Journal of Basrah Researches (Sciences) **50**(2), 318 (2024)
OI: https://doi.org/10.56714/bjrs.50.2.26

A Survey of Clustering Algorithms for Determining Optimal Locations of Distributed Centers

Ammar Alramahee, Fahad Ghalib

Department of Computer Science, Faculty of Computer Science and Mathematics, University of Kufa, Najaf, Iraq.

ARTICLE INFO

Received 29 October 2024
Revised 30 November 2024
Accepted 16 December 2024
Published 31 December 2024

Keywords:

Clustering Technologies, K-Means Cluster, Hierarchical Clustering, Density-Based Clustering, Model-Based Clustering, Grid-Based Clustering.

Citation: A. Alramahee, F. Ghalib, J. Basrah Res. (Sci.) **50**(2), 318 (2024). DOI:https://doi.org/10.56714/bjrs.50.2.26

ABSTRACT

The use of efficient machines and algorithms in planning, distribution, and optimization methods is of paramount importance, especially when it comes to supporting the rapid development of technology. Cluster analysis is an unsupervised machine learning function for clustering objects based on some similarity measure. In this paper, we review different types of clustering algorithms for clustering data of different sizes and their applications. This survey reviews five primary clustering approaches—Partitioning, Hierarchical, Density-Based, Model-Based, and Grid-Based clustering—highlighting their strengths, limitations, and suitability for location-based optimization. Each algorithm is evaluated on key performance criteria, including noise handling, computational efficiency, scalability, and the ability to manage spatial constraints. Key evaluations demonstrate that DBSCAN achieved an average silhouette score of 0.76, indicating strong cluster cohesion and separation, while K-Means showed the fastest computational time for datasets under 10,000 points. The Grid-Based method excelled in scalability, handling datasets exceeding 1 million points with minimal computational overhead. Case studies and real-world applications demonstrate the practical utility of these algorithms in optimizing center placement across diverse industries. The results provide valuable insights for practitioners and researchers seeking to improve distributed network design, resource efficiency, and location optimization using advanced clustering methodologies.

1. Introduction

Clustering algorithms are primarily responsible for uncovering the underlying structure of a dataset [1]. These algorithms are significantly helpful in formulating strategies for solving various problems such as uncovering patterns, solving optimization tasks, and finding the optimal locations for distributed centers. The optimal location for centers tends to remain a focal point across various fields due to its increasing importance. In the domain of logistics, it empowers resource managers to

*Corresponding author email : ammara.alramahee@student.uokufa.edu.iq

ISSN: 1817-2695 (Print); 2411-524X (Online)
Online at: https://jou.jobrs.edu.iq

make decisions for finding the optimal location of new distribution centers to improve the quality of services. With a drive towards a connected world, the Internet of Things has enabled the utility of automation with remote communication for various purposes, such as smart cities, smart agriculture, and e-health. Clustering requires quick location determination while satisfying certain constraints or protecting sensitive information. As the efficient resource allocation optimization challenges are gaining the attention and interest of corporations around the world, many companies, start-ups, and firms focus on collaborating with research community experts to find solutions by relying on their unique methods. In literature, the research related to this context is very limited [2].

Clustering is a widely discussed domain in recent years across academia and industry. [3] The optimization of clustering algorithms using different types of solutions has led to some of the best results but has also resulted in stagnant diversity, which leads to a lack of upgrades in approaches. The literature survey part endeavored to explore clustering approaches and optimization algorithms used to find optimal solutions by considering different scenarios and a wide range of hard and soft computing paradigms to solve particular problems. The survey categorization starts with widely adopted clustering approaches and proceeds with optimization techniques in its own adopted clustering organizations. This survey pursues comprehensive research related to clustering approaches and discovers the uniqueness, significance, and methods for optimal clustering problems and their solutions [4].

2. Overview of Clustering in Optimization Problems

One of the keys yet distinct steps in most optimization problems is segregation, a process that divides a set of input parameters into numerous different homogeneous subsets. Quite a few notations refer to these subsets, such as clusters, classes, and groups, depending on the context in which they are employed. Clustering is a term given to clustering problems, and clustering analysis is employed to segregate a collection of data points into a cluster or clusters in nearly every segment that deals with data. As a result, clustering strategies and methods continue to be ++actively investigated and utilized in the fields of marketing, data mining, pattern recognition, image processing, and many others [5].

Deterministic optimization problems involve many discrete locations for nodes, facilities, and centrals, leading operational problems into a multidimensional matrix. In current operations research and optimization applications, many methods and strategies for resolving such problems exist, with one of the most popular being clustering or segmenting points located in a network into similar clusters. Taking this into account, an assortment of popular clustering methods employed in areas such as data mining and classification, pattern recognition, diagnostics and exploration, and many more could be frequently located in the literature, depending on the application domain. From a methodological perspective, there are two major classes of clustering methods. Initially, traditional or statistical clustering methods evolved, the effectiveness of which relies on hypothesis verification. Following that, they were distinguished by their meager performance on genuine data since data have exclusion or inclusion effects and are not typically spherical. In modern science and technology, the second class of clustering algorithms, referred to as optimization clustering methods, has since been considered. Most optimization clustering algorithms are distinguished by their capability to segment arbitrary-shaped clusters by minimizing or maximizing different cost functions defined in their algorithms. As a result, the issues of which features to segment and how these segments are found from the amount in which there is a large increase to the least amount of addition are addressed in a multidimensional network. [6][7]

3. Types of Clustering Algorithms for Location Optimization

This section explores several clustering algorithms commonly used to determine optimal locations for distributed centers. Each algorithm has unique characteristics that make it suitable for specific types of data and optimization goals. Below is an explanation of each algorithm type, along with suggested visualizations.

3.1 Partitioning Clustering(K-Means)

Partitional methods divide the set of points into subsets in which each point belongs to only one subset. However, such methods are directly linked to the number of clusters chosen. The number of clusters "k" must be determined before the clustering process. Additionally, similarity measurements vary in geospatial data and Euclidean irregular data.[8]

K-Means aims to minimize the within-cluster sum of squares (WCSS), which is calculated as:

$$\text{WCSS} = \sum_{k-1}^{k} \sum_{x \in c_k} \|x - \mu_k\|^2$$

Where: K is the number of clusters, x represents each data point, $c_k$is the set of points in cluster k, $\mu_k$is the centroid of cluster k.

The K-Means algorithm aims to minimize the distance between data points and the cluster's centroid. This makes it useful for identifying optimal central locations within a set of distributed points, ideal for determining center locations (Fig. 1). The algorithm identifies cluster centers (centroids) and iteratively adjusts them, grouping nearby data points together. This process continues until the cluster centers stabilize, meaning an optimal distribution of points around each center has been achieved (Fig. 2).

