Adaptive Search Memory algorithm (MHTSASM) for data clustering, which is based on TS and K-M. MHTSASM is an advanced TS that uses responsive and intelligent memory. The proposed MHTSASM algorithm uses multiple memory elements. For example, it uses the elite list (EL) memory element of TS to store the best solutions and Adaptive Search Memory (ASM) to store information about features of each solution. The ASM partitions each feature range into p partitions and stores information about these partitions in each center. There are two types of ASM used in the proposed algorithm: (1) ASMv stores number of visits; and (2) ASMu stores the number of improvement updates. In addition, there are two TS strategies used in the proposed algorithm: (1) the intensification strategy, which allows MHTSASM to focus on the best pre-visited regions; and (2) the diversification strategy, which allows MHTSASM to explore un-visited regions to ensure good exploration for search space.

The results of MHTSASM are specifically compared with other optimization and metaheuristics algorithms. The performance of MHTSASM is compared with several variable neighborhood searches, such as Variable Neighborhood Search (VNS), VNS+, and VNS-1 algorithms [22], Simulated Annealing (SA) [23], and global K-means (G1) [24]. Additionally, it is compared with hybrid algorithms, such as a new hybrid algorithm based on particle swarm optimization with K-harmonic means (KHM) [25], and an ant colony optimization clustering algorithm (ACO) with KHM [26,27].

The remainder of this paper is organized as follows: Section 2 presents the background of the proposed MHTSASM clustering algorithm. Section 3 presents the proposed MHTSASM clustering algorithm. Section 4 illustrates experimental results. Finally, the conclusion is presented in Section 5.

2. Background

2.1. K-Means (K-M) Clustering Algorithm

The K-M is a common clustering algorithm for data mining used in many real life applications, such as healthcare, environment and air pollution, and industry data. It outputs k centers that partition input points into k clusters [12–14]. However, it has some disadvantages, such as specifying the number of k clusters, as it is based on the practical experience and knowledge. An example of how to select the k number is shown in Figure 1. It has two clusterings, where the dark crosses specify the k centers. There are five k values used on the left. In addition, there are many centers that could be used on the right, even though one center can sufficiently represent the data [15]. The main K-M problems are its sensitivity to initialization and getting trapped in local optima [16]. Therefore, meta-heuristics algorithms are used to escape from these problems.

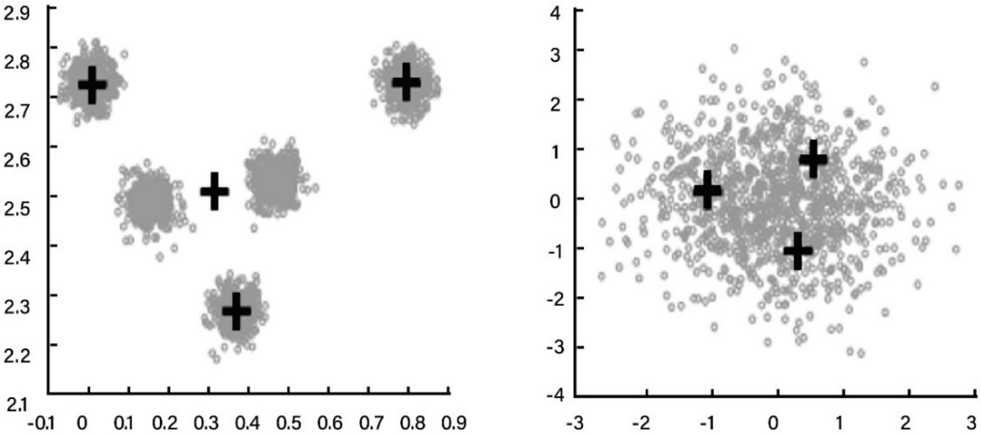

Figure 1. An example of the K-means clustering algorithm.

The K-M calculates the center of each cluster as the mean value of points that belong to the cluster. It works as follows: let D denotes the entire dataset, n the number of data objects, and d the number of features Rd, i.e., D = $\{x_1, x_2, \ldots, x_n\}$, and we want to partition set X into K partitions. K-M tries to obtain the best set of centers C = $\left\{C^1, \ldots, C^k\right\}$. Let C^j denote class j with cardinality $|S_j|$; then we can calculate C^j, j = 1, ... , K according to the following equation:

$$C^j = \frac{1}{|S_j|} \sum_{x \in S_j} x \tag{1}$$

In order to measure the goodness of different sets of centers, objective function f(C) can be calculated for each set of centers using the minimum sum of squares [10,11], defined by the following:

$$f(C) = \frac{1}{n} \sum_{i=1}^{n} \min_{j} d\left(C_j, x_i\right) \tag{2}$$

where $d\left(C_j, x_i\right)$ corresponds to the squared Euclidean distance between the center of class C and the point x_i.

K-M is known for its low computational requirements and ease of implementation. The main K-M problems are its sensitivity to initialization and becoming trapped in local optima [16,28]. Therefore, the meta-heuristics algorithms are used to escape from these problems.

2.2. Meta-Heuristics Clustering Algorithm

There are many meta-heuristics methods that can be used with K-M in order to increase its efficiency and treat its problems. The literature shows that there are several methods proposed in order to compete local optima problem of K-M, such as SA, TS, and the genetic algorithm (GA) [23].

A hybrid algorithm that uses GA and K-M (GKA) was proposed in [29,30]. GKA uses GA as a search operator, as it finds a globally optimal partition [31,32].

In addition, a hybrid algorithm was proposed in [33] to solve the problems of K-M based on GA and the weighted K-M algorithm. Moreover, another hybrid algorithm using K-M and SA was proposed in [34].

A comparison of implementation complexity and computational cost between the most common clustering algorithms was summarized, as shown Table 1 [16,23,35–37].

Table 1. Comparison between different optimization and metaheuristics algorithms [2,22,38–40].

Algorithm Name	**Implementation Complexity Level**			**Computational Cost Level**		
	High	**Medium**	**Low**	**High**	**Medium**	**Low**
K-M			✓			✓
TS			✓			✓
GA		✓		✓		
SA		✓			✓	
Algorithm 1		✓			✓	
J-M+			✓			✓
VNS		✓			✓	
VNS-1		✓			✓	
G1		✓				✓
G2		✓				✓
KHM		✓			✓	
PSO		✓			✓	
PSOKHM	✓				✓	
ACO	✓				✓	
ACOKHM	✓				✓	
Proposed MHTSASM Approach			✓			✓

2.2.1. Tabu Search (TS) Clustering Algorithm

TS is considered to be one of the meta-heuristics algorithms that is based on memory objects to make economic exploitation and exploration for search space. Tabu list (TL) is one of the main memory elements of TS. It is a list that contains recently visited solutions by TS. It is used by TS in order to avoid re-entering pre-visited regions.

