Journal of Information & Optimization Sciences
ISSN 0252-2667 (Print), ISSN 2169-0103 (Online)
Vol. 46 (2025), No. 5, pp. 1459–1478
DOI : 10.47974/JIOS-1603

Integrated mobility for smart cities

Ghada Abdelhady *
Department of General Systems Engineering
October University for Modern Sciences and Arts
Giza
Egypt

Omar M. Anwar †
Back Office Core Circuit Switching
IP Multimedia SubSystem and International Gateway Engineer
Huawei
Cairo
Egypt

Mariam Hassib §
Department of Electrical Communication and Electronics
October University for Modern Sciences and Arts
Giza
Egypt

Abstract

The issue of congestion and traffic remains unresolved in many countries, with significant time spent on trip planning. Thus, the proposed smart mobility system aims to improve travel planning and decision-making for city travelers. This research assists anyone wanting to explore a city within a specific timeframe by offering the most efficient travel options. It focuses on minimizing travel time and costs. The study began by examining clustering techniques and optimization algorithms, leading to the implementation of a hybrid algorithm. The hybrid system integrates the "Ant Colony Optimization" algorithm with the "K-means clustering" technique to determine the shortest paths, thereby coordinating

This article has been corrected with minor changes. These changes do not impact the academic content of the article.

* *E-mail:* **gabdelmouez@msa.edu.eg (Corresponding Author)**
† *E-mail:* **omar.mohammed17@msa.edu.eg**
§ *E-mail:* **mariam.hassib@msa.edu.eg**

© Taru Publications

transportation means and supporting climate change efforts, ultimately contributing to sustainability and integrated mobility in Egypt's smart cities.

Subject Classification: *Primary 90B36 (Programming problems (scheduling, routing, etc.), 68M20, Secondary 90B20 (Traffic problems in operations research).*

Keywords: *Ant colony, City traveller, Integrated mobility, K-means algorithm, Optimization, Pathfinding, Smart mobility system, Traffic optimization, Travel planning.*

1. Introduction

The integration of machine learning and Internet of Things (IoT) technologies into urban infrastructure has gained significant traction in recent years. Sharma and Jailia [1] explored the potential of these advancements in waste management through a study on machine learning and IoT -based garbage detection systems for smart cities. Their study proposes a machine learning -based garbage detection system utilizing Internet of Things (IoT) devices. An IP -enabled CCTV camera, acting as an IoT gadget, captures images and transmits them to a central server. These images are pre-processed in Python, converting them into numerical representations that are handled by the selected machine learning algorithms. The data is then split into training and testing sets (80/20 split), allowing the model to learn and evaluate its performance. A variety of machine learning models including "K-Nearest Neighbours" (KNN), "Logistic Regression" (LR), "Naive Bayes" (NB), and "Support Vector Machines" (SVM) are then employed to classify the images into 'garbage' and 'no garbage' categories This research builds upon their work by delving deeper into the development of smart cities through a novel hybrid system that combines the K-means algorithm and ant colony optimization [1].

It has been noted that considerable time is dedicated to planning travel, and many individuals are not fully aware of the advantages offered by a smart travel system. Consequently, organizing and enjoying a simple trip, especially in a foreign city, can be challenging. To aid travelers in preparing an excellent tour, information about current events, amusements, and facilities at the required destination should be more accessible and enhanced.

One such system is "Smart City Traveler (Planned time to return)", which aims to identify the primary computing requirements to support the enhancement of tourist promotion points and devise a travel route that minimizes factors such as trip time, cost, and other variables. Additionally,

the "Smart Trip Planner" system is designed to create planned routes for foreigners depending on their preferences, thereby facilitating their journeys. This system gathers information about user preferences from a "MySQL" database to generate the planned trip. Furthermore, the social analysis addresses congestion problems in smart cities by providing efficient traffic infrastructure. [2], [3], [4]

Recent research has focused on systems for Integrated Mobility in Smart Cities, presenting the use of "Ant Colony" and "Clustering algorithms" with "K-means", along with a "smart trip planner" represented in the user interface.

