**EURASIAN JOURNAL OF MATHEMATICAL AND COMPUTER APPLICATIONS**
ISSN 2306–6172
Volume 13, Issue 2 (2025) 50 – 61

# ENHANCING SCHOOL BUS ROUTING EFFICIENCY: A STUDY OF CLUSTERING METHODS

**Nuriyeva F.** [1], **Erdemci V.**

**Abstract** School Bus Routing Problem aims to create an efficient routes and allocate serviceable areas for student school buses. Buses pick up students from various locations based on their capacities and closeness or distances. It is a typical clustering problem for different locations. However, the choice of algorithm may vary depending on student locations and area shapes. This study focuses on to compare and evaluate the performance of four well-known clustering methods like K-Means, DBSCAN, Hierarchical Clustering and Gaussian Mixture Model on 500 randomly generated geographical points within the boundaries of Izmir (Türkiye). The evaluation is conducted based on their density and distribution characteristics. Silhouette Score, Davies-Bouldin Score metrics, and running time are used to assess clustering quality. This analysis highlights the advantages and disadvantages of the algorithms to provide insights into their applicability in different scenarios. Additionally, a visual representation of clustering outcomes offers a deeper understanding of the spatial distribution of the data.

**Key words:** School Bus Routing Problem, Clustering, K-Means Clustering, DBSCAN, Hierarchical Clustering, Gaussian Mixture Model.

**AMS Mathematics Subject Classification:** 90B06, 90C59, 68T10.

**DOI:** 10.32523/2306-6172-2025-13-2-50-61

## 1 Introduction

People travel by private car or public transportation for both official and private purposes. The growing population and the number of vehicles cause problems regarding the duration and price of these trips. The increase in the time spent by vehicles in traffic not only causes an increase in the price of transportation but also causes more toxic gas emissions; thus environmental problems, more tension and stress; and thus personal health problems. If we consider the existence of other social and sociological problems that can be added to these problems, it can be understood that transportation problems are among the most important problems of our day.

In this respect, scientists approach transportation problems from different perspectives and seek various solutions. While some scientists are trying to develop vehicle systems that minimize toxic gas emissions in terms of directly affecting environmental problems, some scientists are working on models that will reduce the travel time of vehicles [1, 2, 3, 4].

In this study, we consider the classical School Bus Routing Problem (SBRP). The SBRP is a real-life problem. The most important point to consider in real-life problems is the necessity of taking into account all the situations affecting the problem. In our country - Türkiye, 40 out of every 100 students use school busses, highlighting the significant economic potential of this sector. The SBRP is a significant and practical transportation challenge that affects

[1]Corresponding Author.

countless families globally on a daily basis. School administrators and service providers must design and implement a transportation system that ensures students are transported to and from schools safely, reliably, and cost-effectively.

Classical SBRP was first published by Newton and Thomas in 1969 [5]. Since then, many researchers have addressed the problem from different perspectives. There are also review articles that include these studies [6, 7]. Some researchers have also examined the Vehicle Routing Problem (VRP) in their review articles, and have considered SBRP as a sub-problem and made various evaluations from this perspective [1]. SBRP aims to organize school busses in the most efficient way. This problem basically seeks an answer to the question, "Which student is assigned to which school bus and which routes do these service vehicles follow in order to establish the most efficient system?". The SBRP has been studied in the literature in different ways in terms of solution approaches. The SBRP problem actually includes two basic optimization problems: Clustering and Routing. There are solution methods in the literature that handle these two sub-problems independently or in a hybrid way.

In this study, clustering algorithms like K-Means, DBSCAN, Hierarchical and Gaussian Mixture Model (GMM) were examined according to the density and distribution of 500 randomly selected geographical points in Izmir region (Türkiye) and the advantages and disadvantages of the algorithms were revealed.

The subsequent sections are organized as follows. Sec. 2 presents the literature review about SBRP, the algorithms which are used in this study is described in Sec. 3. The outcomes derived from the computational experiments are reported in Sec 4. Discussion about the advantages and disadvantages of the algorithms are outlined in Sec. 5. Finally, the conclusions are presented in Sec. 6.

## 2 Literature Review

One of the earliest studies is published by Angel et al in 1972 . The study consist of the algorithm about SBRP and it tries to reduce total distance which is taken by buses. The study is investigated on 1500 students located on Indiana, USA [8].

The study which is published in 1997 by Braca et al showed that computerized approach for the student transportation in the area of New York, USA. Also, student capacities, stations and geographic informations are placed in the study [6].