Memory Elements

TS is considered to be one of the meta-heuristic algorithms. It is used to solve hard optimization problems of which classical methods often encounter great difficulty [41]. It depends on memory structures rather than memoryless methods based on semi random processes implementing a form of sampling. The MHTSASM is an advanced TS that uses responsive and intelligent memory. It uses multiple memory elements such as EL, TL, and ASM in order to minimize search time and focus on promising regions.

EL: Elite list stores best visited solutions.

TL: Tabu list stores recently visited solutions.

ASMv: Feature partitions visitation matrix. It is a matrix that partitions each feature into p partitions, and it is stored as a matrix of size K × p × d. It stores the number of visits of each partition for all generated centers.

ASMu: Feature improvement update matrix. It has the same size of ASMv and stores the number of obtained improvements of objective functions while updating the centers.

MHTSASM uses both explicit and attributive memories. Explicit memory records complete solutions such as EL and TL. Attributive memory stores information about features of each solution such as ASMv and ASMu. MHTSASM uses memory elements to guide intensification and diversification strategies.

Center Adjustment

The center adjustment method as shown in Algorithm 1 is used to re-compute the centers according to the data objects related to each center. It re-assigns each data object x to center C^j, j = 1, . . . , K with minimum Euclidean distance between center and data object. Applying K-M, a new set of centers according to mean values of related objects to each center is computed. Equations (3) and (4) are used in Algorithm 1.

$$j^* = \arg \min_{j=1,\ldots,k} \|x^i - \widetilde{C}^{(j)}\| \tag{3}$$

$$\widetilde{C}^{(j)} = \frac{1}{|S_j|} \sum\nolimits_{x \in S_j} x \tag{4}$$

Algorithm 1: Center Adjust(K, $\widetilde{C}$, d, D)

```
1.  Set $S_j$, j = 1, . . . , K to be empty.
2.  Repeat (2.1–2.2) steps for i = 1, . . . , n.
    2.1. Compute the Euclidean distance between data object xi and each center $\widetilde{C}^{(j)}$, j = 1, . . . , K.
    2.2. Add $x^i$ to $S_{j*}$ where
Compute Equation (3)
    Hence, $S_{j*} = S_{j*} \cup x^i$
3.  Compute a new set of centers $\widetilde{C} = \left\{\widetilde{C}^{(1)}, \ldots, \widetilde{C}^{(k)}\right\}$; each center $\widetilde{C}^{(j)} \in \widetilde{C}$ is computed
    according to the following equation:
Compute Equation (4)
4.  Return $\widetilde{C}^{(j)}$, j = 1, . . . , K.
```

3. The Proposed MHTSASM Algorithm

In this paper, the MHTSASM algorithm is proposed for data clustering, which is based on TS and K-M. Figure 2 shows the proposed MHTSASM algorithm framework. The MHTSASM starts with a random initial solution. It obtains neighbors of the current solution called trial solutions and updates memory elements for each iteration. Furthermore, it uses intensification and diversification strategies guided by memory elements to enhance the search process. Thus, it is a high level TS algorithm with long term memory elements. In the following subsections, the main components of MHTSASM are explained.

3.1. Intensification and Diversification Strategies

The TS has two highly important components called (1) intensification strategy; and (2) diversification strategy. The MHTSASM uses both intensification and diversification strategies guided by ASM. Intensification tries to immigrate to best solutions. It can recruit a return into striking regions in order to explore them more systematically. Furthermore, intensification techniques require saving elite solutions to be able to examine their neighbors, and thus the MHTSASM records elite solutions in EL. The MHTSASM may use the responsive memory ASMu in order to keep track of best partitions of each feature in the search space.

The intensification method as shown in Algorithm 2 is guided by ASMu. It selects a random number of features from d features. The MHTSASM selects a partition with the highest updating improvement value for the current center for each selected feature. Each selected feature value is updated with a random value in range between the current feature value and the selected partition. The memory elements are updated, and the updated set of centers is returned. Algorithm 2 states formally the steps of the intensification process.

On the other hand, the diversification strategy can develop the exploration process in order to inspect the non-visited regions and create a new solution. It can improve the exploration for search space. The MHTSASM applies diversification by choosing partitions from ASMv with a minimum number of visits.

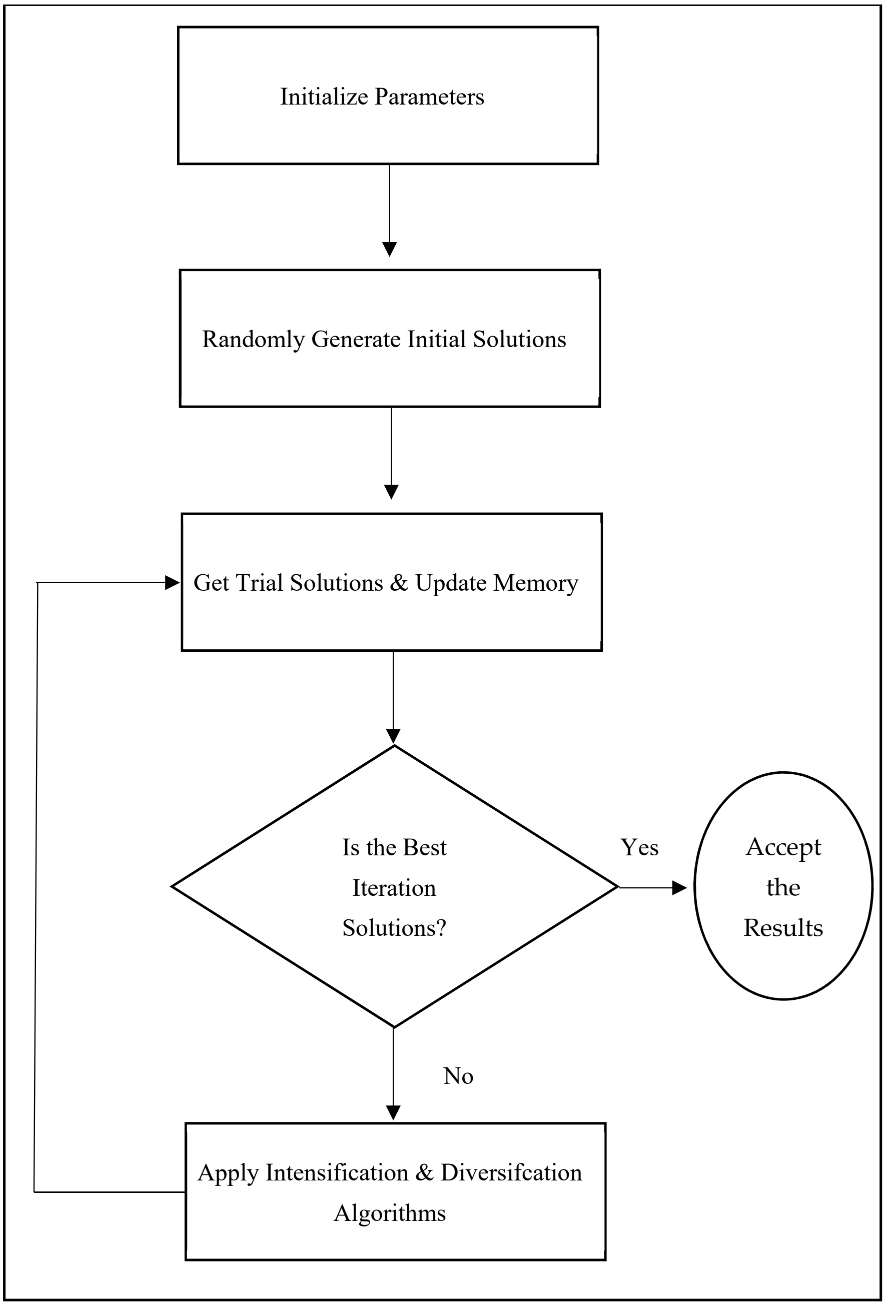
Initialize Parameters
Randomly Generate Initial Solutions
Get Trial Solutions & Update Memory
Is the Best Iteration Solutions?
Yes
Accept the Results
No
Apply Intensification & Diversifcation Algorithms