1.1 *"Smart City Traveler" System*

It aims to determine the primary computation needed to enhance tourist promotion points for travelers and develop travel routes that minimize the objective function, including trip time, cost, and others, considering the following constraints:

a. Customer preferences for each node are based on popularity and historical data.
b. Congestion levels at every node.
c. Time Constraints for traveling.

The system also takes into account the "planned time to return" This method provides a tour route and a sightseeing schedule, automatically guiding users from a specific start time until their return. It performs an automated search for local points of interest, ensuring tourists do not miss any attractions they are unfamiliar with. User interests and preferences are considered, and the system selects places accordingly. Sightseeing information is categorized and listed based on different sight -seeing categories [2].

In the current system, users must accurately specify the location name. If a sight-seeing destination is selected, users typically have no issues as the Google map displays the best places near them. However, users must independently select each destination and then find routes for each one. A significant advantage of this system is its capability to propose a sight-seeing path and schedule within the allotted return time. The current tourist guide system requires users to manually enter each visit. Consequently, travelers need to plan for sightseeing locations themselves, limiting visits to destinations they are aware of. While well-known tourist

attractions can be easily researched online or in guidebooks, many remarkable locations may remain unknown to travelers if they are not famous tourist spots. The system is achieved using many algorithms [2], including:

1. The enhanced "Personalized Route Recommendation (PRR)" Algorithm, which relies on "Bayes' Theorem" and assumes the independence among predictors.
2. The "Ant Colony Algorithm" that is an optimization and problem-solving technique. It is inspired by ants' behavior to solve the traffic problems throughout the following three main mechanisms:
   - *Mechanism Selection:* A higher pheromone line on a route increases the probability of it being chosen.
   - *Mechanism Updating:* The pheromone on the line attracts as more as ants travel along it, but over time, it volatilizes and dissipates.
   - *Mechanism Coordination:* Ants communicate and operate in a coordinated manner via the "pheromone."
3. "A* Heuristic algorithm": It is a knowledgeable search algorithm commonly applied in pathfinding and graph traversal. As previously mentioned, the ant colony algorithm utilizes this optimized mechanism, allowing it to discover the best solution through information exchange and mutual cooperation among individuals, demonstrating a robust capability to find optimal solutions.

Thus, the system displays paths that guide both visually on a map and verbally. By choosing one of the categories, a menu showing the main types of attractions in the desired location is presented, along with all relevant facilities and services for that category.

1.2 *The Smart Trip Planner System*

Smart tourism has evolved into digital formats, utilizing new technologies like personal assistants and recommendation systems to tailor travel experiences based on users' preferences.

This system is designed to generate planned tours for tourists based on their preferences, making their trips easier to plan. It typically gathers data about user selections from a "MySQL" database to create the planned tour. The system's architecture is illustrated in Fig. 1. It shows that users

select their source and destination, travel date, and other destination-related preferences [3].

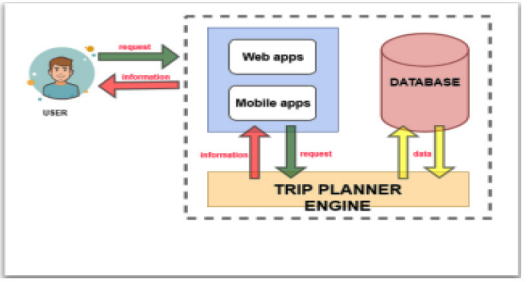


Figure 1

Smart Trip Planner Architecture [3]

It is composed of three components:

- **Trip Planner Engine**: It accesses the database to retrieve Points of Interest (POI) for the specified destination, generating the user's planned tour. It employs a clustering algorithm to recommend destinations based on user preferences saved in the database. The algorithm ensures no time conflicts exist for each POI.
- **Database**: As mentioned before "MySQL" is utilized to store information in connected tables.
- **User Interface**: Mobile and web apps are available for users to create their requests.

