Journal of Applied Research and Technology

www.jart.icat.unam.mx

Journal of Applied Research and Technology 20 (2022) 638-651

Original

Clustering algorithms with prediction of the optimal number of clusters

A. Agárdi* • L. Kovács

The University of Miskolc, 3515 Miskolc-Egyetemváros, Hungary

Received 06 23 2020; accepted 09 20 2021
Available 12 31 2022

Abstract: Clustering is a widely used technique for grouping of objects. The objects, which are similar to each other, should be in the same cluster. One disadvantage of general clustering algorithms is that the user must specify the number of clusters in advance, as input parameter. This is a major drawback since it is possible that the user cannot specify the number of clusters correctly, and the algorithm thus creates a clustering that puts very different elements into the same cluster. The aim of this paper is to present our representation and evaluation technique to determine the optimal cluster count automatically. With this technique, the algorithms themselves determine the number of clusters. In this paper, first, the classical clustering algorithms are introduced; then, the construction and improvement algorithms and then our representation and evaluation method are presented. Then the performance of the algorithms with the test results are compared.

Keywords: clustering, optimal number of clusters

*Corresponding author.
E-mail address: agardianita@iit.uni-miskolc.hu (A. Agárdi).
Peer Review under the responsibility of Universidad Nacional Autónoma de México.

1. Introduction

Clustering is a data mining algorithm whose goal is the grouping of objects. Objects that are similar to each other should belong to the same cluster, and objects that are different from each other should belong to different clusters. There are several clustering algorithms in the literature, for example, partitioning methods, hierarchical methods, density-based method, the neural network-based method and the grid method, graph theory-based method, fuzzy methods. The disadvantage of clustering algorithms is that the algorithms also expect a cluster number as input. The optimal number of clusters can often not be determined by the user, therefore, an algorithm is needed to help the user determine the cluster number. In this paper, we present a representation mode and its evaluation by which the algorithms themselves determine the optimal cluster number. The article is structured as follows: Section 2 contains a brief introduction of clustering algorithms and literature review. Section 3 contains the traditional clustering algorithms, section 4 includes a cluster validation index (Silhouette index). Section 5 contains the construction algorithms, which are the followings: Nearest Neighbor, Nearest Insertion, Cheapest Insertion, Arbitrary Insertion, Farthest Insertion, Greedy. Section 6 includes the improvement algorithms, such as Genetic algorithm, Tabu Search and Particle Swarm Optimization. After that our representation and evaluation is detailed. In section 8 test results are detailed. After that conclusion remarks are made.

2. Clustering

Clustering is a widely used technique for grouping of elements. If two elements are similar, then belong to the same cluster. If they are different, they belong to different clusters. Optimal clustering is a difficult task because there are many ways to group a dataset.

There are lots of types of clustering algorithms, for example (Xu & Wunsch, 2005):

- Partitioning methods: elements are divided into k groups. Each group contains at least one element. After an initial clustering, a re-partitioning follows. At this point, the individual points may be placed in other clusters. The process ends when the elements move slightly (clustering changes only slightly). For example K-Means, Partitioning Around Medoid (PAM)
- Hierarchical methods: clusters can be represented by a dendrogram. There are two main methods of hierarchical clustering: the divisive and the agglomerative methods. The agglomerative methods are Single linkage, complete linkage, group average linkage, median linkage, centroid linkage, Ward's method etc. The divisive methods are divisive analysis (DIANA), monothetic analysis (MONA).
- Other clustering methods include the density-based method, the neural network-based method and the grid method, graph theory-based method, fuzzy methods.

Some publications have been published in recent years that using metaheuristic algorithms for clustering. The Fast Genetic K-means Clustering Algorithm (Lu et al.,2004) combines the K-Means algorithm and the Genetic Algorithm. The algorithm applies the mutation, selection and crossover techniques (based on the Genetic Algorithm) and also has a K-Means operator. The K-Means operator (one step of the classical K-means algorithm) is the following: the elements are re-partitioned based to the closest cluster centroid. Another clustering analysis with the Genetic Algorithm is introduced in paper (Hruschka & Ebecken, 2003), where also the classical genetic operators are used. The objective function is based on the Average Silhouette Width. The author of paper (Maulik & Bandyopadhyay, 2000) is also used the Genetic Algorithm for the clustering of a data object. The objective function is the minimization of the distance of the objects to their cluster centroids. Their fitness calculation is the following: clusters are formed according to the centers encoded in the chromosome, after the clustering, the cluster centers will be the mean points of the respective clusters. Over the years many crossover and mutation techniques are developed to the Clustering Genetic Algorithm, for example, the one-point mutation, biased one-point mutation, which change the value of a center randomly picked (Kudova, 2007). The K-means mutation, which performs several steps of the k-means algorithm. (Kudova, 2007). The cluster addition and cluster removal modify the number of clusters (adds one center chosen randomly from the data set and deletes one randomly chosen center). (Kudova, 2007) For the fitness function in paper (Kudova, 2007) also the Silhouette is used.

Over the years the Particle Swarm Optimization (PSO) is also applied to clustering data. (Li & Yang, 2009). The PSO is applied with Hierarchical Clustering method, called CPSO Algorithm. A hybrid K-Means PSO algorithm is applied in paper (Van der Merwe & Engelbrecht, 2003). In this case, the result of the K-Means algorithm is improved with the PSO algorithm. The objective function of the PSO algorithm is based on the sum of the average distance of the object to their cluster centroids. In paper (Chen & Ye, 2012) also the PSO algorithm is applied to the clustering of the dataset. In this paper the encoding is also presented, which is the following: the string of the particle contains the cluster centers (in the paper the x and y coordinates of the cluster centroids).

In the case of applying the Ant Colony Optimization (ACO) to clustering data objects, the objective function can be also the minimization of the distance between the cluster elements and the centroids. (Runkler, 2005) The representation of the solution can be a string, which elements are numbers. The numbers indicate the cluster-object assignment. If the string is for example 2,1,3,1, it means, that the first object belongs to cluster 2, the second object belongs to cluster 1, the third object belongs to cluster 3, and the fourth object belongs to cluster 1. (Shelokar et al., 2004)

In paper (Osman & Christofides, 1994) the objective function of the Simulated Annealing algorithm is the minimization of the sum of distances between the clusters. The Capacitated Clustering Problem (CCP) is also solved with the Simulated Annealing algorithm. In the case of CCP, each object has a weight, and each cluster has a given capacity which must not be exceeded by the total weight of objects in the cluster.

Cluster Analysis with K-Modes and K-Prototype Algorithms in presented in (Madhuri et al., 2014). The authors used Iris Data Set and Cholesterol Data Set for Incremental k-Means, Contact-Lens Data Set and Post-operative Data Set for Modified k-Modes, Blood Information Data Set and Weather Data Set for k-Prototypes .