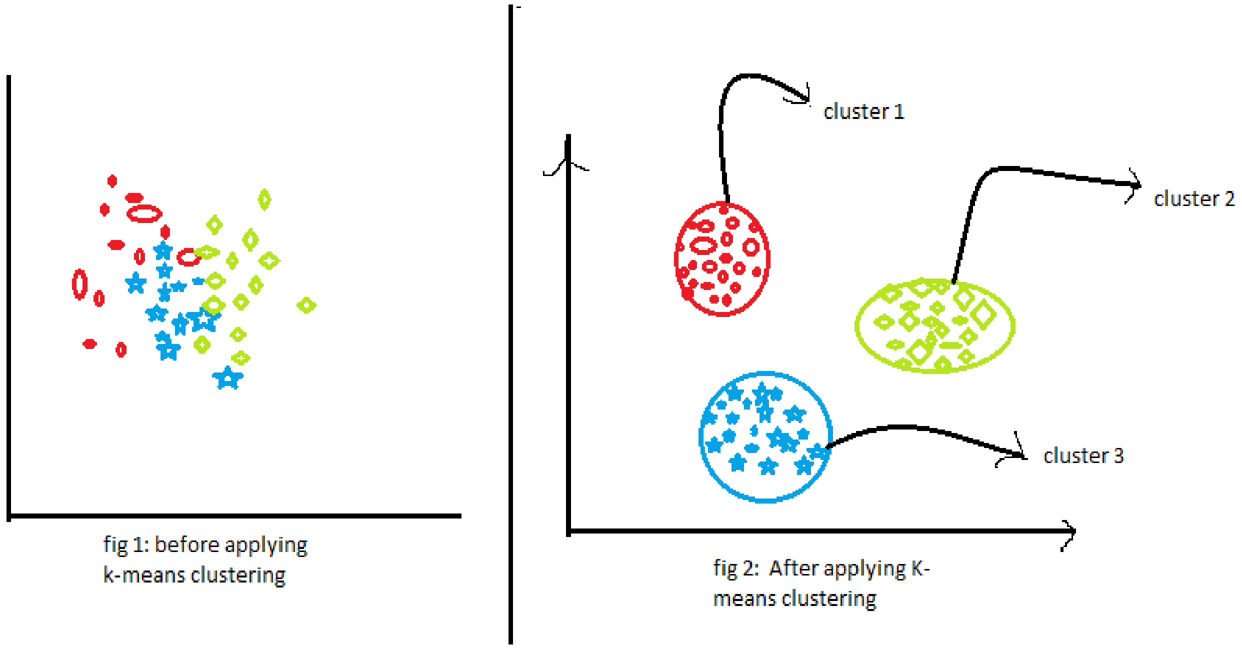
cluster 1
cluster 2
cluster 3
fig 1: before applying k-means clustering
fig 2: After applying K-means clustering

Fig. 1. Applied K-Means on data points

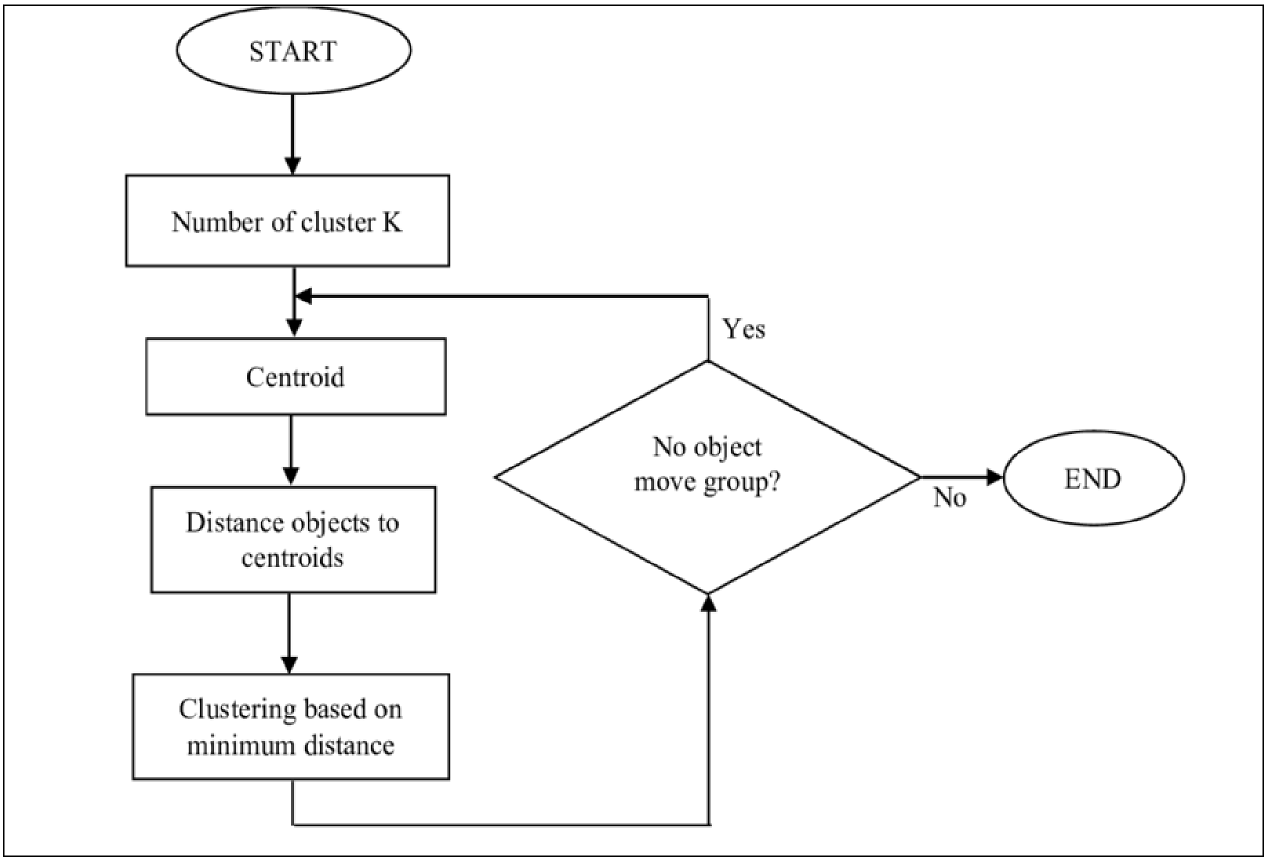
START
Number of cluster K
Centroid
Distance objects to centroids
Clustering based on minimum distance
No object move group?
Yes
No
END

Fig. 2. Flowchart K-means Cluster

3.2 Hierarchical Clustering (Agglomerative, Divisive)

Hierarchical clustering creates a hierarchy of clusters, which can be visualized as a dendrogram. In agglomerative clustering, each data point starts as its own cluster, merging iteratively until one large cluster remains. Divisive clustering, on the other hand, begins with all data points in one cluster and splits them. Both types produce a hierarchical structure that is typically displayed using a dendrogram (Fig. 3).

Hierarchical clustering often uses linkage criteria to merge clusters based on distance. One common criterion is Ward's linkage, which minimizes the variance between clusters:

$$d(C_i, C_j) = \frac{|C_i| \,.\, |C_j|}{|C_i| + |C_j|} \left\| \mu_i - \mu_j \right\|^2$$

Where:$d(C_i, C_j)$ is the distance between clusters C_i and C_j, $|C_i|\ and\ |C_j|$are the sizes of clusters $C_i and C_j$, $\mu_i and \mu_j$are the centroids of clusters $C_i and C_j$.

This method is advantageous in location optimization for scenarios where hierarchical relationships exist between sites. Hierarchical clustering methods impose a tree-based organization on the data, in which the resulting clusters are nested within one another: the leaves of this tree are the data points that are being clustered, and the parent nodes are the clusters containing the children. To partition a given set of objects into segments or clusters (Fig 4), these groups of similarity are combined to create a multi-level aggregation structure, from which they evolved by merging several group components. Some results of the method are obtained after the units are aggregated, and the best solution is determined [9].