One of the studies from the 2000s was published by Li and Fu in 2002. They showed a study consisting of the transportation of 86 students who were located at 54 pick-up points points in Hong Kong, China. Study, includes reducing the number of school buses and optimizing travel time by short routes [9].

A precise branch and price framework for SBRP has been presented by Kinable et al. (2014) with a strong emphasis on the efficiency problems naturally associated with column generation (CG). The experiments were conducted on a set of 128 SBRP samples [10].

An effective routing algorithm for SBRP has been proposed by Kumar et al. (2015). The branch and boundary algorithm provided the best solution for smaller problems, but for a group of schools, it provided the optimal solution to help these schools optimize bus routes, the number of buses used, and therefore the cost [11].

Bus stop selection for SBRP has been taken into consideration by Sarubbi et al. (2016) in order to minimize the number of bus stops while ensuring that all students are assigned to a bus stop within the home-bus stop walking distance restriction [12].

A school bus routing algorithm that is taking into consideration of safety of the students and total amount of time the students stay on the bus is proposed. The proposed algorithm provided a good solution. In addition, in this study a web based software system were

developed to allow the real world application of the algorithm [13].

The SBRP for a single-school configuration, reflecting China's school bus systems with assumptions like a homogeneous bus fleet, home pick-ups, and fixed school times is considered. It addresses the stochastic and time-varying nature of travel times, where path costs fluctuate due to uncertainty. The optimal path selection in stochastic time-dependent (STD) networks is treated as a sub-problem of the SBRP [14].

It is aimed to reduce the problems such as bus purchase cost, employment of drivers and assistants, repair and fuel costs, which increase the global cost of SBRP, by reducing the number of buses [15].

The SBRP with bus stop selection, tackling bus stop determination, student allocation, and route computation to minimize routing costs while limiting student walking distances is allocated. In this study, a fast and efficient mataheuristic that partially allocates students to reachable stops and optimizes routing costs is developed [16] .

Some basic practical requirements such as multiple schools, mixed loads, heterogeneous fleets, various pickup time windows and school bell constraints were taken into consideration for SBRP. In addition, a time-discretized multi-commodity network flow model using a student-loading state-oriented spacetime network has been proposed [17].

The SBRP in Holingol considers both school accessibility and scheme equity. Accessibility is measured by the average student commuting time, while equity is assessed through the average detour time of students [18].

The cumulative SBRP focuses on transporting students home from school by the same buses. The goal is to select drop-off points within a certain walking distance and create routes that minimize the total arrival time for all students. Six mixed integer linear programming formulations based on the original and auxiliary graphs were proposed and compared numerically using real examples. Computational experiments were conducted to evaluate the performance of these models [19].

A Multi-Period School Bus Routing Problem was proposed, which aims to minimize the total fleet distance by taking into account vehicle capacity and walking restrictions. A Mixed Integer Linear Programming model and a metaheuristic algorithm combining Iterative Local Search and Variable Neighborhood Descent have been introduced. In addition, new strategies for student allocation have been presented. The study expanded the existing samples with period-related requests, resulting in 448 new samples for evaluation. The algorithm efficiently solves large samples with low computational effort [20].

There are different variations in the school routing problem. In this study, we consider a single school bus that takes all students to a single school without a time window. The main contribution of this study is to determine the effectiveness of each algorithm on the data, to give the advantages and disadvantages of the algorithms, and to discuss which algorithm is more suitable for practical use.

## 3 Methodology

Clustering is the process of grouping physical or abstract unlabeled objects into classes consisting of similar objects. A cluster is a collection of data objects within the same cluster that are similar to each other and different from objects in other clusters. Clustering has many applications in summarization, learning, segmentation and target marketing. Clustering can be thought of as a concise data model, which can be interpreted in the sense of a summary or a generative model. The basic problem of clustering can be expressed as follows: Given a set of data points, divide them into a set of groups that are as similar as possible. This definition is a very rough definition, and differences in the problem definition can be significant depending

on the specific model used [21, 22].

Clustering problems can be addressed using a wide variety of methods. The most common methods are: Feature Selection Methods, Probabilistic and Generative Models, Distance-Based Algorithms, Density and Grid-Based Methods, Leveraging Dimensionality Reduction Methods, The High Dimensional Scenario and Scalable Techniques for Cluster Analysis.

The data type plays an important role in the selection of the clustering method. there are different types of data available, such as Categorical Data, Text Data, Multimedia Data, Time Series Data, Discrete Sequences, Network Data, Uncertain Data, etc.

Clustering analysis is the process of classifying objects into subsets in such a way that they make sense in the context of a particular problem.