Figure 2. The proposed MHTSASM algorithm framework.

Algorithm 2: Intensification Algorithm $\left(\delta,\ \widetilde{C}, \mathrm{ASM_u},\ \mathrm{ASM_v},\ d\right)$

```
1. For each center in $\widetilde{C}$, repeat the following steps (1.1–1.4).
    1.1. Select $\delta$ features with highest updating improvement value in $\mathrm{ASM_u}$ from d data features.
    1.2. Select a partition from p partitions for each selected feature with the highest updating improvement value for current center in $\mathrm{ASM_u}$.
    1.3. Update the value of the selected feature by computing a random value in the range between the current value and the selected partition value.
    1.4. Update memory elements $\mathrm{ASM_v}$.
2. Return the updated centers $\widetilde{C}$, $\mathrm{ASM_v}$.
```

The diversification method as shown in Algorithm 3 tends to explore unvisited regions guided by ASMv. It selects a random number of features from d features. Next, the MHTSASM selects a partition with a minimum number of visits for the current center for the selected features. Each feature value is updated with a random value in the range of the selected partition depending on the ASMv information. The memory elements are updated, and the updated set of centers is returned. Algorithm 3 states formally the steps of the diversification process.

Algorithm 3: Diversification Method $\left(\delta, \widetilde{C}, \mathrm{ASM_v}, d\right)$

```
1.   For each center in $\widetilde{C}$, repeat the following steps (1.1–1.4).
     1.1.  Select $\delta$ features from features with the least number of visits in $\mathrm{ASM_v}$ from d
           data features.
     1.2.  Select a partition from p partitions for each selected feature with the least number of
           visits for the current center in $\mathrm{ASM_v}$.
     1.3.  Update the value of the selected feature by computing a random value in the range
           between the current value and the selected partition value.
     1.4.  Update memory elements $\mathrm{ASM_v}$.
2.   Return the updated centers $\widetilde{C}$, $\mathrm{ASM_v}$.
```

3.2. Trial Solutions Generation Algorithm

The trial solutions generation method as shown in Algorithm 4 generates μ trial solutions near the current set of centers C. It may select the random features for each center and then generate a new random value that belongs to the range of the current value partition. It makes a thorough search for neighbors of the current set of centers C. Algorithm 4 states formally the steps of the trial solutions generation process.

Algorithm 4: Trial Solutions Method (C, μ, $\mathrm{ASM_v}$, d)

```
1.   Repeat the following steps (1.1–1.3) for j = 1, . . . , $\mu$.
     1.1.  Set $\widetilde{C}^{(j)}$ equal to the input centers C.
     1.2.  Select $\eta$ features randomly from d data features.
     1.3.  Repeat steps (1.3.1–1.3.2) for m = 1, . . . , $\eta$.
           1.3.1.  Update the values of the selected features in $\widetilde{C}^{(j)}$ by generating new values
                   randomly.
           1.3.2.  Update the memory elements $\mathrm{ASM_v}$.
2.   Return the updated set of centers $\widetilde{C}^{(j)}$, j = 1, . . . , $\mu$, $\mathrm{ASM_v}$.
```

3.3. Refinement Method

The refinement method as shown in Algorithm 5 is used to examine the neighbors of the best solutions that have been found so far and return with the better solution if it exists in their neighborhood. It may call the trial solution (Algorithm 4) in order to find the neighbors of the current solution. If a better solution is found, it will call the trial solution (Algorithm 4) again for a new solution; otherwise, it will break. Algorithm 5 states formally the steps of the refinement method.

Algorithm 5: Refinement Method (C, μ, $\mathrm{ASM_v}$, $\mathrm{ASM_u}$, EL, d)

```
1.   Apply Algorithm 4 to generate $\mu$ centers $\widetilde{C}$ near the set of centers C.
2.   Update the memory elements.
3.   If an improvement achieved, then go to step 1; otherwise, go to step 4.
4.   Return C, $\mathrm{ASM_v}$, $\mathrm{ASM_u}$, EL.
```

3.4. The Proposed MHTSASM Algorithm

The MHTSASM cluster algorithm is proposed in this paper. It is an advanced TS that uses responsive and intelligent memory, and it uses multiple memory elements, such as the EL memory element of TS, in order to store the best solutions, and the ASM in order to store information about features of each solution. It may use two important TS strategies: intensification and diversification. It starts with initial random centers and then applies the trail solutions (Algorithm 4) in order to generate trial solutions near initial centers. The parameter variables are used to determine if intensification or diversification is needed.

If the intensification strategy is needed, the intensification (Algorithm 2) is applied to enhance the set of centers using ASM and EL. If the centers are stuck and gain no improvement, diversification (Algorithm 3) is used to make exploration for regions that have not been visited yet. However, if termination conditions are satisfied, then refinement is carried out (Algorithm 5), where the refinement method examines the neighbors of the best solution that has been found so far. If there is no improvement, then the process is terminated. Algorithm 6 states formally the steps of the advanced MHTSASM algorithm.