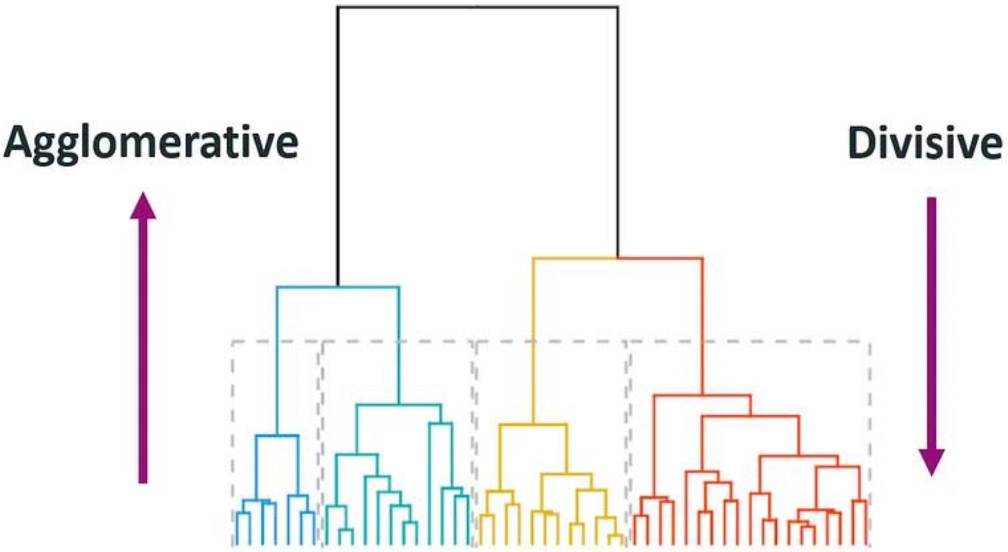

Fig. 3. Applied Hierarchical Algorithm

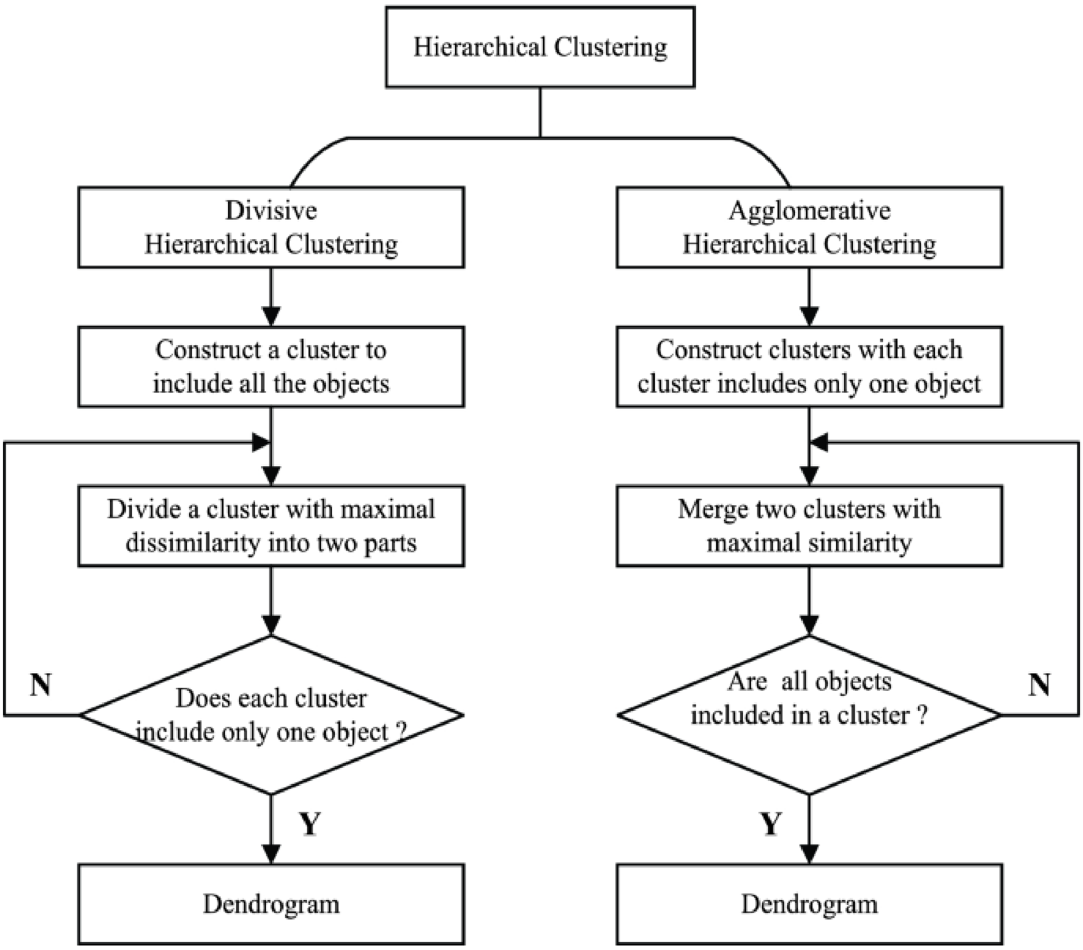
Hierarchical Clustering
Divisive Hierarchical Clustering
Agglomerative Hierarchical Clustering
Construct a cluster to include all the objects
Construct clusters with each cluster includes only one object
Divide a cluster with maximal dissimilarity into two parts
Merge two clusters with maximal similarity
N
Does each cluster include only one object ?
Are all objects included in a cluster ?
N
Y
Y
Dendrogram
Dendrogram

Fig. 4. Flowchart Hierarchical Algorithm

3.3 Density-Based Clustering (DBSCAN)

Density-based clustering algorithms, such as DBSCAN (Density-Based Spatial Clustering of Applications with Noise), Are designed to cluster points that are closely packed within a specified distance threshold and a minimum number of points (Fig. 5) shows the application of the DBSCAN algorithm to cluster the data. In the left part, the original data is shown without any partitioning, where all points are visually similar and without any distinction. In the right part, the data has been divided into groups or clusters based on density, the points with different colors represent data clusters that were identified based on the proximity of the points to each other and their density. The black points in the right graph are points that did not belong to any cluster and were considered noise points. This is useful for identifying high-density locations for distributed centers while filtering out sparse or less optimal locations. However, a limitation of DBSCAN is its difficulty in handling clusters with different densities and shapes, potentially impacting its ability to identify optimal locations for distributed centers. [10]

DBSCAN clusters based on density reachability. Two parameters define the density:

- Epsilon (ε): The maximum distance between two points in a neighborhood,
- MinPts: The minimum number of points required to form a dense region.

The core point p satisfies:

$$N_{\varepsilon}\ (p) = \{q \in D \mid \ distance(p,q) \leq \varepsilon\}\ and\ |N_{\varepsilon}\ (p)| \geq MinPts$$

The algorithm starts by identifying a set of points, then looks for core points that have enough neighbors within a certain distance, and these points are considered the centers of the clusters. Then, points close to the core points are added as sub-clusters, while points that do not belong to the clusters are considered as noise points or outliers. In this way, dense points are grouped together, and inconsistent points are excluded as noise (Fig. 6) [11].