Automatic clustering with Teaching Learning-Based Optimization (TLBO) is presented in paper (Murty, Naik et al., 2014). The efficiency of the TLBO is compared also with Particle Swarm Optimization (PSO), Differential Evolution (DE). The efficiency of the algorithms is compared with the following benchmark datasets: Iris Data, Wine Data, Breast Cancer Data, Glass Data and Vowel Data.

The purpose of cluster validation indices is to compare individual clusters with each other considering certain aspects. Several such indices have been published, but these indices are not suitable for data sets of any size, density, shape. The authors of (Murty, Murthy et al., 2014) have developed a validation index (Homogeneity Separateness) that is effective for clusters of any shape, size, and density. Some clustering problems and the algorithms that solve it are illustrated in Table 1.

3. Traditional clustering algorithms

In this section, the applied traditional clustering algorithms are presented based on the literature.

3.1. K-Means

This procedure belongs to a group of partitioning methods. First, the elements are clustered and then the elements move from the initial clusters to improve the quality of the clustering. The algorithm uses the SSE function. Here, the number of clusters (k) must be specified. (Wagstaff et al., 2001) Figure 1 illustrates the psedo code of the K-Means algorithm.

$$E(C) = \sum_{i=1}^{k} \sum_{u \in C_i} d(u, r(C_i))^2 \tag{1}$$

Table 1. Some clustering problems and the algorithms that solve it.

Article	Clustering algorithm	Problem
(Milano & Koumoutsakos, 2002)	Genetic Algorithm	cylinder drag optimization
(Doval et al., 1999)	Genetic Algorithm	software systems
(Scheunders, 1997)	Genetic Algorithm	color image quantization
(Cui et al., 2005)	Particle Swarm Optimization	document clustering
(Omran et al., 2006)	Particle Swarm Optimization	image segmentation
(Omran et al., 2004)	Particle Swarm Optimization	image classification
(Paoli et al., 2009)	Particle Swarm Optimization	hyperspectral images
(Kalyani & Swarup, 2011)	Particle Swarm Optimization	security assessment in power systems
(Chiu et al., 2009)	Particle Swarm Optimization	intelligent market segmentation system
(Yang et al., 2010)	Ant Colony Optimization	multipath routing protocol
(Gao et al., 2016)	Ant Colony Optimization	dynamic location routing problem
(Zhao et al., 2007)	Ant Colony Optimization	Image segmentation-based
(Chang, 1996)	Simulated Annealing	Chinese words
(França et al., 1999)	Tabu Search	capacitated clustering problem
(El Rhazi & Pierre at al., 2008)	Tabu Search	wireless sensor networks
(Kinney et al., 2007)	Tabu Search	unicost set covering problem
(Hoang et al., 2013)	Harmony Search	energy-efficient wireless sensor networks
(Forsati, 2008)	Harmony Search	web page clustering
(Hoang, 2010)	Harmony Search	wireless sensor networks
(Mahdavi & Abolhassani, 2009)	Harmony Search	document clustering

4. Hierarchical methods

These algorithms organize the clusters into a hierarchical data structure. There are two types of algorithms: bottom-up and top-down. At the bottom-up, each element is initially a cluster, and then each cluster is merged into a single cluster. The top-down procedure is just the opposite. Here at first, there is a single cluster that contains all the elements, and we continually divide the clusters. At the end of the procedure, each element means a separate cluster. The following hierarchical clustering is applied in this paper: (Murtagh, 1983; Olson, 1995)

Single Linkage: The distance of the two clusters is the distance between the nearest objects of the two clusters:

$$d_{min}\left(C_i, C_j\right) = \min_{u \in C_i, v \in C_j} d(u, v) \tag{2}$$

```
BEGIN PROCEDURE
    Step 1. Selecting randomly $k$ objects. Initially, these
    represent the center of the cluster.
    WHILE(cluster centers and their associated objects
    change) DO
        WHILE (not all objects have been selected) DO
            Step 2. The object is classified to the closest
            cluster center.
        END WHILE
    Step 3. Recalculating the cluster centers.
    END WHILE
END PROCEDURE
```

Figure 1. The pseudo code of the K-Means method (Wagstaff et al., 2001).

Complete Linkage: the distance of two clusters is the distance between the farthest objects of two clusters:

$$d_{max}\left(C_i, C_j\right) = \max_{u \in C_i, v \in C_j} d(u, v) \tag{3}$$

Average method: the distance of two clusters is the quotient of the sum of the distances between the objects of two clusters and the number of clusters:

$$d_{avg}\left(C_i, C_j\right) = \frac{1}{|C_i||C_j|} \sum_{u \in C_i, v \in C_j} d(u, v) \tag{4}$$

Centroid method: the distance of two clusters will be the distance of the center of two clusters:

$$d_{mean}\left(C_i, C_j\right) = d\left(\frac{1}{|C_i|}\sum_{u \in C_i} u, \frac{1}{|C_j|}\sum_{u \in C_j} v\right) \tag{5}$$

Ward method: merging the two clusters that cause the least-squares error increase:

$$d_{Ward}\left(C_i, C_j\right) = \sum_{u,v \in C_i \cup C_j} d^2(u, v) - \left(\sum_{u,v \in C_i} d^2(u, v) + \sum_{u,v \in C_j} d^2(u, v)\right) \tag{6}$$

5. Cluster validation index

Let $\{A\}$ be a partitioning, i is the index of the data point, and A_i denotes the container cluster of element i. The silhouette index of the object i is (Wang & Xu, 2019):

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{7}$$

where, $-1 \leq s(i) \leq 1$.

$a(i)$ is the average distance between the object i and other objects in cluster A_i, $b(i)$ is the average distance between the object i and all other cluster elements (except elements in cluster A). Thus $a(i)$ denotes compactness and $b(i)$ denotes separation.

- If $s(i)$ is large (close to 1) then the inner (within the cluster) difference is much smaller than the smallest outer cluster difference. Hence, we can say that object i. is well grouped.
- If $s(i) = 0$, or very close to zero, then $a(i)$ and $b(i)$ are nearly equal, then object i may belong to cluster A or B.
- If $s(i)$ is close to -1, then $a(i)$ is much greater than $b(i)$. This means that cluster B would be a better choice than cluster A.

With the silhouette method, we can measure the efficiency of the whole cluster result. For each element, $s(i)$ should be calculated and the results should be averaged.

The following conclusions can be made from the average silhouette:

- 0.5 or higher value: good clustering
- 0.25-0.5: the clustering method is good, but some object should be moved to another cluster
- Less than 0.25: Not good clustering

Thus, the higher the average silhouette, the better the clustering.

Therefore, the objective function of our improvement algorithms is the average silhouette value.