Algorithm 6: MHTSASM Algorithm

```
1. Initialization. Set values for the number of clusters K, the number of neighborhood trials $\mu$,
   and set the memory elements TL, $ASM_v$, $ASM_u$, and EL to be empty.
2. Generate K random initial centers C.
3. Evaluate the initial centers C and update the memory elements TL and EL.
4. Main Loop.
    4.1. Generate $\mu$ trail solutions $\widetilde{C}^{(j)}$, j = 1, . . . , $\mu$ applying Algorithm 4.
    4.2. Repeat the following steps (4.2.1–4.2.2) for j = 1, . . . , $\mu$.
        4.2.1. Adjust set of centers $\widetilde{C}^{(j)}$ applying Algorithm 1; compute the objective
               function $f(\widetilde{C}^{(j)})$ applying Equation (2).
        4.2.2. If objective function of $f(\widetilde{C}^{(j)})$ is not better than f(C), then go to Step 4.2.1;
               otherwise set C equal to $\widetilde{C}^{(j)}$ and update $ASM_u$.
    4.3. Update the memory elements TL and EL.
5. Termination. If the termination conditions are satisfied then go to Step 8. Otherwise, go to
   Step 6.
6. Intensification. If an intensive solution is needed, then apply Algorithm 3 to generate new
   centers C. Update the memory elements and go to Step 4.
7. Diversification. If a diverse solution is needed, then apply Algorithm 2 to generate new
   centers C. Update the memory elements and go to Step 4.
8. Final Refinement. Apply Algorithm 5 to refine the best solution found so far.
```

4. Numerical Experiments

4.1. Dataset

There are several numerical experiments for small and medium dataset sizes that were done in order to validate the clustering algorithm effectiveness. In other words, there were three clustering test problems used in order to evaluate the proposed MHTSASM algorithm with others clustering techniques, such as K-M, TS, GA, and SA.

Firstly, there are three different standard test problems that were considered to compare the proposed algorithm with the other heuristics and meta-heuristics, such as the K-M algorithm [42], the TS method, GA, the SA method, and the non-smooth optimization technique for the minimum sum for the squares clustering problems (Algorithm 1) [43]. The German towns clustering test, which uses Cartesian coordinates, which has 59 towns and 59 records with two attributes, was used. Secondly, another set including 89 postal zones in Bavaria, which includes 89 records and 3 attributes, was used. Thirdly, another set including 89 postal zones in Bavaria, which includes 89 records and 4 attributes, was used. The results for these datasets are presented in Section 4.2.1, First Clustering Test Problems Situation.

Secondly, another two test datasets were used to compare the proposed MHTSASM algorithm with K-M algorithm, j-means (J-M+) algorithms [25], and the variable neighborhood search, such as VNS, VNS+, and VNS-1 algorithms [22]. In addition, we compared MHTSASM with the Global K-means (G1) [24], Fast global K-means (G2) algorithm [38,39], and Algorithm 1 [40,44]. These datasets are called (1) the first Germany postal zones dataset; and (2) the Fisher's iris dataset that includes 150 instances and 4 attributes [45–47] used in order to evaluate the proposed MHTSASM algorithm with others clustering techniques, such as K-M, J-M+, VNS, VNS+, VNS-1, G1, and G2. The results for these dataset is presented in Section 4.2.2, Second Clustering Test Problems Situation.

Thirdly, another five datasets called Iris, Glass, Cancer, Contraceptive Method Choice (CMC) and Wine as the examples of low, medium, and high dimensions data, respectively, were used to compare the proposed MHTSASM algorithm with K-harmonic means (KHM), particle swarm optimization (PSO), hybrid clustering method based on KHM, and PSO, called the PSOKHM algorithm [25] ACO and the hybrid clustering method based on ACO and KHM called the ACOKHM algorithm [26,27]. The results for these dataset is presented in Section 4.2.3, Third Clustering Test Problems Situation.

Table 2 summarizes the characteristics of these datasets [47]. Furthermore, there are several methods called (1) KHM; (2) PSO; and (3) PSOKHM algorithms [25], which were selected in order to compare their performances. In addition, other method called (1) KHM; (2) ACO; and (3) ACOKHM algorithms [27,28] were selected for the same purpose.

Table 2. Datasets characteristics.

Data Set Name	No. of Classes	No. of Features	Data Set Size	Classes Size in Parentheses
Iris	3	4	150	(50, 50, 50)
Glass	6	9	214	(70, 17, 76, 13, 9, 29)
Cancer	2	9	683	(444, 239)
CMC	3	9	1473	(629, 334, 510)
Wine	3	13	178	(59, 71, 48)

Moreover, there are two characteristics for each dataset that can positively affect the clustering algorithms' performances: (1) the instances which are the data records numbers; and (2) the features which are the data attributes numbers. The list of parameter values for the proposed MHTSASM algorithm is presented in Table 3 [28].

Table 3. Parameters setting.

Parameters	Definition	Value
El	Largest number of solution stored in elite list	5
$\mathrm{ASM_v}$	Number of visits of each partition	9
$\mathrm{ASM_u}$	Number of visit of each partition with improvement	300

4.2. Clustering Test Problems Situation Results

Tables 4 and 5 report the best value for the global minimum. The values are given as $\mathrm{nf(x^*)}$, where $\mathrm{x^*}$ is the local minimizer, and n is the instances number [28]. Moreover, the error E percentage of every algorithm is shown and calculated using Equation (5).

$$\mathrm{E} \; = \frac{\bar{\mathrm{f}} - \; \mathrm{f_{opt}}}{\mathrm{f_{opt}}} * 100 \tag{5}$$

where $\bar{\mathrm{f}}$ represents the algorithm finding solution, and fopt represents the best known solution.

Table 4. Results of first problems situation.

K	Dataset Name	f_{opt}	K-M	TS	GA	SA	Algorithm 1	MHTSASM
2	German towns	0.121426×10^6	0.00	0.00	0.00	0.00	0.00	0.00
	first Bavarian postal zones	0.60255×10^{12}	7.75	0.00	0.00	0.00	0.00	0.00
	second Bavarian postal zones	0.199080×10^{11}	144.25	0.00	144.25	144.25	144.25	144.25
3	German towns	0.77009×10^5	1.45	0.00	0.00	0.29	0.29	0.00
	first Bavarian postal zones	0.29451×10^{12}	23.48	23.48	23.48	23.48	0.00	0.00
	second Bavarian postal zones	0.173987×10^{11}	106.79	0.00	0.00	77.77	0.00	0.00
4	German towns	0.49601×10^5	0.55	0.00	0.00	0.00	0.00	0.00
	first Bavarian postal zones	0.10447×10^{12}	166.88	18.14	0.00	0.39	0.00	0.00
	second Bavarian postal zones	0.755908×10^{10}	303.67	0.00	0.00	9.13	0.00	0.00
5	German towns	0.39453×10^5	2.75	0.00	0.15	0.15	0.15	0.00
	first Bavarian postal zones	0.59762×10^{11}	335.32	33.35	0.00	40.32	0.00	0.00
	second Bavarian postal zones	0.540379×10^{10}	446.13	15.76	15.76	18.72	0.00	0.00

Table 5. Results of second problems situation.