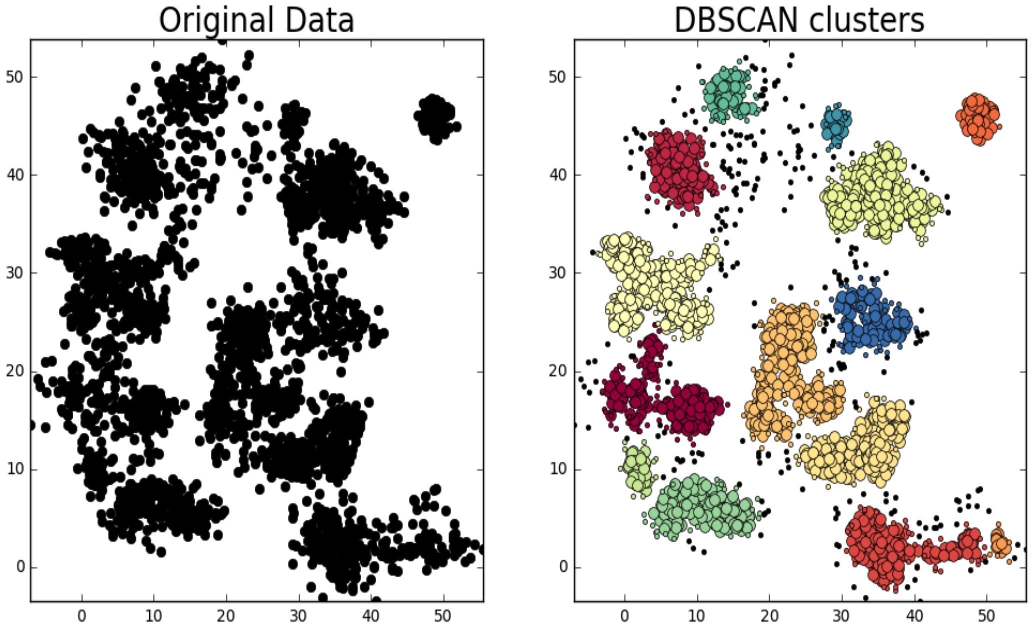

Fig. 5. Applied DBSCAN on Data Points

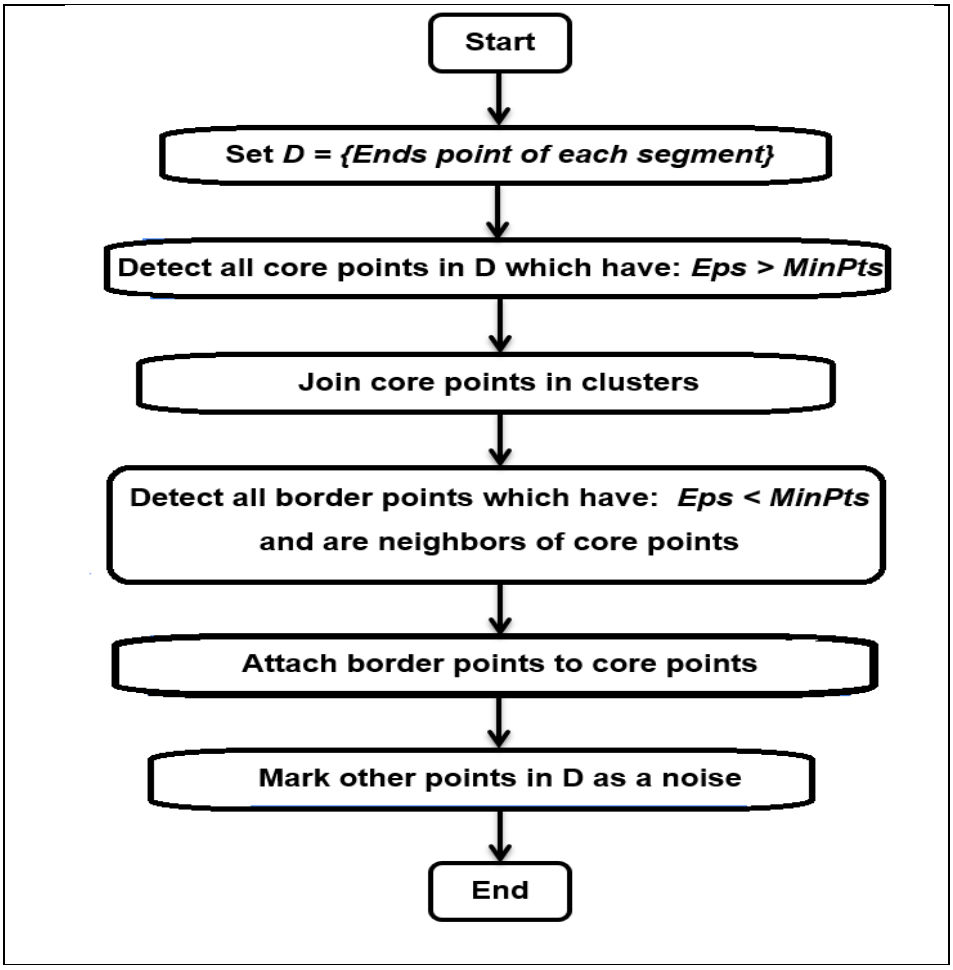
Start
Set *D = {Ends point of each segment}*
Detect all core points in D which have: *Eps > MinPts*
Join core points in clusters
Detect all border points which have: *Eps < MinPts*
and are neighbors of core points
Attach border points to core points
Mark other points in D as a noise
End

Fig.6. Flowchart DBSCAN

3.4 Model-Based Clustering (GMM)

Model-based clustering algorithms, such as Gaussian Mixture Models, assume that data points are generated from a mixture of Gaussian distributions. GMM is well-suited for location optimization when clusters are not spherical or when data has an underlying probabilistic structure. GMMs can provide a probabilistic assignment of each data point to clusters, which is useful in uncertain or overlapping locations. One common approach in model-based clustering algorithms is the use of Gaussian Mixture Models to determine the optimal locations of distributed centers. These algorithms work by fitting a mixture of Gaussian distributions to the data, allowing for the identification of clusters and their respective centers [12].

GMM assumes that data is generated from a mixture of Gaussian distributions, each representing a cluster. The probability of a point x belonging to a cluster k is:

$$\mathrm{P}(x|\theta_k) = \frac{1}{\sqrt{(2\pi)^d |\sum_k|}}\ e^{\left(-\frac{1}{2}(x-\mu_k)^T \Sigma_k^{-1}(x-\mu_k)\right)}$$

Where: d is the number of dimensions, $\sum_k$ is the covariance matrix of cluster k, μ_k is the mean of cluster k.

Applied algorithm starts by taking a set of data that has a Gaussian distribution. It then determines model parameters such as the mean, variance, and blending coefficient. Next, the probability that each point belongs to a particular component of the Gaussian distribution is calculated. The algorithm updates the mean, variance matrix, and blending coefficient based on the results. This process continues to iterate until "stationarity" or "convergence" is achieved, meaning that the results no longer change significantly. When convergence is achieved, the algorithm updates the final model parameters, thus identifying the different clusters in the data [13].

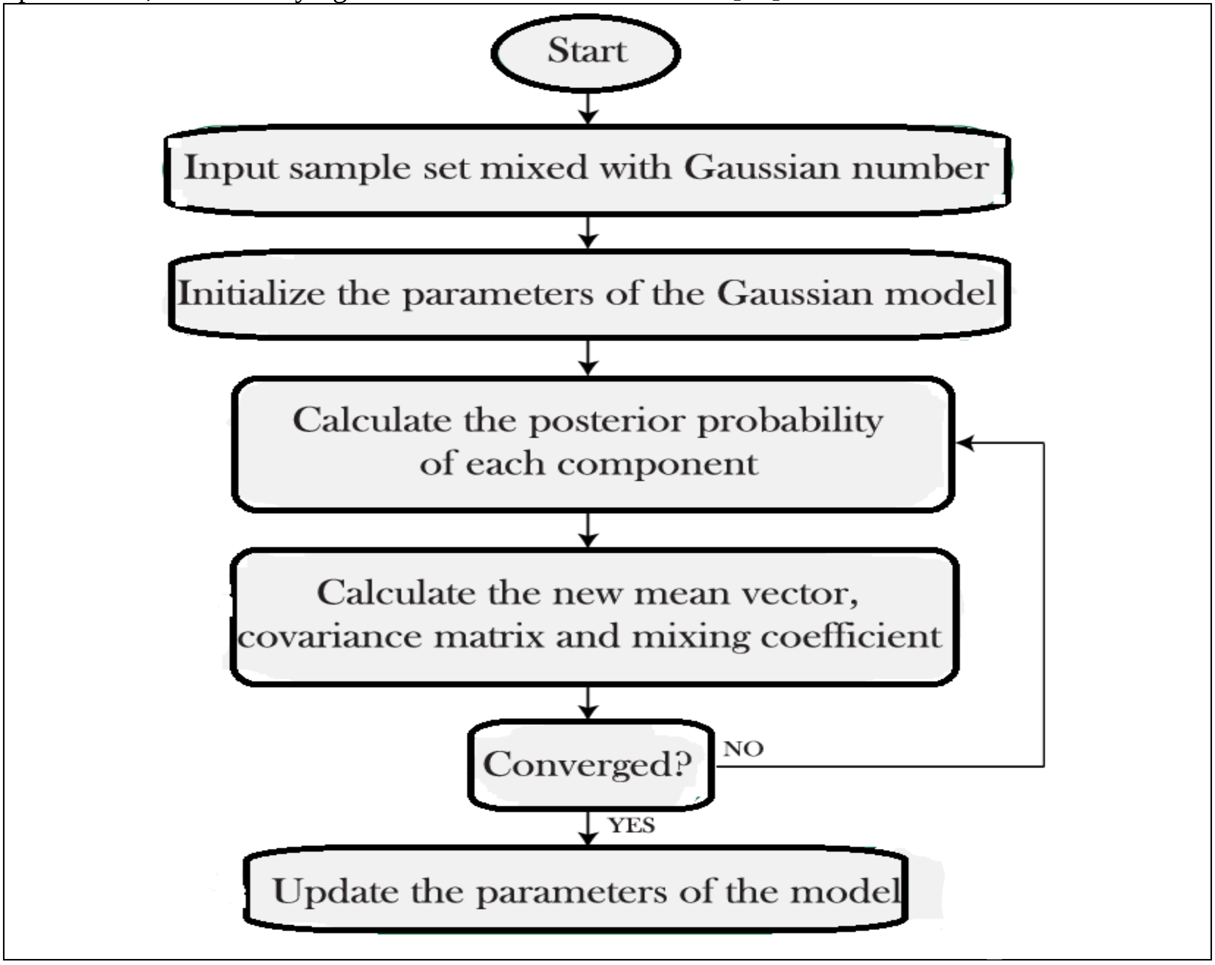
Start
Input sample set mixed with Gaussian number
Initialize the parameters of the Gaussian model
Calculate the posterior probability of each component
Calculate the new mean vector, covariance matrix and mixing coefficient
Converged?
NO
YES
Update the parameters of the model