6. Tour-based construction clustering algorithm

Behind this approach is the shortest route path tour connects neighboring elements. The edge distance is usually small, it connects elements from the same cluster, but the length is large if the edge connects two distinct clusters. Testing the distance of the connecting edges of the optimal tour the edges with high lengths denote existence of the separate clusters.

The construction algorithms construct one possible solution. Running time is relatively low. These algorithms always take locally the best steps. Most of the time, the global optimum is not achieved by their exclusive usage. The construction algorithms are based on the Traveling Salesman Problem algorithms.

6.1. Nearest neighbor

The algorithm always selects the unselected object that is closest to the last selected object. The algorithm is fast and simple. Figure 2 illustrates the pseudo code of the Nearest Neighbor algorithm.

```
BEGIN PROCEDURE
    WHILE (not all objects have been selected) DO
        Step 1. Selecting an object randomly
        Step 2. Selecting the unselected object which is closest to the last selected object
    END WHILE
END PROCEDURE
```

Figure 2. The pseudo code of the Nearest Neighbor algorithm (Nilsson, 2003).

6.2. Nearest insertion

The algorithm belongs to the group of insertion heuristics. The algorithm always selects the unselected object that is closest to the "tour". The distance between the "tour" and an object is interpreted by the algorithm as the minimum distance between the objects in the tour. The pseudo code of the Nearest Insertion algorithm is illustrated in Figure 3.

```
BEGIN PROCEDURE
    Step 1. Randomly selecting an object, denoting it with $i$.
    Step 2. Selecting the object $r$ for which $c_{ir}$ is minimal then making $i - r - i$ „sub-tour”.
    WHILE (not all objects have been selected) DO
        Step 3. Selection Step: Selecting the object $r$ that has not yet been selected and is closest to any $j$ object in the "sub-tour".
        Step 4. Insertion Step: Searching for the $(i, j)$ object pair in the „sub-tour” where $c_{ir} + c_{rj} + c_{ij}$ is minimal. So we select the two adjacent objects that insert the object $r$ between them the insertion cost (cost of increasing the tour) will be minimal. The object $r$ is then inserted between $i$ and $j$.
    END WHILE
END PROCEDURE
```

Figure 3. The pseudo code of the Nearest Insertion algorithm (Golden et al., 1980).

6.3. Cheapest insertion

This algorithm also belongs to the group of insertion heuristics. The algorithm always selects the object with the least "insertion cost" into the “tour". Figure 4 illustrates the pseudo code of the Cheapest Insertion.

```
BEGIN PROCEDURE
    Step 1. Selecting an object randomly, indicated with $i$.
    Step 2. Taking the object $r$, for which $c_{ir}$ is minimal, and making an $i - r - i$ „sub-tour”.
    WHILE (not all objects have been selected) DO
        Step 3. Selection Step: Find the pair of $(i, j)$ objects in the "sub-path" and the object $r$ that is not in the "sub-path" which minimize the following amount: a $c_{ir} + c_{rj} + c_{ij}$.
        Step 4. Insertion Step: Object $r$ will be between the searched object $i$ and $j$.
    END WHILE
END PROCEDURE
```

Figure 4. The pseudo code of the Cheapest Insertion algorithm (Golden et al., 1980).

6.4. Arbitrary insertion

This algorithm also belongs to the group of Insertion Heuristics. The algorithm randomly selects the next object to be inserted into the "tour". Figure 5 presents the presudo code of the Arbitrary Insertion algorithm.

```
BEGIN PROCEDURE
    Step 1. Selecting an object randomly, indicated with $i$.
    Step 2. Selecting an object $r$, for which $c_{ir}$ is minimal, and making an $i - r - i$ „sub-tour”.
    WHILE (not all objects have been selected) DO
        Step 3. Selection Step: Taking randomly the object $r$ that is not already contained in the "sub-tour".
        Step 4. Insertion Step: Finding the $(i, j)$ pair of objects in the "sub-path" that minimizes the following amount: $c_{ir} + c_{rj} + c_{ij}$. So searching for two "adjacent" objects between inserting the object $r$ will have a minimal cost of insertion. Object $r$ will be placed between $i$ and $j$.
    END WHILE
END PROCEDURE
```

Figure 5. The pseudo code of the Arbitrary Insertion algorithm (Rosenkrantz et al., 1974).

6.5. Farthest insertion

This algorithm also belongs to the Insertion Heuristics group. The object that is farthest from the other objects is selected by the algorithm. Figure 6 illustrates the pseudo code of the Farthest Insertion algorithm.

```
BEGIN PROCEDURE
    Step 1. Selecting an object randomly, indicated with $i$.
    Step 2. Selecting an object $r$, for which $c_{ir}$ is minimal, and making an $i-r-i$ „sub-tour”.
    WHILE (not all objects have been selected) DO
        Step 3. Selection Step: Taking randomly the object $r$ that is not already contained in the "sub-tour" and is farthest from any object $j$
        Step 4.: Insertion Step: Selecting those $(i,j)$ object pair from the „sub-tour” which minimizes the following sum: $c_{ir}+c_{rj}+c_{ij}$. So considering the two objects that insert the object $r$ between them to minimize the cost of insertion.
    END WHILE
END PROCEDURE
```

Figure 6. The pseudo code of the Farthest Insertion algorithm (Golden et al., 1980).

6.6. Greedy

Each object builds the order from "edges" (pairs of objects) so that it always selects the shortest "edge" that has not yet been selected and does not form n vertex circles (n indicates the number of objects). Also, the degree of the “edge” should not be more than two. The pseudo code of the Greedy algorithm is illustrated in Figure 7.

```
BEGIN PROCEDURE
Step 1. The edges are sorted by their length.
WHILE ($n$ objects are not selected) DO
        Step 2. Selecting the shortest "edge" (object pair) that has not yet been selected and does not violate the above-mentioned conditions.
    END WHILE
END PROCEDURE
```

Figure 7. The pseudo code of the Greedy algorithm (Nilsson, 2003).

7. Tour improvement-based clustering algorithm

Behind this approach is also the fact, that the shortest route path tour connects neighboring elements. This method tries to improve an existing path by rearranging the order of the elements iteratively. Their running time can be high, and their exclusive usage does not lead to the global optimum in most of the cases.

7.1. Genetic algorithm

The algorithm models natural processes (evolution). The algorithm works with a population of solutions. The population consists of individuals. (Milano & Koumoutsakos, 2002) Individuals have fitness values. Usually, an individual with better fitness value is better against other individuals. The pseudo code of the Genetic Algorithm is presented in Figure 8.

```
BEGIN PROCEDURE
    Step1. Initialization of a population (with random individuals or with individuals generated with construction algorithms).
    Step 2. Calculation the fitness values for individuals.
    WHILE (termination condition is not met) DO
        Step 3. The transition of certain individuals unaltered to the new generation (elitism).
        Step 4. Crossing selected parent pairs.
        Step 5. Mutation of selected new individuals.
        Step 6. Evaluation of new individuals.
        Step 7. Upload the next generation with new individuals.
    END WHILE
END PROCEDURE
```

Figure 8. The pseudo code of the Genetic Algorithm (Whitley, 1994).