Fig. 2 presents the output of this prototype. Users begin by selecting their location and destination. They also set start and dates the start and end dates, to set the itinerary length for each day. The trip planner engine then processes this information. It accesses the database to fetch POI

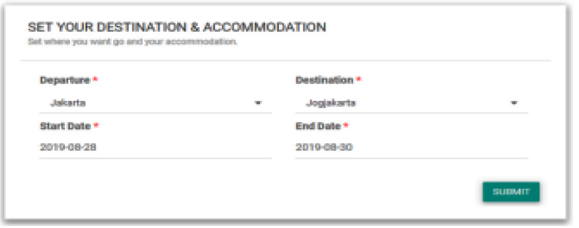


Figure 2

The smart planner prototype output [3]

objects for the destination. Using the available POI data, the trip planner engine calculates and creates the daily itinerary [3], [5].

The results explain that the clustering algorithm guarantees there are no time collisions for any POI. This approach benefits the suggested system by leveraging prototype and user interface. [3]

1.3 *Smart City Social Analysis System*

It is designed to address congestion issues in smart cities and provide an optimized traffic infrastructure.

It incorporates an application cloud server capable of generating text messages and acting as a bridge between nodes and the application cloud server. Implementation is through the Microsoft Azure IoT cloud server due to its enhanced features. A smart city graphical representation is depicted in Fig. 3, where the graph comprises edges and nodes formulating in an adjacency matrix. [4]

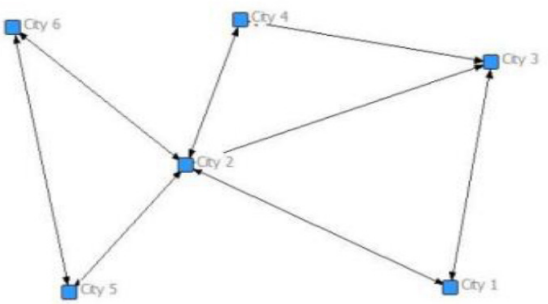


Figure 3

Connected nodes Graph [4]

The adjacency matrix, as shown in Fig. 3, is a square and two-dimensional matrix whose dimensions are the number of nodes, which in this example is 6. The number of input nodes is limited to prevent the matrix from becoming too large. Nodes represent cities, and edges symbolize connections between them. The cities that are not directly connected can communicate via another city using the shortest path algorithm.[4]

Nodes denote cities, and edges signify their connections. In Table 1, the adjacency matrix, '0' indicates no connection, and '1' indicates an existing connection.

Table 1

A 6x6 adjacency matrix [4]

	City 1	City 2	City 3	City 4	City 5	City 6
City 1	0	1	1	0	0	0
City 2	1	0	1	1	1	1
City 3	1	0	0	1	0	0
City 4	0	1	1	0	0	0
City 5	0	1	0	0	0	1
City 6	0	1	0	0	1	0

Additionally, the illustrated higher-level system involves clustering technique using the adjacency matrix. Here, nodes represent clusters of cities, and edges denote specific cities that serve as interfaces between clusters. This method improves communication range between distant cities.

This paper aims to develop a system to resolve congestion and traffic issues, find the shortest path between nodes, and optimize traffic flow in smart cities. It also plans to help tourists navigate to their destinations.

It describes a hybrid system combining clustering and ant colony techniques. Section 2 covers clustering and state-of-the-art algorithms for congestion solutions. Then, it discusses software analysis based on the System Development Life Cycle (SDLC) and agile methodology, followed by the selected clustering technique combined with the ant colony method for optimal pathfinding. Section 4 presents the results and performance evaluations.