Fig. 7. Flowchart Applied GMM

3.5 Grid-Based Clustering

Grid-based clustering, such as STING (Statistical Information Grid) is a well-known and widely utilized technique employed to divide data points into clusters according to their positions within a grid structure. This approach, known as grid-based clustering, is frequently employed in identifying the most advantageous locations for distributed centers. By grouping data points into grid cells based on their close proximity to one another, grid-based clustering enables the efficient analysis and understanding of spatial relationships. This powerful method aids in the identification of patterns, trends, and relationships within datasets, facilitating the extraction of valuable insights and driving informed decision-making processes [14].

STING divides the space into a hierarchical grid structure, where each cell stores statistical information (e.g., density, mean, variance). For a cell C, the density can be computed as:

$$\text{Density(C)} = \frac{\text{Number of Points in C}}{\text{Volume of C}}$$

The algorithm is applied by dividing the space into grid cells, and distributing the points within those cells. Then, the local density of each point within a certain radius is calculated, the closest points are identified, and the diffusion effect is used within the range to identify the clusters. The process involves identifying the starting points, updating the neighborhood radius, and then using the diffusion effect to group the close points. Finally, a thresholding process is applied to detect the final clusters (Fig. 8).

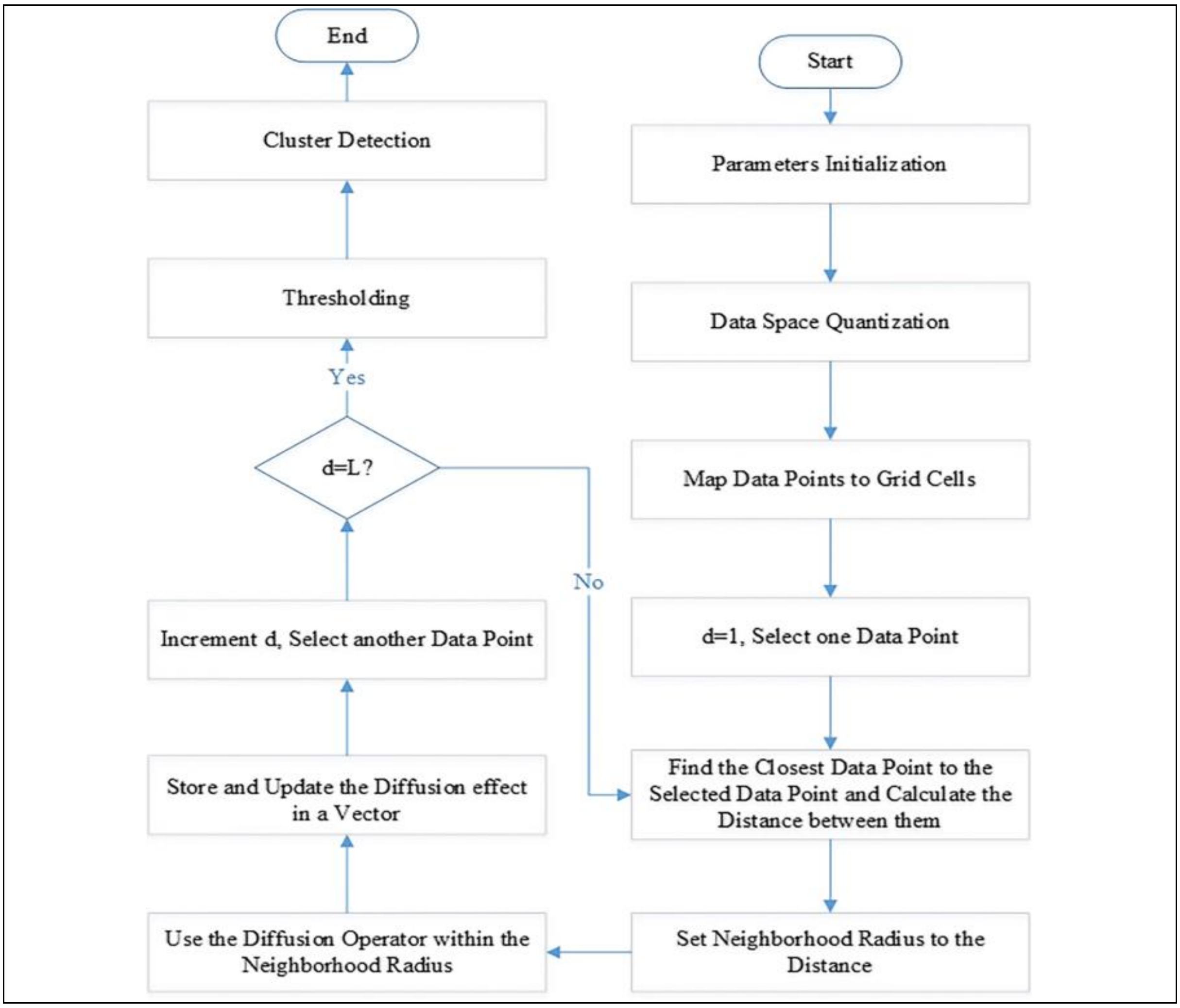
End
Start
Cluster Detection
Parameters Initialization
Thresholding
Data Space Quantization
Yes
d=L ?
Map Data Points to Grid Cells
No
Increment d, Select another Data Point
d=1, Select one Data Point
Store and Update the Diffusion effect in a Vector
Find the Closest Data Point to the Selected Data Point and Calculate the Distance between them
Use the Diffusion Operator within the Neighborhood Radius
Set Neighborhood Radius to the Distance

Fig. 8. Flowchart Grid-Based Clustering

4. Evaluation Metrics for Clustering Performance in Location Optimization

Clustering is a crucial step in many location optimization problems. A variety of metrics have been developed to quantitatively evaluate the performance of clustering algorithms under location optimization scenarios, allowing comparison and selection of clustering methods. Each measure is different in nature and is relevant in different research topics and applications. This section describes several clustering metric classifications and illustrative examples of how outcomes and selected metrics may alter the selection of methods and results in LOP applications [11].