The first step is the initialization of the population. This process is usually done with randomly generated individuals. Then the fitness values of the individuals are calculated. Then the next population is created in a cycle while the termination condition is not met. The termination condition may be to achieve a certain iteration number or runtime. The next population is created by moving certain individuals unchanged, which is called elitism. The other elements are created with crossover and mutation techniques. We have used the 2-opt (Wu et al., 2007) as mutation, and the Partially Matched Crossover (PMX) (Lazzerini & Marcelloni, 2000) (Starkweather et al., 1991), the Order Crossover (OX) (Starkweather et al., 1991), and the Cycle Crossover (CX) (Starkweather et al., 1991) as crossover operators.

7.2. Tabu search (TS)

The algorithm maintains a taboo list containing the results of the last few steps. In the process, we can only take the neighbor of the current solution that is not in the taboo list. The taboo list must be changed at each iteration. If you add a new item, you must delete the first item from the beginning if the list is already full. (Glover & Laguna, 1998) The pseudo code of the Tabu Search is illustrated in Figure 9.

7.3. Particle Swarm Optimization

It maintains a population of possible solutions. The particles move through the search space using a simple mathematical formula. Particle movement is determined by the best search space positions found (the best position of the particle and the best position on the particle - best of all).

```
BEGIN PROCEDURE
    Step 1. Starting with a possible solution that will initially
    be the best solution, indicated with $S_{best}$
    WHILE (termination condition is not met) DO
        Step 2. Making the neighbor of $S_{best}$ Choose the best
        of these, which is not yet in the taboo list, indicated
        with $S_{neighbor}$.
        Step 3. $S_{neighbor}$ is inserted as the last item in the
        taboo list. (When the taboo list is full, we delete the
        first item).
        IF ($S_{neighbor}$ is better than $S_{best}$) THEN DO
            Step 4. $S_{best} = S_{neighbor}$
        END IF
    END WHILE
END PROCEDURE
```

Figure 9. The pseudo code of the Tabu Search (Glover & Laguna, 1998).

The algorithms use the following formulas:

1. Particle velocity updating formula:

$$\boldsymbol{v}_{id}(t+1) = \boldsymbol{v}_{id}(t) \oplus \alpha(\boldsymbol{p}_{id} - \boldsymbol{x}_{id}(t)) \oplus \beta(\boldsymbol{g}_d - \boldsymbol{x}_{id}(t)) \quad (8)$$

Current particle velocity updating:

$$\boldsymbol{x}_{id}(t+1) = \boldsymbol{x}_{id}(t) + \boldsymbol{v}_{id}(t+1) \quad (9)$$

where the following notations are used:

- $\boldsymbol{v}_{id}(t)$: the new velocity (swap sequence).
- $\boldsymbol{p}_{id} - \boldsymbol{x}_{id}(t)$: the difference between the best and current particle position (Basic Swap Sequence - BSS (Wang et al., 2003).
- $\boldsymbol{g}_d - \boldsymbol{x}_{id}(t)$: the difference between the globally best particle position and the current position of a given particle (Basic Swap Sequence - BSS (Wang et al., 2003).
- $\alpha, \beta \in [0,1]$ are random numbers. The $\oplus$ operation (Wang et al., 2003) is the sequential execution of the swap sequences.

Figure 10 illustrates the pseudo code of the Particle Swarm Optimization.

```
BEGIN PROCEDURE
    Step 1. Initializing the positions of the particles, i.e.,
    $\boldsymbol{x}_i(0)$.
    Step 2. Initializing the best positions of the particles, i.e.,
    $\boldsymbol{p}_{id}(0)$, initially with the starting positions of the
    particles, so $\boldsymbol{p}_{id}(0) := \boldsymbol{x}_{id}(0)$.
    Step 3. Generating velocity ($\boldsymbol{v}_{id}(t)$) for each particle
    $\boldsymbol{v}_{id}(t)$. The velocity represents the exchange sequence
    shown in (Wang et al., 2003).
    Step 4. Fitness value calculation and initialization of the
    best particle, $\boldsymbol{g}$.
    WHILE (termination condition is not met) DO
        WHILE (not all particles were selected) DO
            Step 5. Update the particle velocity using eq. (8)
            Step 6. From the $\boldsymbol{v}_{id}(t+1)$ velocity the creation
            of the Basic Swap Sequence (BSS).
            Step 7. Updating the current particle velocity
            with eq. (9).
            Step 8. Updating the $\boldsymbol{p}_{id}$ value.
            Step 9. Updating the global best position, i.e. $\boldsymbol{g}_d$.
        END WHILE
    END WHILE
END PROCEDURE
```

Figure 10. The pseudo code of the Particle Swarm Optimization (Wang et al., 2003).

8. Representation of clustering task, its evaluation, objective function

When applying construction and improvement algorithms for solving the clustering problem, we need to use a representation mode. In this paper permutation representation (mapping vector) is applied (Figure 11). The elements of the mapping vector are the individual objects. The objective function (of the improvement heuristics i.e., PSO, GA, TS) is to maximize the Silhouette value. The evaluation of the mapping vector is illustrated in Figure 12.

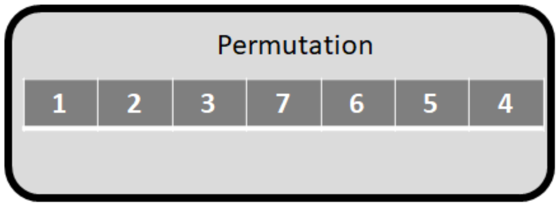

Figure 11. The permutation (mapping vector) representation.

```
BEGIN PROCEDURE
    Step 1. Calculating the average distance between the
    elements.
    Step 2. Start with a cluster into which putting some
    elements of the permutation.
    WHILE (not all objects have been selected) DO
        Step 3. Taking the next element of the permutation.
        IF (The distance between the next and previous
        elements of the permutation is greater than the
        average distance) DO
            Step 4. Starting a new cluster and inserting the
            next element of the permutation here.
        ELSE
            Step 5. Inserting the next element of the
            permutation into the current cluster.
        END IF
    END WHILE
END PROCEDURE
```

Figure 12. The evaluation of the mapping vector.

9. Test results

In this section, the test results are presented. First, test results for our own data then results for benchmark datasets are presented. The abbreviations and their meaning is presented in Table 2. Figure 13-17 illustrates the test result of the algorithms.

Table 2. The abbreviations and their meaning.