K	Dataset Name	f_{opt}	K-M	J-M+	VNS	VNS-1	G1	G2	Algorithm 1	MHTSASM
2	first Germany postal zones	0.60255×10^{12}	7.75	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Fisher's iris	152.35	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.01
3	first Germany postal zones	0.29451×10^{12}	23.40	0.00	7.04	0.00	0.00	0.00	0.00	0.00
	Fisher's iris	78.851	13.35	0.00	0.00	0.00	0.00	0.01	0.00	0.11
4	first Germany postal zones	0.10447×10^{12}	156.17	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Fisher's iris	57.228	11.26	2.40	0.00	0.00	0.00	0.00	0.00	0.16
5	first Germany postal zones	0.59762×10^{11}	315.28	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	Fisher's iris	46.446	13.83	3.97	1.46	0.00	0.00	0.06	0.00	0.19
6	first Germany postal zones	0.35909×10^{11}	531.44	27.70	11.06	0.00	0.00	0.00	0.00	0.00
	Fisher's iris	39.040	16.21	4.26	0.01	0.00	0.00	0.07	0.00	0.00
7	first Germany postal zones	0.21983×10^{11}	832.60	44.00	7.07	0.00	0.00	0.00	1.48	0.00
	Fisher's iris	34.298	17.43	2.86	0.00	0.00	0.02	1.55	1.34	0.00
8	first Germany postal zones	0.13385×10^{11}	1239.64	0.24	0.00	0.00	0.00	0.00	0.00	0.00
	Fisher's iris	29.989	20.81	2.76	0.02	0.00	0.01	0.25	0.00	0.00
9	first Germany postal zones	0.84237×10^{10}	1697.17	28.59	0.00	0.00	0.00	0.00	0.00	0.00
	Fisher's iris	27.786	18.80	1.62	0.00	0.00	0.01	0.82	1.37	0.23
10	first Germany postal zones	0.64465×10^{10}	1638.30	0.16	0.00	0.00	0.00	0.00	11.32	0.00
	Fisher's iris	25.834	17.43	2.84	0.42	0.00	0.51	1.00	0.00	0.28

4.2.1. First Clustering Test Problems Situation

The numerical experimental results in Table 4 were taken for seven runs. In these tables, K refers to the number of clusters, and f_{opt} refers to the best identified value.

The proposed MHTSASM algorithm can reach the best results compared with all other clusters techniques for the German towns dataset, as shown in Table 4. The MHTSASM, TS, and GA results were the same. SA and Algorithm 1 gave the same results for all clusters.

Furthermore, the proposed MHTSASM algorithm achieved the best results for all numbers of clusters for the first Bavarian postal zones dataset, as shown in Table 4. The results from Algorithm 1 and MHTSASM were the same. Cluster three gave bad results for K-M, TS, GA, and SA. Cluster four gave better results for K-M, TS, GA, and SA than cluster three. Furthermore, GA gave good results except for cluster three.

The results presented in Table 4 show that TS achieved the best results for all numbers of clusters except for clusters five, where it gave a bad result for the second Bavarian postal zones dataset. The results from Algorithm 1 and the MHTSASM were the same for all clusters. Cluster two gave bad results for all algorithms except TS. Clusters three and four gave good results for the MHTSASM, Algorithm 1, and GA but bad results for SA and K-M. Cluster five gave good results for the proposed MHTSASM algorithm and Algorithm 1 but bad for K-M, TS, GA, and SA.

The results indicated that the proposed MHTSASM algorithm could reach better or very similar solutions to those found using the global optimization methods. Therefore, the proposed MHTSASM algorithm can deeply compute the local minimum in the clustering problem for the objective function.

4.2.2. Second Clustering Test Problems Situation

Another two datasets called (1) the first Germany postal zones dataset, and (2) the Fisher´s iris dataset that includes 150 instances and 4 attributes [8,21] were used in order to evaluate the proposed MHTSASM algorithm with other clustering techniques, such as K-M, J-M+, VNS, VNS+, VNS-1, G1, and G2. The numerical experimental results for this evaluation is presented in Table 5, where K refers to the number of clusters, and f_{opt} refers to the best identified value [28].

The average results for 10 restarts for KM, J-M+, and VNS were used, as seen in Table 5. The results of the first Germany postal zones dataset indicated that the proposed MHTSASM algorithm achieved better results than those for K-M, J-M+, VNS, G1, G2, and Algorithm 1. The proposed MHTSASM algorithm performed the same as VNS-1.

As shown in Table 5, the results of the Fisher´s iris dataset indicated that the proposed MHTSASM algorithm achieved better performance than the K-M technique for all numbers of clusters except for K = 2 and better results than J-M+ for K = 2, 3 [28]. The MHTSASM gave better results than G2 for K = 6, 7, 8, 9, 10. The proposed MHTSASM algorithm and VNS-1 gave results for K = 6, 7, 8 better than all other algorithms. Algorithm 1 and G1 were quite similar; they gave better results than the proposed MHTSASM algorithm for K = 2, 3, 4, 5. VNS gave better results than the MHTSASM for K = 2, 3, 4, 9. Compared with the Algorithm 1, the G1 technique achieved similar or better performance for all K values excluding K = 10. The proposed MHTSASM algorithm gave better results than Algorithm 1 for K = 7, 9. It gave better results than G2 for K = 6, 7, 8, 9, 10. It gave better results than G1 for K = 7, 8, 10. In addition, it gave better results than VNS for K = 6, 8, 10. The VNS-1 can be considered as the best technique for the Fisher´s iris dataset, as it could achieve the best identified optimal values. Hence, the results indicated that the results deviation found using the proposed MHTSASM algorithm from the comprehensive minimum was equal to or less than zero for all K.

4.2.3. Third Clustering Test Problems Situation

Another five datasets, known as Iris, Glass, Cancer, Contraceptive Method Choice (CMC), and Wine were employed to validate the proposed MHTSASM algorithm. Furthermore, there were several methods called (1) KHM, (2) PSO, and (3) PSOKHM algorithm [25] that were selected in order to compare their performances [28]. In addition, methods called (1) KHM, (2) ACO, and (3) ACOKHM algorithm [26,27] were selected for the same purpose.