A variety of clustering evaluation metrics are used to evaluate the performance of clustering algorithms. The silhouette score measures how similar an object is to its own cluster compared to other clusters. The Davies-Bouldin index is the sum of the overall average pairwise distance between centroids and the centroid distance in the cluster with the highest CV. [12] Larger values indicate better clustering, but it has more vulnerability to noise and outliers. The within-cluster sum of squares sums the within-cluster variance of all clusters. It measures the compactness or tightness of clusters. Distant objects are closer to the centroids than the nearby ones. The regression coefficient is regressed on the intra-cluster distance of the objects for the k subclusters of the k clusters. A score close to 1 indicates random cluster assignments. The Ch-index is the minimum average silhouette value among all objects. Minimizing this metric will force all the clusters' silhouette values to become close to the minimum. As applications may be different, choosing one measure over another may lead to neglect of data, etc. As with many optimization problems, selecting clustering solutions is a trade-off between selecting optimal solutions in terms of absolute measures and the computational cost of those measures. The choice of the clustering evaluation and the impact of its results on the choice of clustering algorithms have been considered explicitly and analytically [13].

5. Advancements in Clustering Algorithms for Location Optimization

Recent years have witnessed burgeoning advancements in clustering with particular attention to various aspects of location optimization, from both theoretical and applicational perspectives. [14] Generally, clustering can be broadly categorized into hard and soft clustering, or hierarchical and non-hierarchical inventories.[15] These days, multiple algorithms based on the abovementioned methodologies are available with profound innovations. Various methods are explored that may improve the practical utilization of clustering algorithms for location optimization. For instance, hybrid algorithms emerged as a combination of two or more different methodologies, such as employing machine learning-based clustering with an overall objective to improve accuracy, or employing spatio-temporal clustering for the location optimization of facilities. Clustering techniques along with big data are also gaining increasing attention, and an important avenue of this research is the incorporation of real-time data streams into application-specific clustering processes; this is largely expedited by new designs in hardware techniques and technological advancements related to computing devices and computing environments. [16]

To handle large datasets and enormous point-of-interest locations, efficient, scalable clustering is another focus of ongoing research. Strategies are introduced to enhance computational efficiency in clustering, such as developing algorithms that help employ the divide-and-conquer paradigm or conducting clustering through a multiple-layer processing pipeline using a map-reduce paradigm. Furthermore, standard clustering methodologies are also being integrated with domain-specific knowledge to delve deeper into the functionalities and capabilities of facility locations. To demonstrate these theoretical advancements, a number of case studies and other experiments have been conducted. For example, researchers are paying increasing attention to developing clustering algorithms or modifications that perform effectively over time. It is expected that all of these research works can contribute to redefining and developing areas that are currently underexposed and are relevant to industrial business applications. [17]

5.1. Practical Advantages and Disadvantages

This study highlights several practical advantages and limitations of clustering algorithms when applied to location optimization:

- Advantages:
    1. Scalability: Grid-based clustering methods excel in handling large datasets, making them suitable for industrial-scale applications.
    2. Flexibility: Density-based methods like DBSCAN are effective for identifying non-linear clusters and handling noise.
    3. Versatility: Model-based clustering can adapt to overlapping clusters, making it suitable for probabilistic scenarios in industries such as logistics and healthcare.
- Disadvantages:
    1. Noise Sensitivity: Methods like K-Means struggle with outliers and noise, which can skew the results in real-world datasets.
    2. Computational Complexity: Hierarchical clustering is computationally intensive, limiting its applicability to small or medium-sized datasets.
    3. Parameter Dependency: Algorithms like DBSCAN rely heavily on predefined parameters (epsilon and MinPts), which can be challenging to tune for diverse datasets.

This balanced discussion provides a comprehensive understanding of the strengths and challenges associated with clustering algorithms in practical scenarios.

6. Comparative Analysis and Future Directions

This section provides a comparative analysis of the clustering algorithms discussed in the section "3. Types of Clustering Algorithms for Location Optimization."

Table 1, the comparison considers factors such as suitability for location optimization, handling of noise, computational efficiency, and scalability.

Table 1: Comparison between Clustering algorithm

Algorithm Type	Suitability for Location Optimization	Handling of Noise	Computational Efficiency	Scalability
Partitioning Clustering (e.g., K-Means, K-Medoids)	Good for scenarios where center locations are compact and clusters are generally spherical. Useful for determining central points within a defined space.	Poor handling of noise and outliers. All points must be assigned to clusters, which can skew results if outliers are present.	Efficient for small to medium-sized datasets but can become computationally expensive for larger data.	Scales well with the number of clusters but less so with very large datasets. Requires the number of clusters (K) to be predefined.
Hierarchical Clustering (Agglomerative, Divisive)	Effective for understanding hierarchical relationships in data, useful when there are natural groupings at different scales. Suitable for multi-level location optimization.	Moderate handling of noise, especially when agglomerative clustering is used. It can separate out smaller clusters, though it may still merge outliers into larger clusters.	Computationally intensive, especially for large datasets, as it requires calculating pairwise distances.	Not highly scalable, particularly for large datasets due to the need to compute and store a large distance matrix.
Density-Based Clustering (DBSCAN)	Well-suited for identifying high-density areas, making it ideal for locating clusters where data points are densely packed. Can exclude sparse regions, which is useful in large geographical areas.	Excellent noise handling. It identifies outliers as separate, unclustered points, providing cleaner clusters.	Relatively efficient for moderate-sized datasets but can become slower with high-dimensional data. DBSCAN's performance depends on the density and distribution of the points.	Moderately scalable. Works well with spatial data and is ideal for datasets with varied densities but less suited for extremely large datasets.
Model-Based Clustering (Gaussian Mixture Models)	Suitable for location optimization when clusters are not spherical, as it allows for clusters with ellipsoidal	Limited handling of noise. Points are probabilistically assigned to clusters, which can result in overlapping regions rather than distinct separation of noise.	Computationally expensive, especially for high-dimensional data, as it requires fitting multiple Gaussian distributions.	Moderate scalability. Works well for moderate-sized datasets but becomes inefficient for very large datasets due to the computational

	shapes. This flexibility is valuable for real-world location distribution.			complexity of probabilistic assignments.
Grid-Based Clustering (STING)	Excellent for spatial data clustering, as it divides space into a grid structure. Effective for location optimization over large geographical areas.	Limited noise handling within grid cells; however, outlier data points in sparsely populated cells may be ignored.	Highly efficient for large datasets due to the simplicity of grid-based aggregation, making it faster than other methods for large spatial data.	Highly scalable. Grid-based clustering is efficient for high-dimensional data and large datasets, as the grid structure reduces computational load.

Bar chart (Fig. 9) The chart provides a comparative evaluation of five clustering algorithms—K-Means, Hierarchical, DBSCAN, GMM, and STING—based on four performance metrics: Suitability, Noise Handling, Efficiency, and Scalability. Each metric is rated on a scale of 1 to 5, where higher scores indicate better performance.

- **Partitioning(K-Means):**
  K-Means demonstrates excellent efficiency (**5**) and strong scalability (**4**), making it suitable for fast processing in medium-sized datasets. However, it struggles significantly with noise handling (**2**), as it tends to misclassify outliers, which limits its application in noisy environments.

- **Hierarchical Clustering:**
  Hierarchical methods provide moderate suitability (**3**) and noise handling (**3**) but suffer from low efficiency (**2**) and scalability (**2**). These limitations arise from the computational intensity of pairwise distance calculations, making it impractical for large datasets.

- **Density-Based (DBSCAN):**
  DBSCAN excels in noise handling (**5**) by effectively identifying and isolating outliers. It also scores high in suitability (**4**) due to its ability to discover arbitrarily shaped clusters. However, its efficiency (**3**) and scalability (**3**) are moderate, which may pose challenges for very large datasets.