2. Clustering

2.1 *Overview*

There are several clustering algorithms, including partitioning, hierarchical, density-based, and model-based methods.[6] Clustering is widely used in machine learning and data analysis to group similar data points. Common clustering types include:[5], [6]

K-means Clustering: Partitions data into k clusters based on minimizing squared distances between data points and cluster centroids. [7]

Hierarchical Clustering Forms a hierarchy of clusters, either agglomerative (bottom-up) or divisive (top-down). Agglomerative clustering merges closest pairs of clusters until one remains, while divisive clustering splits the dataset recursively until each data point is separate.[5]

Density-Based Spatial Clustering of Applications with Noise (DBSCAN): Groups of data points based on density, identifying clusters as high-density areas separated by low-density regions. It handles arbitrary shapes and outliers well.[5]

Gaussian Mixture Models (GMM): Uses Gaussian distributions to represent clusters, assuming data points within a cluster are generated from a mixture of these distributions. GMM assigns a probability to each data point for cluster membership.[7]

Mean Shift Clustering: Mean Shift iteratively identifies dense regions of data points by calculating the mean of data points within a given neighbourhood. It shifts the center of the neighborhood towards the region of higher density, ultimately converging to the modes of the data distribution. [8]

Spectral Clustering: It treats data points as nodes in a graph and performs clustering based on the spectral features of the graph Laplacian matrix. It projects the data into a lower-dimensional space and then applies a clustering algorithm on the transformed data. [5].

Affinity Propagation: Affinity Propagation does not require specifying the number of clusters in advance. It uses a message-passing technique to let data points communicate and determine exemplars, which represent the most representative points in the data set. [5]

The clustering method selection depends on the specific requirements and the nature of the data being analyzed.

2.2 *Clustering Algorithm Selection*

Different clustering algorithms have their own strengths and weaknesses, and their performance can vary based on the data's characteristics.

K-means clustering is often favored for its simplicity and ease of implementation. It is efficient and widely used, but it assumes clusters are spherical, equally sized, and have similar densities, which may not always be true in real-world datasets.[7]

Hierarchical clustering is intuitive and easy to interpret due to its creation of a cluster hierarchy. However, it can be computationally intensive for large datasets.

DBSCAN is user-friendly and can detect clusters of arbitrary shapes. It is robust to noise and doesn't require a predefined number of clusters. However, determining the appropriate neighborhood parameters can be challenging.[9]

Gaussian Mixture Models (GMM) are flexible and can model clusters of various shapes and sizes, providing probabilities for data point assignments. However, estimating GMM parameters can be computationally demanding, and the algorithm is sensitive to initialization.

Mean Shift clustering is relatively straightforward and can handle complex data structures, but it may struggle with high-dimensional data due to the "curse of dimensionality."

Spectral clustering is effective for data with non-linear structures but requires more computational resources and can be sensitive to parameter settings.

Affinity Propagation doesn't need a predefined number of clusters and can automatically identify exemplars. However, it is sensitive to initial exemplar selection and can be computationally expensive.

The choice of clustering algorithm depends on the dataset's characteristics, research goals, and the balance between accuracy and simplicity. In this research, after evaluating various algorithms, the K-means algorithm was selected for its simplicity.

3. Ant Colony

The "ant colony algorithm", or "Ant Colony Optimization" (ACO), is inspired by how colony-dwelling ants find the optimized path to transport food to the nest by analyzing ants' behavior. This technique is highly adaptable for solving optimization problems with large datasets. ACO is effective in planning travel routes and urban transit systems and is also extensively utilized in clustering and classification.[10][11]

ACO utilizes pheromones, a chemical substance emitted by ants as they walk, to mark the shortest path. The presence of more pheromone on a path increases the likelihood that other ants will follow it. Three types of ants are used in the ACO for routing traffic:[11][12]

1. The local ants keep pathways within a sector or they maintain the pathways to the nearest node in a neighbouring sector.
2. The exploration ants find and maintain paths within a sector.
3. The backward ants update routing information at each node.