The clustering methods quality, which could be measured using the F-measure criteria, was compared. F-measure is an external quality measure applicable for quality evaluation tasks [48–52]. It uses the ideas of precision and recall from information retrieval [53–57]. It finds how each point is clustered correctly relative to its original class. Every class (i) assumed using the benchmark dataset class labels is observed as n_i items set chosen for the query. Every cluster (j) created using the data cluster algorithms is observed as n_i items set for the query retrieval. Let n_{ij} denotes the class i element numbers inside cluster j. For every class i inside cluster j, the precision and recall are then defined as shown in Equation (6):

$$P_{(i,j)} = \frac{n_{ij}}{n_j},\ r_{(i,j)} = \frac{n_{ij}}{n_i} \tag{6}$$

and the corresponding value under the F-measure is calculated by Equation (7):

$$F_{(i,j)} = \frac{\left(b^2+1\right) * P_{(i,j)} * r_{(i,j)}}{b^2 * P_{(i,j)} + r_{(i,j)}} \tag{7}$$

where we are assuming that (b = 1) in order to gain equal weighting for $r_{(i,j)}$ and $p_{(i,j)}$. The total F-measure for the dataset of size n can be calculated using Equation (8):

$$F = \sum_i \frac{n_i}{n} \max_j \left\{F_{(i,j)}\right\} \tag{8}$$

The results in Table 6 show the quality of clustering evaluated using the F-measure over five real datasets. The results show standard deviations and means for 10 independent runs.

Table 6. Results of KHM, PSO, PSOKHM, and the proposed MHTSASM algorithm when p = 2.5, 3, and 3.5, respectively.

p No	Dataset Name	KHM	PSO	PSOKHM	MHTSASM
2.5	Iris	0.750	0.711	0.753	0.892
	Glass	0.421	0.387	0.424	0.548
	Cancer	0.829	0.819	0.829	0.872
	CMC	0.335	0.298	0.333	0.404
	Wine	0.516	0.512	0.516	0.715
3	Iris	0.744	0.740	0.744	0.892
	Glass	0.422	0.378	0.427	0.548
	Cancer	0.834	0.817	0.834	0.872
	CMC	0.303	0.250	0.303	0.404
	Wine	0.538	0.519	0.553	0.715
3.5	Iris	0.770	0.660	0.762	0.892
	Glass	0.396	0.373	0.396	0.548
	Cancer	0.832	0.820	0.835	0.872
	CMC	0.332	0.298	0.332	0.404
	Wine	0.502	0.530	0.535	0.715

The results of numerical experiments are presented in Table 6 for KHM, PSO, and PSOKHM when p = 2.5 and indicate that the proposed MHTSASM algorithm performed better than results for KHM, PSO, and PSOKHM for all datasets [28]. Furthermore, the PSOKHM performed better than results for KHM and PSO for datasets Iris and Glass. KHM performed better than results for PSO and PSOKHM for the CMC dataset. PSOKHM and KHM performed the same for datasets Wine and Cancer.

The results of numerical experiments are presented in Table 6 for KHM, PSO, and PSOKHM when p = 3 and show that the proposed MHTSASM algorithm performed better than results for KHM, PSO, and PSOKHM for all datasets. The PSOKHM performed better than results for KHM and PSO for datasets Glass and Wine. The PSOKHM and KHM performed the same for datasets Iris, Cancer, and CMC.

The results of numerical experiments are presented in Table 6 for KHM, PSO, and PSOKHM when p = 3.5 and show that the proposed MHTSASM algorithm performed better than results for KHM, PSO, and PSOKHM for all datasets. PSOKHM performed better than results for KHM and PSO for datasets Cancer and Wine. The KHM performed better than results for PSO and PSOKHM for the Iris dataset. The PSOKHM and KHM performed the same for datasets Glass and CMC.

4.3. Discussion

As shown in Table 4, the proposed MHTSASM algorithm could reach the best results compared with all others cluster techniques for the German towns dataset. Furthermore, it achieved the best results for all numbers of clusters for the first Bavarian postal zones dataset. TS achieved the best results for all numbers of clusters except cluster five, where it gave a bad result for the second Bavarian postal zones dataset. Cluster three gave bad results for K-M [24], TS, GA, and SA [23]. Cluster four gave better results for K-M, TS, GA, and SA than cluster three. Furthermore, GA gave good results except for cluster three. The results indicated that the proposed MHTSASM algorithm could reach better or very similar solutions to those found using the global optimization methods. Therefore, the proposed MHTSASM algorithm could deeply compute the local minimum in the clustering problem for the objective function.

As shown in Table 5, the results of the Fisher´s iris dataset indicated that the proposed MHTSASM algorithm achieved better performance than the K-M [42] technique for all numbers of clusters, except for K = 2, and better results than J-M+ for K = 2, 3. Hence, the results indicate that the results deviation found using the proposed MHTSASM algorithm from the comprehensive minimum is equal to or less than zero for all K.

As shown in Table 6, the results of numerical experiments indicate that the proposed MHTSASM algorithm performed better than results for KHM [25], PSO, and PSOKHM [26,27] for all datasets.

5. Conclusions

Clustering is a common data analysis and data mining problem. It aims to group similar data objects into sets of disjoint classes. Symmetry can be considered as a pre-attentive feature which that improves the shapes and objects, as well as reconstruction and recognition. The symmetry-based clustering methods search for clusters that are symmetric with respect to their centers. In addition, the K-M algorithm is one of the most common clustering methods. It can be easily implemented and works faster in most conditions. However, it is sensitively initialized and it can be easily trapped in the local targets. The TS algorithm is a stochastic global optimization technique. A new algorithm using the TS with K-M clustering, called MHTSASM, was presented in this paper. The MHTSASM algorithm fully uses the merits of both TS and K-M algorithms. It uses TS to make economic exploration for data with the help of ASM. It uses different strategies of TS, such as the intensification and diversification. The proposed MHTSASM algorithm performance is compared with multiple clustering techniques based on both optimization and meta-heuristics. The experimental results ensure that the MHTSASM overcomes initialization sensitivity of K-M and reaches the global optimal effectively. However, dealing and identifying clusters with non-convex shapes is one of the paper's limitation, and it will be one of the future research direction.

Funding: This research is funded by Deanship of Scientific Research at Umm Al-Qura University, Grant Code: 22UQU4281768DSR02.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The study did not report any data.

Acknowledgments: The author would like to thank the Deanship of Scientific Research at Umm Al-Qura University for supporting this work by Grant Code: (22UQU4281768DSR02).

Conflicts of Interest: The author declares no conflict of interest.