Abbreviation	Meaning
KM	K-Means
SL	Hierarchical Clustering: Single Linkage
CL	Hierarchical Clustering: Complete Linkage
AM	Hierarchical Clustering: Average Method
CM	Hierarchical Clustering: Centroid Method
WM	Hierarchical Clustering: Ward Method
AI	Arbitrary Insertion
CI	Cheapest Insertion
FI	Farthest Insertion
G	Greedy
NI	Nearest Insertion
NN	Nearest Neighbour
PSO+R	Particle Swarm Optimization with randomly generated initial solutions
PSO+C,R	Particle Swarm Optimization with randomly and construction algorithms (AI, CI, FI, G, NN, NI) generated initial solutions
GA+R	Genetic Algorithm with randomly generated initial solutions
GA+C,R	Genetic Algorithm with randomly and construction algorithms (AI, CI, FI, G, NN, NI) generated initial solutions
TS+R	Tabu Search with randomly generated initial solution
TS+AI	Tabu Search with Arbitrary Insertion (AI) generated initial solution
TS+CI	Tabu Search with Cheapest Insertion (CI) generated initial solution
TS+FI	Tabu Search with Farthest Insertion (FI) generated initial solution
TS+G	Tabu Search with Greedy (G) generated initial solution
TS+NI	Tabu Search with Nearest Insertion (NI) generated initial solution
TS+NN	Tabu Search with Nearest Neighbour (NN) generated initial solution

In Table 3-7 the N means the number of objects and k means the optimal number of clusters. In these tables the average values of 10 test runs are detailed. Table 8 present the summary of the test results.

In the following test results of the benchmark datasets from (Fränti & Sieranoja, 2014) is presented.

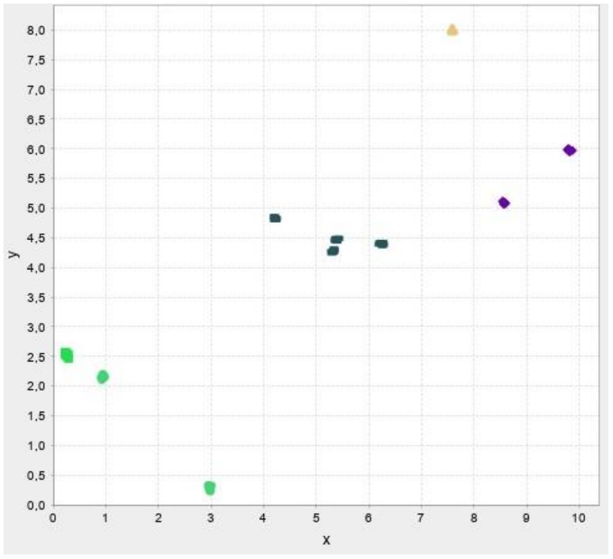

Figure 13. The result of average linkage for our data.

Table 3. Test results for our data.

Own data (N=100, k=10)			
Method	Number of clusters	Silhoutte value	Running time (min)
KM	6.0	0.8764	1.8126 E-5
SL	5.0	0.8470	2.7036 E-5
CL	5.0	0.8470	2.7898 E-5
AM	5.0	0.8470	2.7661 E-5
CM	5.0	0.8470	3.8139 E-5
WM	5.0	0.8470	3.8730 E-5
AI	5.5	0.8901	3.5881 E-6
CI	5.3	0.8863	3.4056 E-5
FI	5.8	0.8344	3.1482 E-4
G	5.0	0.9200	4.7099 E-5
NI	5.6	0.8152	2.9473 E-4
NN	5.0	0.9133	8.1065 E-6
PSO+R	1.0	-	0.0956
PSO+C,R	5.9	0.93055	0.09122
GA+R	1.0	-	0.0111
GA+C,R	1.0	-	0.0158
TS+R	1.0	-	0.1364
TS+AI	1.0	-	0.1342
TS+CI	1.0	-	0.1313
TS+FI	1.0	-	0.1352
TS+G	1.0	-	0.1233
TS+NI	5.3	0.7619	0.1352
TS+NN	1.0	-	0.1324

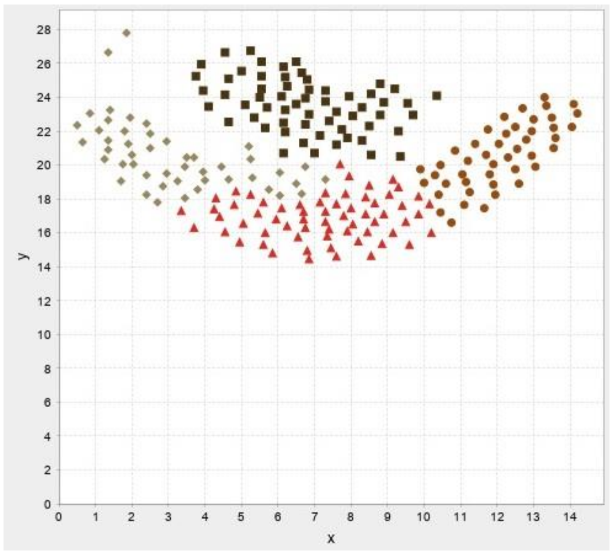

Figure 14. The result of complete linkage for flame data.

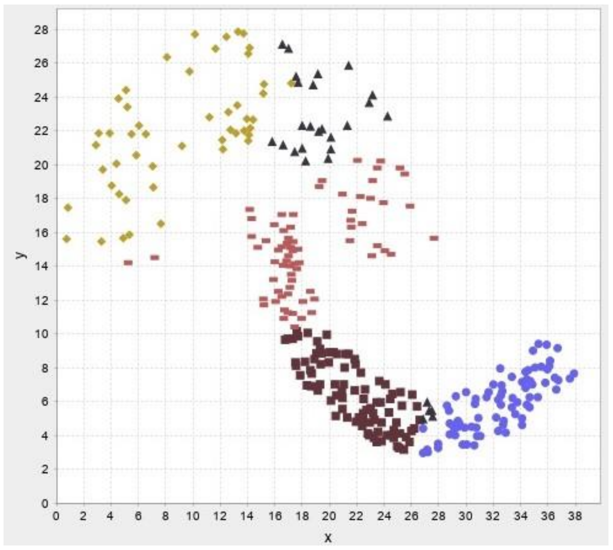

Figure 15. The result of K-Means for Jain data.

Table 4. Test results for Flame data.