The ACO algorithm works as follows:[13]

3.1 *The Status Transition Stage*

It continues until the specified criteria are met. As more ants follow a particular path, the likelihood of other ants choosing the same path increases, enabling them to move from the initial state *i* to the next state j based on a higher likelihood.[10][11]

3.2 *The Pheromone Update Stage*

It includes two rules: the updating of local and global pheromone. The update of local pheromone randomly determines the direction of the paths being formed, allowing already traveled points to be ignored by other ants. The effect of this update is to dynamically decrease interest in segments that have been used, as each use quickly diminishes the pheromone amount on that path. Consequently, other ants will indirectly choose unexplored areas. Equations (1) and (2) illustrate the changes in pheromone concentration over time.

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij} \tag{1}$$

$$\Delta\tau_{ij}(t) = \sum_{k=1}^{N} \Delta\tau_{i^k j}(t) \tag{2}$$

where,

$\tau_{ij}(t)$ is the concentration of the pheromone between point *i* and point *j*.

N is the total ants' number.

ρ is a number between 0 and 1 that represents the evaporation value of pheromone.

On the other hand, The update of global pheromone stage is designed to increase the levels of pheromone on the shorter routes taken by ants. As long as ants complete their tour at the end of each iteration, additional pheromones are deposited on the paths traversed by those ants that found the optimal routes, while other paths remain unchanged. Consequently, the shortest paths will have higher pheromone levels, aiming to achieve optimal performance. The global pheromone update is governed by equations (3), (4), and (5).[10][11][14]

$$\tau_{ij} = (1-\rho)\tau_{ij} + \sum_{k=1}^{N} \Delta\tau_{ij} \tag{3}$$

$$\Delta\tau_{ij} = \sum_{k=1}^{N} \Delta\tau_{i^k j} \tag{4}$$

$$\Delta\tau_{kij} = \begin{cases} \frac{Q}{L_k} & \text{if } k \text{ ia an ant used at a tour of } i,j \\ 0 & \text{The others} \end{cases} \tag{5}$$

where,

L is the distance elapsed by the ant k.

Q is constant that represents the pheromone level.

$\Delta\tau_{kij}$ is the pheromone amount resulted from the obtained best path.

The flowchart of basic Ant Colony algorithm is illustrated in Fig. 4:

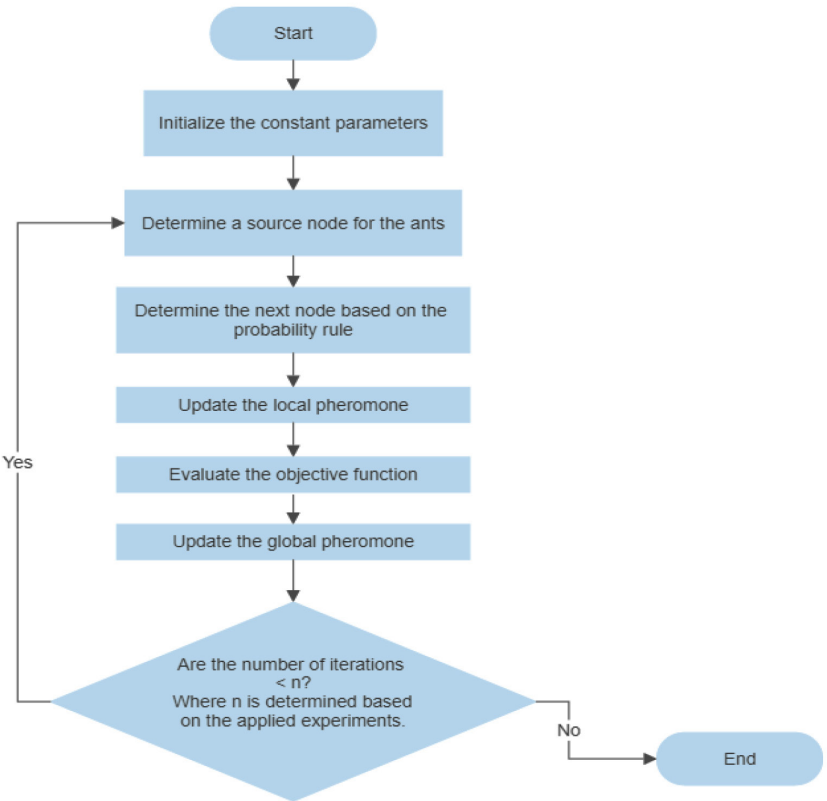


Figure 4

Ant Colony Algorithm

4. The Hybrid Integrated Mobility

Traffic routing entails finding the shortest route for a user at a specific moment, which is a challenging task. The efficiency of finding the optimal path can be influenced by various objectives and conditions, making the problem even more challenging. From a driver's perspective, minimizing route changes is crucial, while optimizing path usage and reducing average travel time are also important. However, these objectives often conflict in a dynamic environment. An algorithm like Ant Colony Optimization can estimate the probability of selecting less congested roads, offering a solution. Combined with the K-means clustering technique, it forms a hybrid system suitable for dynamic implementation. [11]

4.1 *Methodology*

The hybrid algorithm starts with implementing K-means Clustering to determine cluster centroids. These centroids are then optimized using the Ant Colony algorithm to find the shortest and optimized path within every cluster. "K-means Clustering" is reapplied to the improved previous results. Fig. 5 illustrates the block diagram of the hybrid algorithm.

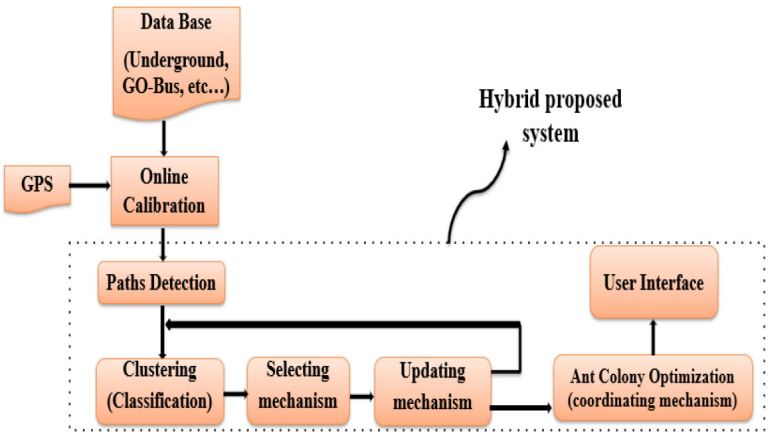


Figure 5

Hybrid System Block Diagram

The steps for implementing K-means Clustering are:

1. Generate the number of clusters (k).
2. Select random objects as initial centroids.

3. Assign objects to the nearest centroid.
4. Calculate the mean value of object members in each cluster to obtain new centroids.
5. Repeat until centroids no longer change or a maximum iteration is reached.