In the context of urban planning, clustering can be applied to identify high-density areas, optimize resource allocation, and analyze patterns in spatial datasets. The Izmir region of Türkiye, with its diverse geographical features and urban structure, provides an ideal setting for this analysis. This study evaluates the clustering performance of K-Means, DBSCAN, Hierarchical Clustering, and GMM algorithms on 500 random points within Izmir's boundaries. By using real-world spatial metrics and visualizing the clustering outcomes, the study aims to uncover algorithmic strengths and limitations in handling geographical data.

### 3.1 K-Means Clustering Algorithm

The K-Means is one of the simple and well-known method for data clustering. It is widely used in practical applications due to its simplicity.

The algorithm starts by selecting K points as the initial centroids. Then, each selected point is assigned to the closest centroid according to a certain measure of proximity. After the clusters are created, the centroids of each cluster are updated. The algorithm then iteratively repeats these two steps until the center points do not change or another alternative relaxed convergence criterion is met [23, 24, 25, 26].

Consider a set of observations $(x_1, x_2, \ldots, x_n)$, where each observation is a $d$-dimensional real vector. The objective of k-means clustering is to divide these $n$ observations into $k \leq n$ clusters $S = \{S_1, S_2, \ldots, S_k\}$, in such a way that the within-cluster sum of squares (WCSS), or variance, is minimized. The formal goal can be expressed as:

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 = \arg\min_{\mathbf{S}} \sum_{i=1}^{k} |S_i| \operatorname{Var}(S_i)$$

Here, $\boldsymbol{\mu}_i$ denotes the mean (or centroid) of the points in $S_i$, defined by:

$$\boldsymbol{\mu}_i = \frac{1}{|S_i|} \sum_{\mathbf{x} \in S_i} \mathbf{x},$$

where $|S_i|$ is the number of elements in $S_i$, and $\|\cdot\|$ represents the standard $L^2$ norm. This formulation is equivalent to minimizing the squared pairwise deviations of points within the same cluster:

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \frac{1}{|S_i|} \sum_{\mathbf{x},\mathbf{y} \in S_i} \|\mathbf{x} - \mathbf{y}\|^2 .$$

This equivalence can be derived from the following identity:

$$|S_i| \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 = \frac{1}{2} \sum_{\mathbf{x},\mathbf{y} \in S_i} \|\mathbf{x} - \mathbf{y}\|^2 .$$

Since the total variance remains constant, the problem can be reinterpreted as maximizing the between-cluster sum of squares (BCSS), which measures the squared differences between points from different clusters. This relationship is closely tied to the law of total variance in probability theory.

The steps of the K-Means algorithm are as follows:

1. **Initialization:**

   Select $k$ initial centroids randomly from the dataset $X$. Let the centroids be $\mu_1, \mu_2, \ldots, \mu_k$.

2. **Assignment Step:**

   For each data point $x_j \in X$, assign it to the cluster whose centroid is the nearest:

$$S_i^{(t)} = \left\{ x_p : \left\| x_p - m_i^{(t)} \right\|^2 \leq \left\| x_p - m_j^{(t)} \right\|^2, \quad \forall j, 1 \leq j \leq k \right\},$$

   where each $x_p$ is assigned to exactly one $S^{(t)}$, even if it could be assigned to two or more of them.

3. **Update Step:**

   After assigning all points to the nearest centroids, recalculate the centroids of the clusters:

$$m_i^{(t+1)} = \frac{1}{\left| S_i^{(t)} \right|} \sum_{x_j \in S_i^{(t)}} x_j$$

4. **Repeat:**

   Repeat steps 2 and 3 until:

   - The centroids do not change significantly (i.e., convergence is reached).
   - OR the algorithm reaches the maximum number of iterations.

5. **Output the Clusters:**
   Once the algorithm has converged, return the clusters $S_1, S_2, \ldots, S_k$ and their final centroids $\mu_1, \mu_2, \ldots, \mu_k$.

The most critical challenge in clustering analysis is determining the appropriate number of clusters. Despite numerous studies on this topic, there is no universally conclusive method for addressing this issue. One of the earliest and most well-known approaches proposed to tackle this problem is:

$$k = \sqrt{n/2} \tag{1}$$

In Eq. (1), $k$ represents the number of clusters, and $n$ denotes the number of data points. This method is recommended for small sample studies, as it becomes challenging to achieve accurate results in large sample studies [6]. Another well-known approach is:

$$M = k^2 + |W| \tag{2}$$

In Eq. (2), $W$ represents the sum of squared distances within each cluster.

$$W = \sum_{j=1}^{k} \sum_{i=1}^{n_j} \left( (x_{ij} - x'_{ij})(x_{ij} - x'_{ij})^T \right) \tag{3}$$

In Eq. (3), $n_j$ represents the number of data points in the $j$-th cluster, $k$ denotes the number of clusters, $x_{ij}$ is the $i$-th value in the $j$-th cluster, and $x'_{ij}$ is the average vector of the $j$-th cluster.