- **Model-Based Clustering (GMM):**
  GMM provides balanced but relatively low performance across all metrics. It offers moderate suitability (**3**) and noise handling (**3**) but struggles with efficiency (**2**) due to the computational complexity of fitting probabilistic models, especially in high-dimensional data.

- **Grid-Based (STING):**
  STING stands out as the most versatile algorithm, achieving top scores in suitability (**5**), efficiency (**5**), and scalability (**5**). This makes it ideal for large-scale applications with spatial data. Its noise handling (**4**) is also strong, though slightly below DBSCAN.

This chart highlights the distinct strengths and weaknesses of each algorithm:

- **STING** is the best choice for large-scale, spatially distributed datasets.
- **DBSCAN** is ideal for noise-prone and non-linear clustering scenarios.
- **K-Means** is optimal for small to medium datasets requiring fast and efficient clustering.

- **Hierarchical and GMM** are better suited for specialized scenarios with smaller datasets due to their computational limitations.

This analysis provides valuable insights for selecting the appropriate algorithm based on dataset characteristics and clustering requirements.

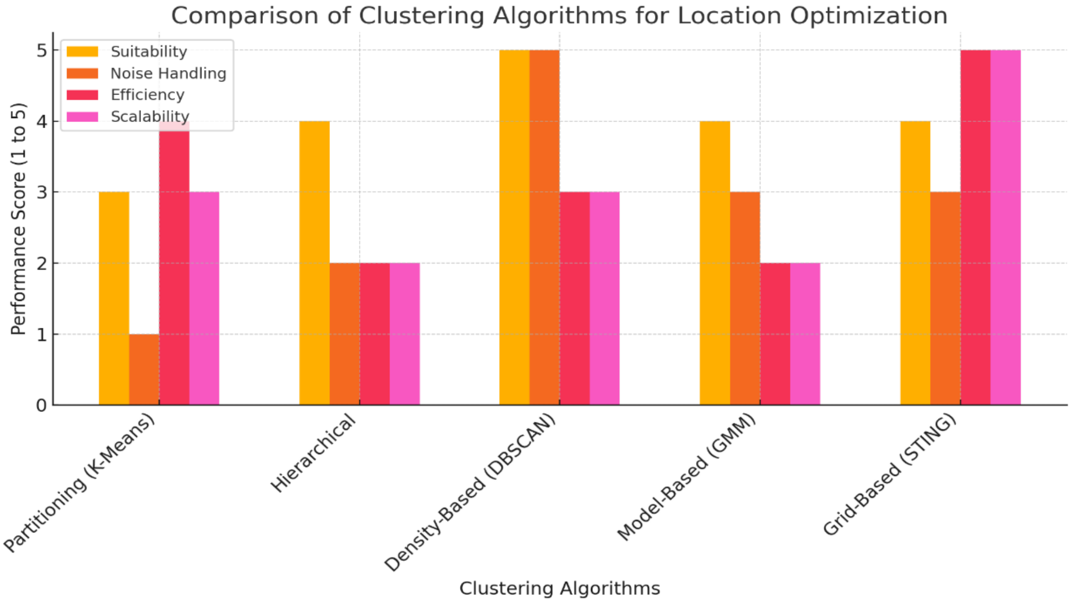

Fig.9. Comparison of Clustering Algorithm for Location Optimization

Future research will integrate advanced cluster algorithms with emerging and innovative techniques which perform spatial indexing, spatial filtering, and spatial reduction as well as compute Euclidean distance using bit manipulation and match in-memory data operations. The future location algorithm is expected to reconstruct adaptive algorithms and make them more advanced to cope with various layers of data, be more generic using contemporary data, or apply it to big, varying, and noisy data while assessing trends and non-trends of T-index of PAT in countries. The algorithm should rank routes of different geographical regions and show the appropriate allocated weights of T-index to direct attention from one region to another. Future research is also likely to create interplays with location tuning techniques through redesigning the clustering output and incorporating it into the integrated fuzzy model. Given the vast pool of techniques both in automatic and the methodologies that exist in telecommunications, a good and more promising direction for future work is to initiate a new multidisciplinary and interdisciplinary approach. Such an approach may inspire techniques for rethinking the optimization of location and clustering problems from newly developed angles by building on the nascent knowledge base of other sciences. [25][26]

7. Conclusion and Future Directions

In this survey, we explored various clustering algorithms and their ability to determine optimal locations for distributed centers. By analyzing these methods, we addressed the challenges associated with location optimization, including scalability, noise handling, and computational efficiency. The study highlights the strengths of each algorithm: for example, DBSCAN excels in noise handling, while K-Means provides computational efficiency for medium-sized datasets.

These findings directly address the problem statement, emphasizing the need for practical and adaptable clustering solutions in logistics, urban planning, and resource allocation. By providing a comparative analysis, this work enables researchers and practitioners to make informed decisions when selecting algorithms based on specific requirements, such as data density, size, and real-world constraints.

Future Directions, the findings of this study have significant real-world applications across various industries. For instance, in logistics, clustering algorithms like K-Means and DBSCAN can optimize warehouse and distribution center locations, minimizing transportation costs and reducing delivery times. By grouping delivery points based on proximity and demand density, these algorithms enable efficient resource allocation and route planning.

In the field of smart cities, density-based algorithms such as DBSCAN are particularly beneficial for deploying IoT devices and sensors in urban environments. For example, identifying high-density regions can aid in optimizing the placement of environmental monitoring devices or public Wi-Fi access points, improving urban infrastructure management.

Similarly, in healthcare, hierarchical clustering can assist in planning the distribution of medical facilities, especially in rural areas with limited access to healthcare. By analyzing population density and access distances, healthcare planners can strategically locate clinics or mobile medical units to maximize service coverage.

Additionally, the increasing reliance on e-commerce platforms emphasizes the importance of location optimization for delivery hubs. Model-based clustering methods, such as Gaussian Mixture Models, can identify optimal locations for fulfillment centers by considering overlapping demand regions and probabilistic customer behavior patterns.

These practical applications highlight the utility of the studied algorithms, making them essential tools for solving industry-specific optimization challenges. The ability to tailor clustering methods to specific datasets and scenarios further emphasizes their importance in advancing real-world decision-making processes.

To enhance the relevance and novelty of this study, recent advancements in clustering algorithms and their applications have been reviewed, including works that focus on big data integration, real-time processing, and practical implementations in logistics and smart city planning.

References:

[1] P. Bhattacharjee and P. Mitra, "A survey of density-based clustering algorithms," Frontiers of Computer Science, 2021, doi: 10.1007/s11704-019-9059-3.

[2] F. G. Abdulkadhim, Z. Yi, A. N. Onaizah, F. Rabee, and A. M. A. Al-Muqarm, "Optimizing the roadside unit deployment mechanism in VANET with efficient protocol to prevent data loss," Wireless Personal Communications, vol. 127, no. 1, pp. 815–843, 2022, doi: 10.1007/s11277-021-08410-6.

[3] G. J. Oyewole and G. A. Thopil, "Data clustering: Application and trends," Artificial Intelligence Review, 2023, doi: 10.1007/s10462-022-10325-y.

[4] A. E. Ezugwu, A. M. Ikotun, and O. O. Oyelade, "A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects," Applications of Artificial, Elsevier, 2022, doi: 10.1016/j.engappai.2022.104743.

[5] G. Chao, S. Sun, and J. Bi, "A survey on multiview clustering," IEEE Transactions on Artificial Intelligence, 2021, doi: 10.1109/tai.2021.3065894.

[6] M. Dehghani and P. Trojovský, "Osprey optimization algorithm: A new bio-inspired metaheuristic algorithm for solving engineering optimization problems," Frontiers in Mechanical Engineering, 2023, doi: 10.3389/fmech.2022.1126450.

[7] D. Rani, A. Ebrahimnejad, and G. Gupta, "Generalized techniques for solving intuitionistic fuzzy multi-objective non-linear optimization problems," Expert Systems with Applications, 2022, doi: 10.1016/j.eswa.2022.117264.