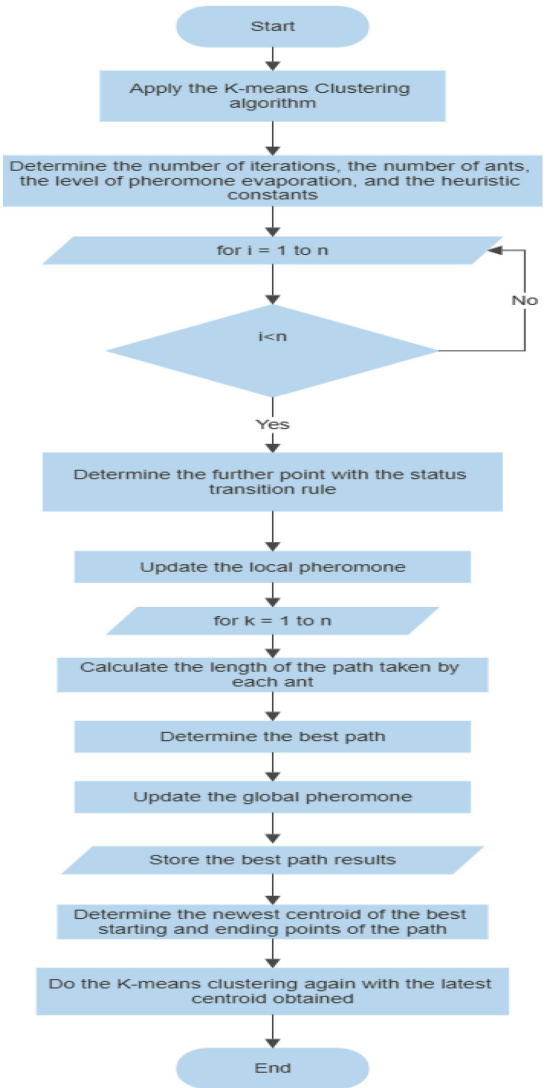


Figure 6

Hybrid Algorithm Flowchart

Optimizing the new centroids with the Ant Colony algorithm involves:

1. Determine parameters: iterations (t), ants (N), pheromone evaporation (ρ), pheromone weight (α), and path visibility weight (β).
2. Apply the ACO algorithm to each K-means cluster.
3. Save the most efficient paths for each cluster.
4. Specify new centroids based on the starting points of these paths.

A loop checks the parameter i (starting point and number of iterations) until i equals n (number of iterations). Another loop checks path lengths of each ant until parameter k equals n, ensuring previously saved data is not overwritten. The hybrid algorithm is detailed in Fig. 6.

4.2 *Software Analysis*

Examining sequence and use case diagrams helps understand the system's operation and requirements. Sequence diagrams show the order of object interactions, aiding in identifying additional requirements or supporting existing procedures. Fig. 7 illustrates the proposed solution sequence, leading to the implementation phase.[15][16]

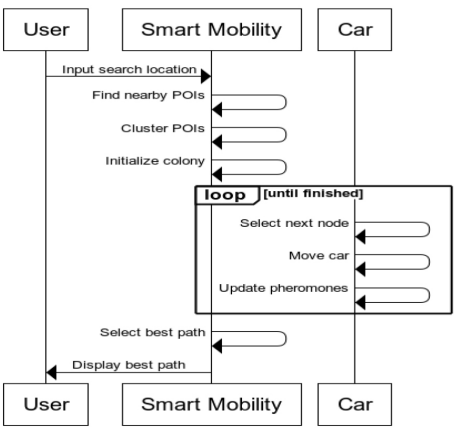


Figure 7

Sequence Diagram

A use case diagram, shown in Fig. 8, condenses information about the system and its users, often depicted as interactions between system aspects.

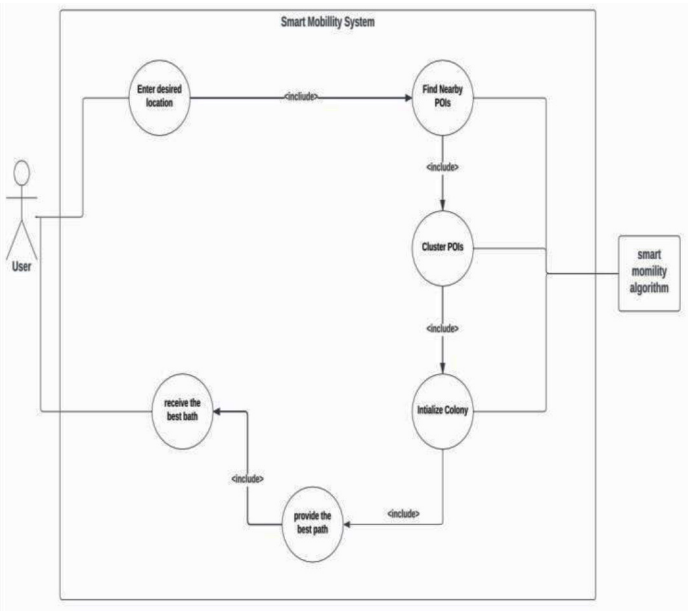


Figure 8

Use Case Diagram

5. Practical Experiment in Egypt

This section tests the K-means Clustering and Ant Colony Optimization techniques using different Egyptian locations from a dataset of 159 locations (latitude and longitude). Fig. 9 presents a sample of the dataset.