A wide variety of proximity measurements can be used to calculate the closest centroid within the K-Means algorithm. And this choice can significantly affect the centerpoint assignment and the quality of the solution. The measurements that can be used are Euclidean distance, Manhattan distance and Cosine similarity. The Euclidean distance metric is the most popular choice for K-means clustering, but the Haversine formula is used in geographic clustering, logistics optimization, and school bus routing.

Haversine formula: The term "Haversine" comes from the mathematical function known as the haversine.

$$\text{Haversine}(\theta) = \sin^2(\theta/2) \tag{4}$$

Eq. (4) is adapted to incorporate latitude and longitude coordinates. The distance between two points with latitude and longitude coordinates $(\varphi_1, \psi_1)$ and $(\varphi_2, \psi_2)$ can be calculated using the Haversine formula:

$$d = 2r \sin^{-1} \sqrt{\sin^2 \frac{\varphi_2 - \varphi_1}{2} + \cos(\varphi_1) \cos(\varphi_2) \sin^2 \frac{\psi_2 - \psi_1}{2}} \tag{5}$$

The variable $d$ in a given Eq. (5), represents the distance between two points with latitude and longitude coordinates $(\psi, \varphi)$, and $r$ denotes the Earth's radius.

Haversine Formula is used to calculate distance between two points using latitude and longitude. It is important for use in navigation. This formula is particularly significant in navigation as it serves as the foundational equation for understanding distance calculations on a sphere.

### 3.2 Density Based Spatial Clustering of Applications with Noise (DBSCAN)

Density-based spatial clustering is a widely used data clustering algorithm. The of Applications with Noise (DBSCAN) algorithm groups regions of similar density into the same cluster, aiming to distinguish high-density clusters from low-density regions. It is particularly useful for identifying clusters of arbitrary shape and handling noise within the data.

In this algorithm Eps-neighborhood, is a specified Radius, MinPts is a specified number of points, Density is a number of points within a specified Radius

If a point has at least MinPts in the Eps neighborhood, this point is called a core point, if it is less than MinPts in the Eps but around the core point, this point is called a boundary point, if any point is not a core point or boundary point, this point is called a noise point [23, 24, 25, 26].

The steps of the DBSCAN algorithm are as follows:

Step 1. Initialization: For each point in the dataset, DBSCAN checks its neighborhood using the specified Eps (radius) value. The neighborhood of each point is calculated to determine the density around that point.

Step 2. Classification of Points: If the neighborhood of a point contains more than or equal to MinPts, the point is classified as a Core point, if a point is within the Eps-radius of a core point but has fewer than MinPts in its own neighborhood, it is classified as a Border point, if a point is neither a core point nor a border point, it is classified as a Noise point.

Step 3. Cluster Expansion: Starting with a core point, all other points within its Eps-neighborhood are added to the cluster. If a border point is within the neighborhood of a core point, it is added to the same cluster as the core point.

Step 4. Repeat for all Points: This process is repeated for all points in the dataset. New clusters are formed by expanding from core points. Once all points are processed, the algorithm finishes.

Step 5. Termination: The clustering process concludes when no new points can be added to any cluster. Noise points are left unclassified and are excluded from the final clusters.

Unlike K-means or other clustering algorithms, DBSCAN does not require the user to specify the number of clusters in advance. DBSCAN is effective in detecting noise points that do not belong to any cluster, which can be excluded from further analysis. DBSCAN is particularly useful for discovering clusters with irregular shapes, unlike algorithms like K-means that assume spherical clusters.

The accuracy of DBSCAN depends heavily on the choice of Eps and MinPts values. If these parameters are not chosen appropriately, the algorithm may fail to detect meaningful clusters or may classify too many points as noise. DBSCAN can struggle when clusters have significantly varying densities.

### 3.3 Hierarchical Clustering

Hierarchical clustering is a method that calculates the distance between two clusters in a dataset using specific linkage methods and combines them to form a cluster hierarchy [27, 4]. This approach builds a tree-like structure called a dendrogram, which illustrates the hierarchical relationships between data points. Unlike some clustering methods, hierarchical clustering does not require the number of clusters to be specified in advance [23, 24, 25, 26].

The steps of the hierarchical clustering are as follows:

Step 1. Initialization: Assign each item to its own cluster. For a dataset with N items, there are initially N clusters, each containing one item. The distances (or similarities) between clusters are set equal to the distances (or similarities) between the items they contain.