Flame (N=240, k=2)			
Method	Number of clusters	Silhoutte value	Running time (min)
KM	4.0	0.6275	2.4643 E-4
SL	3.0	0.3866	0.0010
CL	4.0	0.5545	8.8634 E-4
AM	5.0	0.5125	0.0010
CM	4.0	0.4754	0.0010
WM	4.0	0.5201	0.0010
AI	5.0	0.5662	1.0601 E-4
CI	5.2	0.4964	6.3844 E-4
FI	5.0	0.4397	0.0093
G	5.0	0.5192	0.0013
NI	5.0	0.4793	0.0094
NN	5.1	0.5155	2.1485 E-4
PSO+R	3.0	0.4377	13.1455
PSO+C,R	3.0	0.5388	13.7894
GA+R	3.0	0.5287	14.7877
GA+C,R	3.0	0.4012	12.1245
TS+R	3.2	0.5723	13.4832
TS+AI	3.1	0.5955	12.9809
TS+CI	3.0	0.4020	13.7845
TS+FI	3.0	0.5807	14.2906
TS+G	3.0	0.5292	12.8740
TS+NI	3.0	0.5237	13.1240
TS+NN	3.0	0.5070	14.1859

Table 5. Test results for Jain data.

Jain (N=373, k=2)			
Method	Number of clusters	Silhoutte value	Running time (min)
KM	5.0	0.6530	9.8182 E-5
SL	6.0	0.6892	0.0013
CL	6.0	0.7244	0.0011
AM	6.0	0.7314	0.0011
CM	6.0	0.7313	0.0018
WM	6.0	0.7235	0.0019
AI	7.0	0.7279	2.8611 E-5
CI	7.2	0.6703	0.0015
FI	7.0	0.6295	0.0550
G	6.0	0.7294	0.0018
NI	6.6	0.6726	0.0503
NN	7.0	0.6846	2.4654 E-4
PSO+R	7.0	0.6026	21.0572
PSO+C,R	7.1	0.7367	22.7064
GA+R	7.2	0.6980	20.4734
GA+C,R	7.0	0.6777	21.9766
TS+R	7.0	0.6347	21.2806
TS+AI	7.0	0.6724	22.5610
TS+CI	7.0	0.7990	21.7342
TS+FI	7.0	0.7335	20.5112
TS+G	7.0	0.6438	20.4698
TS+NI	7.0	0.7059	21.2041
TS+NN	7.0	0.7266	21.6740

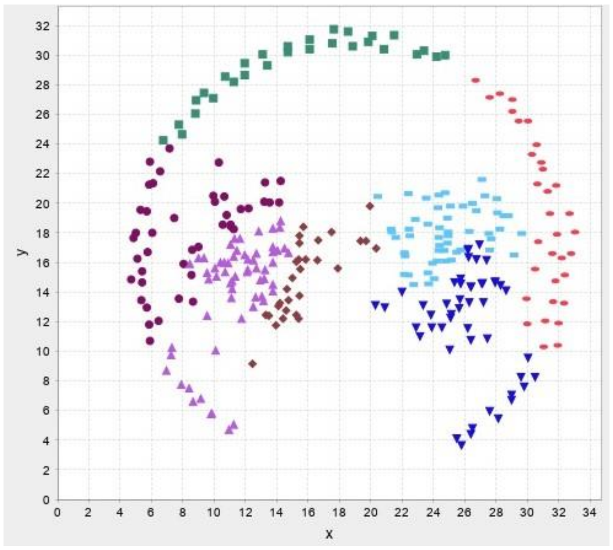

Figure 16. The result of Arbitrary Insertion for Pathbased data.

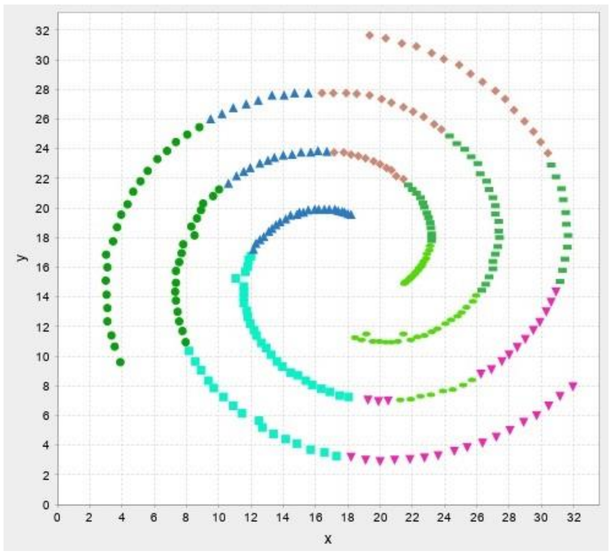

Figure 17. The result of Arbitrary Insertion for Spiral data.

Table 6. Test results for Pathbased data.

Pathbased (N=300, k=3)			
Method	Number of clusters	Silhoutte value	Running time (min)
KM	5.0	0.5676	4.5017 E-5
SL	5.0	0.4952	6.1105 E-4
CL	5.0	0.6247	6.0886 E-4
AM	6.0	0.5743	5.976 E-4
CM	6.0	0.6191	9.4529 E-4
WM	6.0	0.6360	0.0010
AI	6.4	0.6135	1.4689 E-5
CI	7.1	0.5564	7.6838 E-4
FI	6.1	0.5478	0.0231
G	8.0	0.6627	9.6926 E-4
NI	6.8	0.5479	0.0216
NN	6.8	0.5840	1.3669 E-4
PSO+R	6.5	0.6124	19.7260
PSO+C,R	6.8	0.4605	18.0736
GA+R	5.9	0.5030	19.0785
GA+C,R	5.7	0.5846	20.7408
TS+R	6.2	0.4322	19.4064
TS+AI	6.3	0.5785	19.3890
TS+CI	5.9	0.4850	19.3867
TS+FI	5.7	0.4553	19.8644
TS+G	6.4	0.5654	20.5816
TS+NI	6.5	0.5559	20.3076
TS+NN	6.0	0.5717	20.6610

Table 7. Test results for Spiral data.

Spiral (N=312, k=3)			
Method	Number of clusters	Silhoutte value	Running time (min)
KM	6.0	0.5431	4.9849 E-5
SL	8.0	0.4710	7.4046 E-4
CL	7.0	0.5133	7.4932 E-4
AM	6.0	0.5448	0.0010
CM	6.0	0.5330	0.0010
WM	6.0	0.5448	0.0010
AI	7.4	0.6022	1.6550 E-5
CI	9.3	0.5499	8.5700 E-4
FI	8.1	0.5270	0.0271
G	10.0	0.5498	0.0010
NI	9.1	0.4985	0.0247
NN	9.6	0.5738	1.4543 E-4
PSO+R	8.2	0.5468	20.6559
PSO+C,R	8.5	0.4828	21.4954
GA+R	9.1	0.6571	20.0329
GA+C,R	8.4	0.8893	20.2922
TS+R	8.0	0.8092	21.0241
TS+AI	8.4	0.6826	20.5617
TS+CI	8.4	0.5627	20.8311
TS+FI	8.5	0.6349	20.1871
TS+G	7.9	0.6809	21.9084
TS+NI	8.2	0.6366	20.0576
TS+NN	8.3	0.5030	20.6369

Table 8. Summary.