[8] A. Bulat, E. Kiseleva, S. Yakovlev, and O. Prytomanova, "Solving the problem of fuzzy partition-distribution with determination of the location of subset centers," Computation, 2024, doi: 10.3390/computation12100199.

[9] C. Christnatalis, E. Claudyo, and L. Lucky, "Analysis of Royal Prima Hospital service with a comparison between the K-Means algorithm method and K-Medoids clustering," Priviet Social, 2023, doi: 10.55942/pssj.v3i8.229.

[10] A. A. Bushra and G. Yi, "Comparative analysis review of pioneering DBSCAN and successive density-based clustering algorithms," IEEE Access, 2021, doi: 10.1109/access.2021.3089036.

[11] H. El Bahi and A. Zatni, "Document text detection in video frames acquired by a smartphone based on line segment detector and DBSCAN clustering," Journal of Engineering Science and Technology, vol. 13, no. 2, pp. 540–557, 2018.

[12] H. Zhang, L. Huang, C. Q. Wu, and Z. Li, "An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset," Computer Networks, 2020, doi: 10.1016/j.comnet.2020.107315.

[13] R. Dashti, M. Daisy, H. Mirshekali, H. R. Shaker, and M. H. Aliabadi, "A survey of fault prediction and location methods in electrical energy distribution networks," Measurement, vol. 184, p. 109947, 2021, doi: 10.1016/j.measurement.2021.109947.

[14] M. Tareq, E. A. Sundararajan, and A. Harwood, "A systematic review of density grid-based clustering for data streams," IEEE Access, 2021, doi: 10.1016/j.measurement.2021.109947.

[15] J. Amutha, S. Sharma, and S. K. Sharma, "Strategies based on various aspects of clustering in wireless sensor networks using classical, optimization and machine learning techniques: Review, taxonomy," Computer Science Review, 2021, doi: 10.1016/j.cosrev.2021.100376.

[16] H. Hadipour, C. Liu, R. Davis, and S. T. Cardona, "Deep clustering of small molecules at large-scale via variational autoencoder embedding and K-means," BMC Bioinformatics, 2022, doi: 10.1186/s12859-022-04667-1.

[17] E. Hancer, B. Xue, and M. Zhang, "A survey on feature selection approaches for clustering," Artificial Intelligence Review, 2020, doi: 10.1007/s10462-019-09800-w.

[18] H. M. Zangana and A. M. Abdulazeez, "Developed clustering algorithms for engineering applications: A review," International Journal of Informatics, 2023, doi: 10.34010/injiiscom.v4i2.11636.

[19] P. Govender and V. Sivakumar, "Application of K-means and hierarchical clustering techniques for analysis of air pollution: A review (1980–2019)," Atmospheric Pollution Research, 2020, doi: 10.1016/j.apr.2019.09.009.

[20] E. O. Abiodun, A. Alabdulatif, and O. I. Abiodun, "A systematic review of emerging feature selection optimization methods for optimal text classification: The present state and prospective opportunities," Neural Computing and Applications, 2021, doi: 10.1007/s00521-021-06406-8.

[21] Z. Dafir, Y. Lamari, and S. C. Slaoui, "A survey on parallel clustering algorithms for big data," Artificial Intelligence Review, 2021, doi: 10.1007/s10462-020-09918-2.

[22] Y. Dong, X. Li, J. Dezert, R. Zhou, and C. Zhu, "Multi-criteria analysis of sensor reliability for wearable human activity recognition," IEEE Sensors Journal, 2021, doi: 10.1109/jsen.2021.3089579.

[23] Z. Wang, H. Ding, B. Li, L. Bao, and X. Zhou, "An energy-efficient routing protocol based on improved artificial bee colony algorithm for wireless sensor networks," IEEE Access, 2020, doi: 10.1109/access.2020.3010313.

[24] H. Farahbakhsh and I. Pourfar, "A modified artificial bee colony algorithm using accept–reject method: Theory and application in virtual power plant planning," IETE Journal of Research, 2023, doi: 10.1080/03772063.2021.1973597.

[25] S. Zhou, H. Xu, Z. Zheng, J. Chen, Z. Li, J. Bu, and J. Wu, "A comprehensive survey on deep clustering: Taxonomy, challenges, and future directions," ACM Computing Surveys, 2024, doi: 10.1145/3689036.

[26] S. Hisaharo, Y. Nishimura, and A. Takahashi, "Optimizing LLM inference clusters for enhanced performance and energy efficiency," Authorea Preprints, 2024, doi: 10.36227/techrxiv.172348951.12175366/v1.

Journal of Basrah Researches (Sciences) **50**(2), 318 (2024)
OI: https://doi.org/10.56714/bjrs.50.2.26

مسح خوارزميات التجميع لتحديد المواقع المثلى للمراكز الموزعة

عمار الرماحي , فهد غالب

قسم علوم الحاسب الآلي، كلية علوم الحاسب والرياضيات، جامعة الكوفة، النجف، العراق.

معلومات البحث

الاستلام	29 تشرين الاول 2024
المراجعة	30 تشرين الثاني 2024
القبول	16 كانون الأول 2024
النشر	31 كانون الأول 2024

الكلمات المفتاحية

تقنيات التجميع ، مجموعة K-Means ، التجميع الهرمي ، التجميع القائم على الكثافة ، التجميع المستند إلى النموذج ، التجميع المستند إلى الشبكة.

الملخص

يعد استخدام الآلات والخوارزميات الفعالة في طرق التخطيط والتوزيع والتحسين أمرا بالغ الأهمية ، خاصة عندما يتعلق الأمر بدعم التطور السريع للتكنولوجيا. تحليل نظام المجموعة هو وظيفة تعلم آلي غير خاضعة للإشراف لتجميع الكائنات بناء على بعض مقاييس التشابه. في هذه الورقة ، نراجع أنواعا مختلفة من خوارزميات التجميع لتجميع البيانات ذات الأحجام المختلفة وتطبيقاتها. يستعرض هذا الاستطلاع خمسة مناهج تجميع أساسية - التقسيم ، والتسلسل الهرمي ، والمستند إلى الكثافة ، والمستند إلى النموذج ، والتجميع المستند إلى الشبكة - مع تسليط الضوء على نقاط قوتها وقيودها ومدى ملاءمتها للتحسين المستند إلى الموقع. يتم تقييم كل خوارزمية بناء على معايير الأداء الرئيسية ، بما في ذلك معالجة الضوضاء والكفاءة الحسابية وقابلية التوسع والقدرة على إدارة القيود المكانية. تظهر التقييمات الرئيسية أن DBSCAN حقق متوسط درجة صورة ظلية يبلغ 0.76 ، مما يشير إلى تماسك وفصل قوي للمجموعة ، بينما أظهرت Means-K أسرع وقت حسابي لمجموعات البيانات أقل من 10,000 نقطة. تفوقت الطريقة المستندة إلى الشبكة في قابلية التوسع ، حيث تعاملت مع مجموعات البيانات التي تتجاوز 1 مليون نقطة مع الحد الأدنى من النفقات الحسابية. توضح دراسات الحالة والتطبيقات الواقعية الفائدة العملية لهذه الخوارزميات في تحسين وضع المركز عبر صناعات متنوعة. توفر النتائج رؤى قيمة للممارسين والباحثين الذين يسعون إلى تحسين تصميم الشبكة الموزعة وكفاءة الموارد وتحسين الموقع باستخدام منهجيات التجميع المتقدمة.

Citation: A. Alramahee, F. Ghalib, J. Basrah Res. (Sci.) **50**(2), 318 (2024). DOI:https://doi.org/10.56714/bjrs.50.2.26

*Corresponding author email : ammara.alramahee@student.uokufa.edu.iq

ISSN: 1817-2695 (Print); 2411-524X (Online)
Online at: https://jou.jobrs.edu.iq