Step 2. Merge Closest Clusters: Identify the closest (most similar) pair of clusters based on the chosen distance metric and linkage method. Merge them into a single cluster, reducing the total number of clusters by one.

Step 3. Update Distance Matrix: Recalculate the distances (or similarities) between the newly formed cluster and all remaining clusters. The calculation depends on the selected linkage method.

Step 4. Repeat: Steps 2 and 3 are repeated iteratively until all data points are merged into a single cluster of size N.

Step 5. Construct Dendrogram: The hierarchical structure is visualized using a dendrogram, where the height of each merge represents the distance or dissimilarity between the merged clusters.

In hierarchical clustering, linkage methods determine how the distance between two clusters is calculated during the merging process. The choice of linkage method influences the shape and structure of the resulting clusters. The most commonly used linkage methods are:

Single Linkage: The distance between two clusters is defined as the shortest distance between any two points, one from each cluster.

$$d(r,s) = \min\{d(x_i, x_j) \mid x_i \in r, x_j \in s\}$$

where $d(r,s)$ is the distance between clusters $r$ and $s$, and $x_i$,$x_j$ are data points in the respective clusters.

Complete Linkage: The distance between two clusters is defined as the longest distance between any two points, one from each cluster.

$$d(r,s) = \max\{d(x_i, x_j) \mid x_i \in r, x_j \in s\}$$

Average Linkage: The distance between two clusters is calculated as the average distance between all pairs of points, one from each cluster.

$$d(r,s) = \frac{1}{|r| \cdot |s|} \sum_{x_i \in r} \sum_{x_j \in s} d(x_i, x_j),$$

where $|r|$ and $|s|$ are the sizes of clusters $r$ and $s$, respectively.

Ward's Linkage: The distance between two clusters is based on the increase in the total within-cluster variance after merging.

$$d(r,s) = \frac{|r| \cdot |s|}{|r| + |s|} \|c_r - c_s\|^2$$

where $c_r$ and $c_s$ are the centroids of clusters $r$ and $s$, respectively.

### 3.4 Gaussian Mixture Model

The Gaussian Mixture Model (GMM) is a powerful machine learning approach for modeling the underlying distributions of data. GMM assumes that the data consists of multiple Gaussian (Normal) distributions, each referred to as a component. The model probabilistically

assigns observations to these components and attempts to uncover the overall data structure by optimizing the parameters of each component [23, 24, 25, 26].

A GMM is a mixture of Gaussian distributions, where each distribution represents a component. It consists of K components. An observation $x_i$ is expressed as follows:

$$P(x_i) = \sum_{k=1}^{K} \pi_k N(x_i \mid \mu_k, \Sigma_k)$$

where, $\pi_k$ - is the weight of the $k$-th component in the mixture (mixture weight

$N(x_i \mid \mu_k, \Sigma_k)$ - The density function of the Gaussian distribution for the $k$-th component.

Parameters:

$\mu_k$ - Mean of the $k$ th component.

$\Sigma_k$ - covariance matrix of the $k$-th component.

### 3.4.1 Solution Process of GMM: Expectation-Maximisation Algorithm

The Expectation-Maximization (EM) algorithm is used to iteratively estimate the parameters of GMM. Below are the steps:

Initialization.

- Randomly initialize the parameters: Mixture weights $\pi_k$, means $\mu_k$ and covariance matrices $\Sigma_k$
- Define the number of components $K$.
- Input the dataset $\{x_1, x_2, \ldots, x_n\}$

1. Iterative Steps

   **Step 1. Expectation Step (E-Step)**

   Compute the posterior probability (responsibility) that each data point $x_i$ belongs to each component $k$:

   $$\gamma_{ik} = \frac{\pi_k N(x_i \mid \mu_k, \Sigma_k)}{\sum_{j=1}^{K} \pi_j N(x_i \mid \mu_j, \Sigma_j)}$$

   where: $\gamma_{ik}$: The probability that $x_i$ belongs to the $k$-th component.

   $\pi_k$: The mixture weight of the $k$-th component.

   $N(x_i \mid \mu_k, \Sigma_k)$: The Gaussian density function for the $k$-th component.