Summary		
Method	Silhoutte value	Running time
KM	+	+
SL	+	+
CL	+	+
AM	+	+
CM	+	+
WM	+	+
AI	+	+
CI	+	+
FI	+	+
G	+	+
NI	+	+
NN	+	+
PSO+R	+	-
PSO+C,R	+	-
GA+R	+	-
GA+C,R	+	-
TS+R	+	-
TS+AI	+	-
TS+CI	+	-
TS+FI	+	-
TS+G	+	-
TS+NI	+	-
TS+NN	+	-

The test results show that despite the failure to reach the desired number of clusters, the silhouette values are high, so the implemented algorithms cluster relatively well. We do not recommend using improvement algorithms due to their high running time, we only recommend modified versions of construction algorithms and traditional clustering procedures.

10. Conclusion

In this article different tour-based clustering algorithms are compared with the classical methods and analyzed. After the literature review the traditional clustering algorithms (K-Means, Hierarchical Methods) are presented, then the Silhouette index to measure the quality of the clustering result. After that construction algorithms and improvement algorithms are detailed. Then our cluster representation technique and evaluation is described. After that test results are presented. In the test we have implemented and analyzed the main clustering methods and the tour construction and tour improvement methods. The comparison test performed on self-generated dataset and several clustering benchmark test: Flame, Jain, Pathbased and Spiral. Based on the test results the traditional clustering algorithms and the construction algorithms have efficiency in partitioning datasets with our representation and evaluation technique.

Conflict of interest

The authors have no conflict of interest to declare.

Acknowledgments

Supported by the ÚNKP-19-3 New National Excellence Program of the Ministry for Innovation and Technology.

Financing

Supported by the ÚNKP-19-3 New National Excellence Program of the Ministry for Innovation and Technology.

References

Chang, C. H. (1996). Simulated annealing clustering of Chinese words for contextual text recognition. *Pattern Recognition Letters*, *17*(1), 57-66.
https://doi.org/10.1016/0167-8655(95)00080-1

Chen, C. Y., & Ye, F. (2012). Particle swarm optimization algorithm and its application to clustering analysis. In 2012 Proceedings of 17th Conference on Electrical Power Distribution, 789-794.

Chiu, C. Y., Chen, Y. F., Kuo, I. T., & Ku, H. C. (2009). An intelligent market segmentation system using k-means and particle swarm optimization. *Expert Systems with Applications*, *36*(3), 4558-4565.
https://doi.org/10.1016/j.eswa.2008.05.029

Cui, X., Potok, T. E., & Palathingal, P. (2005). Document clustering using particle swarm optimization. *In Proceedings 2005 IEEE Swarm Intelligence Symposium*, 2005. SIS 2005, 185-191.
https://doi.org/10.1109/SIS.2005.1501621

Doval, D., Mancoridis, S., & Mitchell, B. S. (1999). Automatic clustering of software systems using a genetic algorithm. In *STEP'99. Proceedings Ninth International Workshop Software Technology and Engineering Practice*, 73-81.
https://doi.org/10.1109/STEP.1999.798481

El Rhazi, A., & Pierre, S. (2008). A tabu search algorithm for cluster building in wireless sensor networks. *IEEE Transactions on Mobile Computing*, *8*(4), 433-444.
https://doi.org/10.1109/TMC.2008.125

Forsati, R., Mahdavi, M., Kangavari, M., & Safarkhani, B. (2008). Web page clustering using harmony search optimization. In *2008 Canadian Conference on Electrical and Computer Engineering*, 001601-001604.
https://doi.org/10.1109/CCECE.2008.4564812

França, P. M., Sosa, N. M., & Pureza, V. (1999). An adaptive tabu search algorithm for the capacitated clustering problem. *International Transactions in Operational Research*, *6*(6), 665-678.
https://doi.org/10.1111/j.1475-3995.1999.tb00180.x

Fränti, P., & Sieranoja, S. (2014). K-means properties on six clustering benchmark datasets. *Applied Intelligence, 48*(12), 4743-4759.
https://doi.org/10.1007/s10489-018-1238-7

Gao, S., Wang, Y., Cheng, J., Inazumi, Y., & Tang, Z. (2016). Ant colony optimization with clustering for solving the dynamic location routing problem. *Applied Mathematics and Computation*, 285, 149-173.
https://doi.org/10.1016/j.amc.2016.03.035

Glover, F., & Laguna, M. (1998). Tabu search. In Handbook of combinatorial optimization, 2093-2229.
https://doi.org/10.1007/978-1-4613-0303-9_33

Golden, B., Bodin, L., Doyle, T., & Stewart Jr, W. (1980). Approximate traveling salesman algorithms. *Operations research*, *28*(3-part-ii), 694-711.
https://doi.org/10.1287/opre.28.3.694

Hoang, D. C., Yadav, P., Kumar, R., & Panda, S. K. (2010). A robust harmony search algorithm-based clustering protocol for wireless sensor networks. In *2010 IEEE International Conference on Communications Workshops*, 1-5.
https://doi.org/10.1109/ICCW.2010.5503895

Hoang, D. C., Yadav, P., Kumar, R., & Panda, S. K. (2013). Real-time implementation of a harmony search algorithm-based clustering protocol for energy-efficient wireless sensor networks. *IEEE transactions on industrial informatics*, *10*(1), 774-783.
https://doi.org/10.1109/TII.2013.2273739

Hruschka, E. R., & Ebecken, N. F. (2003). A genetic algorithm for cluster analysis. Intelligent Data Analysis, 7(1), 15-25.

Kalyani, S., & Swarup, K. S. (2011). Particle swarm optimization based K-means clustering approach for security assessment in power systems. *Expert systems with applications*, *38*(9), 10839-10846.
https://doi.org/10.1016/j.eswa.2011.02.086

Kinney, G. W., Barnes, J. W., & Colletti, B. W. (2007). A reactive tabu search algorithm with variable clustering for the unicost set covering problem. International Journal of Operational Research, 2(2), 156-172.