References

1. Bharany, S.; Sharma, S.; Badotra, S.; Khalaf, O.I.; Alotaibi, Y.; Alghamdi, S.; Alassery, F. Energy-Efficient Clustering Scheme for Flying Ad-Hoc Networks Using an Optimized LEACH Protocol. *Energies* **2021**, *14*, 6016. [CrossRef]
2. Li, G.; Liu, F.; Sharma, A.; Khalaf, O.I.; Alotaibi, Y.; Alsufyani, A.; Alghamdi, S. Research on the natural language recognition method based on cluster analysis using neural network. *Math. Probl. Eng.* **2021**, *2021*, 9982305. [CrossRef]
3. Jinchao, J.; Li, W.P.Z.; He, F.; Feng, G.; Zhao, X. Clustering Mixed Numeric and Categorical Data with Cuckoo Search. *IEEE Access* **2020**, *8*, 30988–31003.
4. Su, M.C.; Chou, C.H. A modified version of the K-means algorithm with a distance based on cluster symmetry. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 674–680. [CrossRef]
5. Vijendra, S.; Laxman, S. Symmetry based automatic evolution of clusters: A new approach to data clustering. *Comput. Intell. Neurosci.* **2015**, *2015*, 796276. [CrossRef] [PubMed]
6. Khan, S.S.; Ahmad, A. Cluster center initialization algorithm for K-means clustering. *Pattern Recognit. Lett.* **2004**, *25*, 1293–1302. [CrossRef]
7. Alotaibi, Y.; Subahi, A.F. New goal-oriented requirements extraction framework for e-health services: A case study of diagnostic testing during the COVID-19 outbreak. *Bus. Process Manag. J.* **2021**, *28*. [CrossRef]
8. Subramani, N.; Mohan, P.; Alotaibi, Y.; Alghamdi, S.; Khalaf, O.I. An Efficient Metaheuristic-Based Clustering with Routing Protocol for Underwater Wireless Sensor Networks. *Sensors* **2022**, *22*, 415. [CrossRef] [PubMed]
9. Huizhen, Z.; Liu, F.; Zhou, Y.; Zhang, Z. A hybrid method integrating an elite genetic algorithm with tabu search for the quadratic assignment problem. *Inf. Sci.* **2020**, *539*, 347–374.
10. Suganthi, S.; Umapathi, N.; Mahdal, M.; Ramachandran, M. Multi Swarm Optimization Based Clustering with Tabu Search in Wireless Sensor Network. *Sensors* **2022**, *22*, 1736. [CrossRef] [PubMed]
11. Kareem, S.S.; Mostafa, R.R.; Hashim, F.A.; El-Bakry, H.M. An Effective Feature Selection Model Using Hybrid Metaheuristic Algorithms for IoT Intrusion Detection. *Sensors* **2022**, *22*, 1396. [CrossRef] [PubMed]
12. Rajendran, S.; Khalaf, O.I.; Alotaibi, Y.; Alghamdi, S. MapReduce-based big data classification model using feature subset selection and hyperparameter tuned deep belief network. *Sci. Rep.* **2021**, *11*, 24138.
13. Liqin, Y.; Cao, F.; Gao, X.Z.; Liu, J.; Liang, J. k-Mnv-Rep: A k-type clustering algorithm for matrix-object data. *Inf. Sci.* **2020**, *542*, 40–57.
14. Mansouri, N.; Javidi, M.M. A review of data replication based on meta-heuristics approach in cloud computing and data grid. *Soft Comput.* **2020**, *24*, 14503–14530. [CrossRef]
15. Li, J.Q.; Duan, P.; Cao, J.; Lin, X.P.; Han, Y.Y. A hybrid Pareto-based tabu search for the distributed flexible job shop scheduling problem with E/T criteria. *IEEE Access* **2018**, *6*, 58883–58897. [CrossRef]
16. Amir, A.; Khan, S.S. Survey of state-of-the-art mixed data clustering algorithms. *IEEE Access* **2019**, *7*, 31883–31902.
17. Laith, A. Group search optimizer: A nature-inspired meta-heuristic optimization algorithm with its results, variants, and applications. *Neural Comput. Appl.* **2020**, *33*, 2949–2972.
18. Alotaibi, Y. A New Database Intrusion Detection Approach Based on Hybrid Meta-Heuristics. *CMC-Comput. Mater. Contin.* **2021**, *66*, 1879–1895. [CrossRef]
19. Alotaibi, Y.; Malik, M.N.; Khan, H.H.; Batool, A.; ul Islam, S.; Alsufyani, A.; Alghamdi, S. Suggestion Mining from Opinionated Text of Big Social Media Data. *CMC-Comput. Mater. Contin.* **2021**, *68*, 3323–3338. [CrossRef]
20. Kosztyán, Z.T.; Telcs, A.; Abonyi, J. A multi-block clustering algorithm for high dimensional binarized sparse data. *Expert Syst. Appl.* **2022**, *191*, 116219. [CrossRef]
21. Chou, X.; Gambardella, L.M.; Montemanni, R. A tabu search algorithm for the probabilistic orienteering problem. *Comput. Oper. Res.* **2021**, *126*, 105107. [CrossRef]
22. Yiyong, X.; Huang, C.; Huang, J.; Kaku, I.; Xu, Y. Optimal mathematical programming and variable neighborhood search for k-modes categorical data clustering. *Pattern Recognit.* **2019**, *90*, 183–195.
23. Amit, S.; Prasad, M.; Gupta, A.; Bharill, N.; Patel, O.P.; Tiwari, A.; Er, M.J.; Ding, W.; Lin, C.T. A review of clustering techniques and developments. *Neurocomputing* **2017**, *267*, 664–681.
24. Rabbani, M.; Mokhtarzadeh, M.; Manavizadeh, N. A constraint programming approach and a hybrid of genetic and K-means algorithms to solve the p-hub location-allocation problems. *Int. J. Manag. Sci. Eng. Manag.* **2021**, *16*, 123–133. [CrossRef]
25. Ali, N.A.; Han, F.; Ling, Q.H.; Mehta, S. An improved hybrid method combining gravitational search algorithm with dynamic multi swarm particle swarm optimization. *IEEE Access* **2019**, *7*, 50388–50399.
26. Amir, A.; Hashmi, S. K-Harmonic means type clustering algorithm for mixed datasets. *Appl. Soft Comput.* **2016**, *48*, 39–49.
27. Sina, K.; Adibeig, N.; Shanehbandy, S. An improved overlapping k-means clustering method for medical applications. *Expert Syst. Appl.* **2017**, *67*, 12–18.
28. Adil, B.M.; Yearwood, J. A new nonsmooth optimization algorithm for minimum sum-of-squares clustering problems. *Eur. J. Oper. Res.* **2006**, *170*, 578–596.
29. Mustafi, D.; Sahoo, G. A hybrid approach using genetic algorithm and the differential evolution heuristic for enhanced initialization of the k-means algorithm with applications in text clustering. *Soft Comput.* **2019**, *23*, 6361–6378. [CrossRef]
30. Rout, R.; Parida, P.; Alotaibi, Y.; Alghamdi, S.; Khalaf, O.I. Skin Lesion Extraction Using Multiscale Morphological Local Variance Reconstruction Based Watershed Transform and Fast Fuzzy C-Means Clustering. *Symmetry* **2021**, *13*, 2085. [CrossRef]