   **Step 2: Maximization Step (M-Step)**

   Update the model parameters using the computed responsibilities $\gamma_{ik}$:

   Mixture weights $\pi_k$:

   $$\pi_k = \frac{1}{N} \sum_{i=1}^{N} \gamma_{ik}$$

   Means $\mu_k$:

   $$\mu_k = \sum_{i=1}^{N} \gamma_{ik} x_i \Big/ \sum_{i=1}^{N} \gamma_{ik}$$

   Covariance matrices $\Sigma_k$:

   $$\Sigma_k = \sum_{i=1}^{N} \gamma_{ik} (x_i - \mu_k)(x_i - \mu_k)^T \Big/ \sum_{i=1}^{N} \gamma_{ik}$$

   **Step 3: Convergence Check**

   **Log-Likelihood Calculation:** Compute the log-likelihood of the data given the parameters:

$$L = \sum_{i=1}^{N} \log \left( \sum_{k=1}^{K} \pi_k N(x_i \mid \mu_k, \Sigma_k) \right)$$

**Stopping Criterion:** If the change in log-likelihood is smaller than a predefined threshold $\epsilon$, stop the iteration. Otherwise, repeat the E-Step and M-Step.

After convergence, the algorithm outputs the optimized parameters $pi_k$, $\mu_k$, $\Sigma_k$, probabilities $\gamma_{ik}$ for assigning data points to components, cluster assignments based on the highest probability component for each data point.

# 4 Computational Experiments

In this section, we will explain in more detail the data provided by the methods and the calculation results. In our study, in the K-Means algorithm, the data was divided into $k$ clusters using the Haversine distance metric for geographic data. In the DBSCAN algorithm, density-based clustering was performed with the neighborhood radius (epsilon) and MinPts (minimum number of neighbors) parameters. In the Hierarchical Clustering method, the distances between clusters were calculated with the average linking method and distance measurements were made using the Haversine distance metric. Finally, in the Gaussian Mixture Model (GMM), the data were modeled with Gaussian distributions with the cluster number and full covariance type parameters.

## 4.1 Data preparation

We tested the problem on randomly generated sample data and determined within the borders of Izmir (Türkiye) province for geographical analysis. For this purpose, 500 points were determined and latitude and longitude information were used as coordinates when determining the points. These points were generated using a random sampling method to ensure uniform distribution across the city's urban and suburban regions.

Preprocessing steps included normalization and distance metric. In normalization process, latitude and longitude values were normalized to standardize the scale for clustering algorithms and in distace metric process the Haversine formula was used to calculate the great-circle distance between points, which is essential for spatial clustering in spherical coordinates.

Four clustering algorithms, K-Means, DBSCAN, Hierarchical Clustering, and GMM algorithms were selected based on their popularity and applicability to spatial data.

## 4.2 Performance Metrics

To evaluate and compare the clustering outcomes, the following metrics were used:

**Silhouette Score:** Measures the quality of clustering by evaluating cohesion within clusters and separation between clusters. The silhouette ranges from -1 to +1, with higher scores indicating better-defined clusters.

**Davies-Bouldin Score:** The Davies-Bouldin score is a metric used to evaluate clustering algorithms. Quantifies cluster quality based on intra-cluster and inter-cluster distances. It is an internal evaluation scheme where how well the clustering is done is verified using quantities and properties specific to the dataset. Lower scores indicate more distinct and compact clusters.

**Running Time:** Running time is the length of time required to perform a computational process.

**Visual Analysis:** Map-based visualizations were created to assess the spatial coherence of clusters and their alignment with geographic regions.

### 4.3 Clustering Results

The performance of the algorithms is summarized in Tab. 1. These computational experiments were performed on the specified computer, which features an Intel(R) Core(TM) i7-8665U CPU (1.90 GHz-2.11 GHz), 32GB of RAM, and a 512GB SSD.

Table 1: Performance Comparison of Clustering Algorithms

| Algorithm | Silhouette Score | Davies-Bouldin Score | Running Time (s) |
|---|---|---|---|
| K-Means | 0.32 | 0.93 | 5.66 |
| DBSCAN | 0.14 | 5.43 | 2.79 |
| Hierarchical | 0.27 | 1.045 | 2.82 |
| Gaussian Mixture | 0.299 | 0.983 | 0.70 |

The epsilon value for DBSCAN was set to 3, determined using a $k$-NN graph. During the computation trials, the max_clusters parameter for Hierarchical Clustering was set to 5.

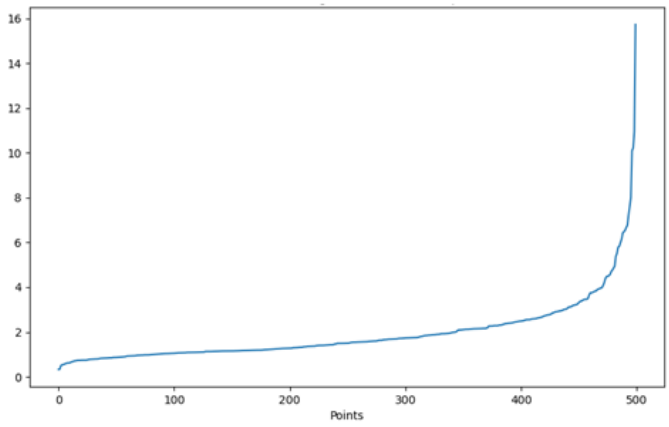


Figure 1: $k$-nearest neighbor distances to determine eps in DBSCAN

The clustering results, as shown in Fig. 2, visualize the performance of the algorithms. The results highlight the distinctiveness of the clusters formed by each algorithm.