Kudova, P. (2007). Clustering genetic algorithm. *In 18th International Workshop on Database and Expert Systems Applications (DEXA 2007)*, 138-142.
https://doi.org/10.1109/DEXA.2007.65

Lazzerini, B., & Marcelloni, F. (2000). A genetic algorithm for generating optimal assembly plans. *Artificial Intelligence in Engineering, 14*(4), 319–329.
https://doi.org/10.1016/S0954-1810(00)00011-X

Li, C., & Yang, S. (2009). A clustering particle swarm optimizer for dynamic optimization. *In 2009 IEEE Congress on Evolutionary Computation*, 439-446.
https://doi.org/10.1109/CEC.2009.4982979

Lu, Y., Lu, S., Fotouhi, F., Deng, Y., & Brown, S. J. (2004). FGKA: A fast genetic k-means clustering algorithm. *In Proceedings of the 2004 ACM symposium on Applied computing*, 622-623.
https://doi.org/10.1145/967900.968029

Madhuri, R., Murty, M. R., Murthy, J. V. R., Reddy, P. P., & Satapathy, S. C. (2014). Cluster analysis on different data sets using K-modes and K-prototype algorithms. *In ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India-Vol II* (pp. 137-144). Springer, Cham.
https://doi.org/10.1007/978-3-319-03095-1_15

Mahdavi, M., & Abolhassani, H. (2009). Harmony K-means algorithm for document clustering. *Data Mining and Knowledge Discovery*, *18*(3), 370-391.
https://doi.org/10.1007/s10618-008-0123-0

Maulik, U., & Bandyopadhyay, S. (2000). Genetic algorithm-based clustering technique. *Pattern recognition*, *33*(9), 1455-1465.
https://doi.org/10.1016/S0031-3203(99)00137-5

Milano, M., & Koumoutsakos, P. (2002). A clustering genetic algorithm for cylinder drag optimization. *Journal of Computational Physics*, *175*(1), 79-107.
https://doi.org/10.1006/jcph.2001.6882

Murtagh, F. (1983). A survey of recent advances in hierarchical clustering algorithms. *The Computer Journal, 26*(4), 354-359.
https://doi.org/10.1093/comjnl/26.4.354

Murty, M. R., Murthy, J. V. R., Reddy, P. P., Naik, A., & Satapathy, S. C. (2014). Homogeneity separateness: a new validity measure for clustering problems. In *ICT and Critical Infrastructure: Proceedings of the 48th Annual Convention of Computer Society of India-Vol I* (pp. 1-10). Springer, Cham. https://doi.org/10.1007/978-3-319-03107-1_1

Murty, M. R., Naik, A., Murthy, J. V. R., Reddy, P. P., Satapathy, S. C., & Parvathi, K. (2014). Automatic clustering using teaching learning based optimization. *Applied Mathematics*, 5(8), 1202-1211. https://doi.org/10.4236/am.2014.58111

Nilsson, C. (2003). Heuristics for the traveling salesman problem. Linkoping University, 38, 00085-9.

Olson, C. F. (1995). Parallel algorithms for hierarchical clustering. *Parallel computing, 21*(8), 1313-1325. https://doi.org/10.1016/0167-8191(95)00017-I

Omran, M. G., Engelbrecht, A. P., & Salman, A. (2004). Image classification using particle swarm optimization. *In Recent advances in simulated evolution and learning*, 347-365. https://doi.org/10.1142/9789812561794_0019

Omran, M. G., Salman, A., & Engelbrecht, A. P. (2006). Dynamic clustering using particle swarm optimization with application in image segmentation. *Pattern Analysis and Applications*, *8*(4), 332. https://doi.org/10.1007/s10044-005-0015-5

Osman, I. H., & Christofides, N. (1994). Capacitated clustering problems by hybrid simulated annealing and tabu search. *International Transactions in Operational Research, 1*(3), 317-336. https://doi.org/10.1016/0969-6016(94)90032-9

Paoli, A., Melgani, F., & Pasolli, E. (2009). Clustering of hyperspectral images based on multiobjective particle swarm optimization. *IEEE transactions on geoscience and remote sensing*, *47*(12), 4175-4188. https://doi.org/10.1109/TGRS.2009.2023666

Rosenkrantz, D. J., Stearns, R. E., & Lewis, P. M. (1974). Approximate algorithms for the traveling salesperson problem. In *15th Annual Symposium on Switching and Automata Theory (SWAT 1974)*, 33-42. https://doi.org/10.1109/SWAT.1974.4

Runkler, T. A. (2005). Ant colony optimization of clustering models. *International Journal of Intelligent Systems, 20*(12), 1233-1251. https://doi.org/10.1002/int.20111

Scheunders, P. (1997). A genetic c-means clustering algorithm applied to color image quantization. *Pattern recognition*, *30*(6), 859-866. https://doi.org/10.1016/S0031-3203(96)00131-8

Shelokar, P. S., Jayaraman, V. K., & Kulkarni, B. D. (2004). An ant colony approach for clustering. *Analytica Chimica Acta*, *509*(2), 187-195. https://doi.org/10.1016/j.aca.2003.12.032

Starkweather, T., McDaniel, S., Mathias, K. E., Whitley, L. D., & Whitley, C. (1991). A Comparison of Genetic Sequencing Operators. ICGA, 69-76.

Van der Merwe, D. W., & Engelbrecht, A. P. (2003). Data clustering using particle swarm optimization. *In The 2003 Congress on Evolutionary Computation, 2003. CEC'03*, 215-220. https://doi.org/10.1109/CEC.2003.1299577

Wagstaff, K., Cardie, C., Rogers, S., & Schrödl, S. (2001). Constrained k-means clustering with background knowledge. In Proceedings of the Eighteenth International Conference on Machine Learning (Vol. 1, pp. 577-584).

Wang, K. P., Huang, L., Zhou, C. G., & Pang, W. (2003). Particle swarm optimization for traveling salesman problem. In Proceedings of the 2003 International Conference on Machine Learning and Cybernetics, 1583-1585. https://doi.org/10.1109/ICMLC.2003.1259748

Wang, X., & Xu, Y. (2019). An improved index for clustering validation based on Silhouette index and Calinski-Harabasz index. In IOP Conference Series: Materials Science and Engineering (Vol. 569, No. 5, p. 052024). IOP Publishing

Whitley, D. (1994). A genetic algorithm tutorial. *Statistics and computing*, *4*(2), 65-85. https://doi.org/10.1007/BF00175354

Wu, X., Chu, C. H., Wang, Y., & Yan, W. (2007). A genetic algorithm for cellular manufacturing design and layout. *European journal of operational research, 181*(1), 156-167. https://doi.org/10.1016/j.ejor.2006.05.035

Xu, R., & Wunsch, D. (2005). Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3), 645-678. https://doi.org/10.1109/TNN.2005.845141

Yang, J., Xu, M., Zhao, W., & Xu, B. (2010). A multipath routing protocol based on clustering and ant colony optimization for wireless sensor networks. *Sensors*, *10*(5), 4521-4540. https://doi.org/10.3390/s100504521

Zhao, B., Zhu, Z., Mao, E., & Song, Z. (2007). Image segmentation based on ant colony optimization and K-means clustering. In *2007 IEEE International Conference on Automation and Logistics*, 459-463. https://doi.org/10.1109/ICAL.2007.4338607