31. Mohan, P.; Subramani, N.; Alotaibi, Y.; Alghamdi, S.; Khalaf, O.I.; Ulaganathan, S. Improved Metaheuristics-Based Clustering with Multihop Routing Protocol for Underwater Wireless Sensor Networks. *Sensors* **2022**, *22*, 1618. [CrossRef] [PubMed]
32. Yan, Y.; He, M.; Song, L. Evaluation of Regional Industrial Cluster Innovation Capability Based on Particle Swarm Clustering Algorithm and Multi-Objective Optimization. Available online: https://www.springerprofessional.de/en/evaluation-of-regional-industrial-cluster-innovation-capability-/19688906 (accessed on 20 January 2022).
33. Chen, L.; Guo, Q.; Liu, Z.; Zhang, S.; Zhang, H. Enhanced synchronization-inspired clustering for high-dimensional data. *Complex Intell. Syst.* **2021**, *7*, 203–223. [CrossRef]
34. Sung-Soo, K.; Baek, J.Y.; Kang, B.S. Hybrid simulated annealing for data clustering. *J. Soc. Korea Ind. Syst. Eng.* **2017**, *40*, 92–98.
35. Yi-Tung, K.; Zahara, E.; Kao, I.W. A hybridized approach to data clustering. *Expert Syst. Appl.* **2008**, *34*, 1754–1762.
36. Ezugwu, A.E.; Adeleke, O.J.; Akinyelu, A.A.; Viriri, S. A conceptual comparison of several metaheuristic algorithms on continuous optimisation problems. *Neural Comput. Appl.* **2020**, *32*, 6207–6251. [CrossRef]
37. Alsufyani, A.; Alotaibi, Y.; Almagrabi, A.O.; Alghamdi, S.A.; Alsufyani, N. Optimized Intelligent Data Management Framework for a Cyber-Physical System for Computational Applications. Available online: https://link.springer.com/article/10.1007/s40747-021-00511-w (accessed on 1 January 2022).
38. Pasi, F.; Sieranoja, S. How much can k-means be improved by using better initialization and repeats? *Pattern Recognit.* **2019**, *93*, 95–112.
39. Xiao-Dong, W.; Chen, R.C.; Yan, F.; Zeng, Z.Q.; Hong, C.Q. Fast adaptive K-means subspace clustering for high-dimensional data. *IEEE Access* **2019**, *7*, 42639–42651.
40. Asgarali, B.; Hatamlou, A. An efficient hybrid clustering method based on improved cuckoo optimization and modified particle swarm optimization algorithms. *Appl. Soft Comput.* **2018**, *67*, 172–182.
41. Yinhao, L.; Cao, B.; Rego, C.; Glover, F. A Tabu Search based clustering algorithm and its parallel implementation on Spark. *Appl. Soft Comput.* **2018**, *63*, 97–109.
42. Alotaibi, Y. A New Secured E-Government Efficiency Model for Sustainable Services Provision. *J. Inf. Secur. Cybercrimes Res.* **2020**, *3*, 75–96. [CrossRef]
43. Costa, L.R.; Aloise, D.; Mladenović, N. Less is more: Basic variable neighborhood search heuristic for balanced minimum sum-of-squares clustering. *Inf. Sci.* **2017**, *415*, 247–253. [CrossRef]
44. Gribel, D.; Vidal, T. HG-means: A scalable hybrid genetic algorithm for minimum sum-of-squares clustering. *Pattern Recognit.* **2019**, *88*, 569–583. [CrossRef]
45. Safwan, C.F. A Genetic Algorithm that Exchanges Neighboring Centers for Fuzzy c-Means Clustering. Ph.D. Thesis, Nova Southeastern University, Lauderdale, FL, USA, 2012.
46. Fisher, R.A. The use of multiple measurements in taxonomic problems. *Ann. Eugen.* **1936**, *7*, 179–188. [CrossRef]
47. Alotaibi, Y. Automated Business Process Modelling for Analyzing Sustainable System Requirements Engineering. In Proceedings of the 2020 6th International Conference on Information Management (ICIM), London, UK, 27–29 March 2020; pp. 157–161.
48. Katebi, J.; Shoaei-parchin, M.; Shariati, M.; Trung, N.T.; Khorami, M. Developed comparative analysis of metaheuristic optimization algorithms for optimal active control of structures. *Eng. Comput.* **2020**, *36*, 1539–1558. [CrossRef]
49. Majid, E.; Shahmoradi, H.; Nemati, F. A new preference disaggregation method for clustering problem: DISclustering. *Soft Comput.* **2020**, *24*, 4483–4503.
50. Rawat, S.S.; Alghamdi, S.; Kumar, G.; Alotaibi, Y.; Khalaf, O.I.; Verma, L.P. Infrared Small Target Detection Based on Partial Sum Minimization and Total Variation. *Mathematics* **2022**, *10*, 671. [CrossRef]
51. Ashraf, F.B.; Matin, A.; Shafi, M.S.R.; Islam, M.U. An Improved K-means Clustering Algorithm for Multi-dimensional Multi-cluster data Using Meta-heuristics. In Proceedings of the 2021 24th International Conference on Computer and Information Technology (ICCIT), Dhaka, Bangladesh, 18–20 December 2021; pp. 1–6.
52. Zhao, L.; Wang, Z.; Zuo, Y.; Hu, D. Comprehensive Evaluation Method of Ethnic Costume Color Based on K-Means Clustering Method. *Symmetry* **2021**, *13*, 1822. [CrossRef]
53. Beldjilali, B.; Benadda, B.; Sadouni, Z. Vehicles Circuits Optimization by Combining GPS/GSM Information with Metaheuristic Algorithms. Rom. *J. Inf. Sci. Technol.* **2020**, *23*, T5–T17.
54. Zamfirache, I.A.; Precup, R.E.; Roman, R.C.; Petriu, E.M. Reinforcement Learning-based control using Q-learning and gravitational search algorithm with experimental validation on a nonlinear servo system. *Inf. Sci.* **2022**, *583*, 99–120. [CrossRef]
55. Pozna, C.; Precup, R.E.; Horvath, E.; Petriu, E.M. Hybrid Particle Filter-Particle Swarm Optimization Algorithm and Application to Fuzzy Controlled Servo Systems. *IEEE Trans. Fuzzy Syst.* **2022**, *1*. [CrossRef]
56. Thakur, N.; Han, C.Y. A Study of Fall Detection in Assisted Living: Identifying and Improving the Optimal Machine Learning Method. *J. Sens. Actuator Netw.* **2021**, *10*, 39. [CrossRef]
57. Jayapradha, J.; Prakash, M.; Alotaibi, Y.; Khalaf, O.I.; Alghamdi, S. Heap Bucketization Anonymity-An Efficient Privacy-Preserving Data Publishing Model for Multiple Sensitive Attributes. *IEEE Access* **2022**, *1*. [CrossRef]