## 5 Discussion

In terms of Silhouette Score, K-Means is an algorithm that efficiently clusters distinct and spherical data, achieving the highest value (0.32), suggesting that the clusters are well-separated. However, K-Means requires the number of clusters to be determined in advance and may perform poorly with non-spherical clusters.

In contrast, DBSCAN showed the lowest Silhouette score, indicating poor separation of clusters. Despite this, DBSCAN is robust to noise and capable of identifying clusters of any shape, though it suffers from parameter sensitivity and computational inefficiency for large datasets.

For exploratory analysis, Hierarchical Clustering is useful as it visualizes the nested structure of the data. However, it is computationally expensive and has limited scalability.

The Gaussian Mixture Model (GMM) provides a flexible probabilistic framework to model complex distributions. However, it is sensitive to the choice of covariance type and parameter initialization.

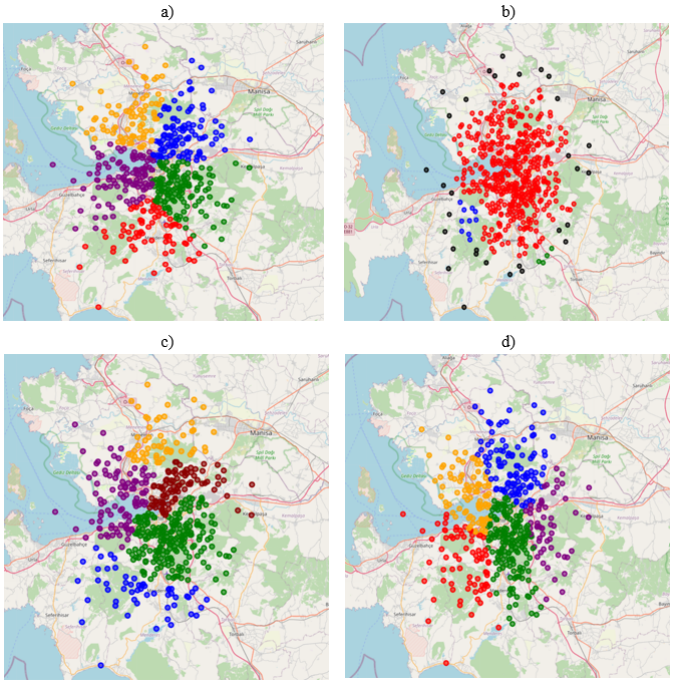


Figure 2: Results get with algorithms: a) K-means clustering, b) DBSCAN clustering, c) Hierarchical clustering, d) GMM clustering

## 6 Conclusion

This study provided a comparative analysis of K-Means, DBSCAN, Hierarchical Clustering, and GMM algorithms applied to 500 randomly selected geographical points in the Izmir region. K-Means performed the best overall, achieving both the highest Silhouette score and the lowest Davies-Bouldin score. On the other hand, DBSCAN performed the worst, with a high Davies-Bouldin score and a low Silhouette score, suggesting that the clusters were not well-defined.

While Gaussian Mixture Model (GMM) and Hierarchical Clustering showed moderate performance, GMM achieved slightly better results than Hierarchical Clustering. The findings emphasize the importance of choosing an appropriate clustering algorithm according to the data characteristics and application requirements.

In conclusion, this study not only contributes to the field by offering a detailed performance comparison of these algorithms within a specific geographical context, but also provides guidance for practitioners in selecting the most suitable method based on their data attributes and needs.

## References

[1] Bodin L., Golden B.L., Assad A., Ball M.O. (1983). *Routing and scheduling of vehicles and crews: the state of the art*, Comput. Oper. Res., 10.2, 63–211.

[2] Chapleau L., Ferland J.A., Rousseau J.M. (1985). *Clustering for routing in densely populated areas*, Eur. J. Oper. Res., 20.1, 48–57.

[3] Magnanti T.L. (1981). *Combinatorial optimization and vehicle routing*, Networks, 11.2, 179–213.

[4] Nuriyev U., Nuriyeva F. (2018). *Practical aspects of solving combinatorial optimization problems*, Adv. Math. Models Appl., 3.3, 179–191.

[5] Newton R.M., Thomas W.H. (1969). *Design of school bus routes by computer*, Socio-Econ. Plan. Sci., 3.1, 75–85.

[6] Braca J., Bramel J., Posner B., Simchi-Levi D. (1997). *A computerized approach to the New York City school bus routing problem*, IRT J. Math. Modelling Algorithms, 7.3, 263–293.

[7] Park J., Kim B.I. (2010). *The school bus routing problem: A review*, Eur. J. Oper. Res., 202.2, 311–319.

[8] Angel A., Beasley J., Richards J. (1972). *Computer-assisted school bus scheduling*, Oper. Res., 20.6, 279–288.

[9] Li X., Fu X. (2002). *A hybrid algorithm for the school bus routing problem*, Transp. Res. Part B: Methodol., 36.6, 535–549.

[10] Kinable J., Van Hentenryck P., Aissi H. (2014). *A hybrid heuristic for the school bus routing problem*, Transp. Res. Part C: Emerg. Technol., 46, 96–116.

[11] Kumar S., Vohra R., Arora A. (2015). *A metaheuristic approach to the school bus routing problem*, Eur. J. Oper. Res., 243.2, 576–588.

[12] Sarubbi J.F.M., Mesquita C.M.R., Wanner E.F., Santos V.F., Silva C.M. (2016). *A strategy for clustering students minimizing the number of bus stops for solving the school bus routing problem*, Proc. NOMS 2016 - 2016 IEEE/IFIP Netw. Oper. Manage. Symp., 1175–1180.

[13] Wang Z., Shafahi A., Haghani A. (2017). *SCDA: School Compatibility Decomposition Algorithm for Solving the Multi-School Bus Routing and Scheduling Problem.*

[14] Sun Y., Wang Z., Wang S. (2018). *School bus routing problem in the stochastic and time-dependent transportation network*, PLOS ONE, 13.8, e0202618.

[15] Ümit Ü.G., Kılıç F. (2019). *A school bus routing problem using genetic algorithm by reducing the number of buses*, Proc. of 2019 Innov. Intell. Syst. Appl. Conf. (ASYU), 1–6.

[16] Calvete H.I., GalΓ© C., Iranzo J.A., Toth P. (2021). *The school bus routing problem with student choice: A bilevel approach and a simple and effective metaheuristic*, Int. Trans. Oper. Res., 30.2, 1092–1119.

[17] Shang P., Yang L., Zeng Z. (2021). *Solving school bus routing problem with mixed-load allowance for multiple schools*, Comput. Ind. Eng., 151, 106978.

[18] Liu K., Zhang Y., Wang L. (2022). *Iterated clustering optimization of the split-delivery vehicle routing problem considering passenger walking distance*, Comput. Ind. Eng., 165, 107926.

[19] Farzadnia F., Bektas T., Lysgaard J. (2023). *The cumulative school bus routing problem: Polynomial-size formulations*, Networks, 82.4, 571–591.

[20] Melo J.M., Kramer J.S. (2024). *A multi-period approach for the school bus routing problem considering student stop selection and bus scheduling*, Comput. Oper. Res., 144, 105937.

[21] Jain A.K., Murty M.N., Flynn P.J. (1999). *Data clustering: A review*, ACM Comput. Surv., 31.3, 264–323.

[22] Aggarwal C.C., Reddy C.K. (2014). *Data clustering: Algorithms and applications*, CRC Press, Boca Raton.

[23] Hartigan J.A. (1975) *Clustering algorithms*, John Wiley & Sons, New York.

[24] Han J., Kamber M. (2006). *Data mining: Concepts and techniques*, 2nd ed., Morgan Kaufmann Publishers, San Francisco.

[25] Everitt S.B. (1974). *Cluster analysis*, Heinemann Educational Books, London.

[26] Marriott F.H.C. (1971). *Practical problems in a method of cluster analysis*, Biometrics, 27.3, 501–514, doi:10.2307/2528587.

[27] Nuriyeva F. (2019). *A method based on hierarchical clustering for Travelling Salesman Problem*, J. Mod. Technol. Eng., 4.2.

Fidan Nuriyeva,
Dokuz Eylul University, Izmir, Türkiye,
Institute of Control Systems, Baku, Azerbaijan,
Email: fidan.nuriyeva@deu.edu.tr,

Veysel Erdemci,
Denizbank Intertech, Istanbul, Türkiye,
Email: verdemci@gmail.com

Received 24.02.2025 , Revised 10.12.2024 Accepted 28.05.2025, Available online 30.06.2025.