	city	city_ascii	lat	lng	country	iso2	iso3	\
0	Cairo	Cairo	30.0444	31.2358	Egypt	EG	EGY	
1	Giza	Giza	29.9870	31.2118	Egypt	EG	EGY	
2	Alexandria	Alexandria	31.1975	29.8925	Egypt	EG	EGY	
3	Shubrā al Khaymah	Shubra al Khaymah	30.1286	31.2422	Egypt	EG	EGY	
4	Al Manşūrah	Al Mansurah	31.0500	31.3833	Egypt	EG	EGY	
..	...	...	...	...	...	...	...	
153	As Sallūm	As Sallum	31.5500	25.1600	Egypt	EG	EGY	
154	Al ʻAlamayn	Al `Alamayn	30.8333	28.9500	Egypt	EG	EGY	
155	Mandīshah	Mandishah	28.3515	28.9317	Egypt	EG	EGY	
156	Qaşr al Farāfirah	Qasr al Farafirah	27.0583	27.9700	Egypt	EG	EGY	
157	Al Qaşr	Al Qasr	25.6959	28.8837	Egypt	EG	EGY	

Figure 9

Sample of the dataset

5.1 *K- Means Clustering Technique*

The dataset is initially grouped into three clusters (k = 3) and plotted on Egypt's map (Fig. 10). Due to a 98.63% error rate, the number of clusters is increased to four, reducing the error to 20.2%. The new centroids are plotted on Egypt's map in Fig. 11.

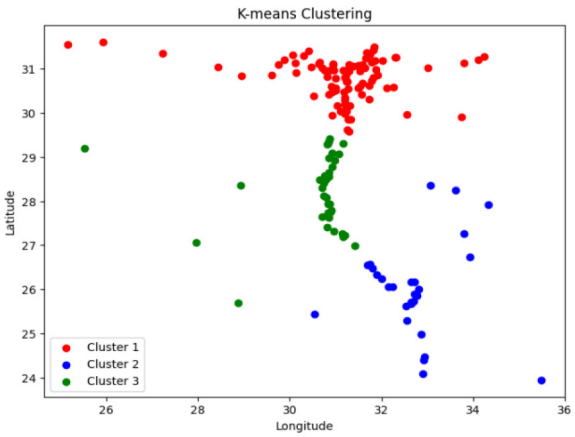


Figure 10

Plotting of the clusters (k = 3).

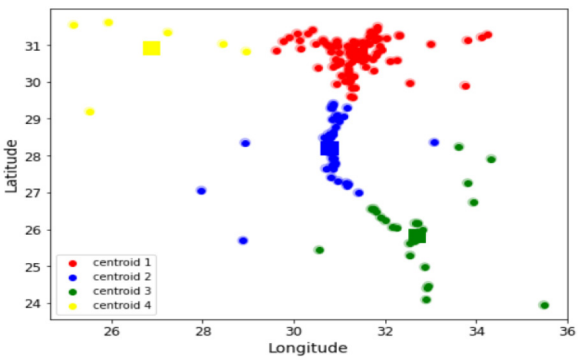


Figure 11

Plotting of the clusters (k = 4)

Results vary with the number of clusters, showing a direct relationship between increased clusters and algorithm accuracy. Experiments highlight differences in centroid and total errors for k = 3 and k = 4.

5.2 *The Hybrid Algorithm*

The algorithm's steps are:

1. Clustering the data.
2. Running the ant colony optimization to find the optimal path.
3. Ants move between cities based on distances and pheromone levels, constructing paths probabilistically.
4. Pheromone levels are updated based on paths, tracking the best path.
5. The process terminates based on conditions like time, round trips, number of ants, or a stop factor, returning the best path.

Performance is evaluated by plotting round trips vs. path cost (Fig. 12).

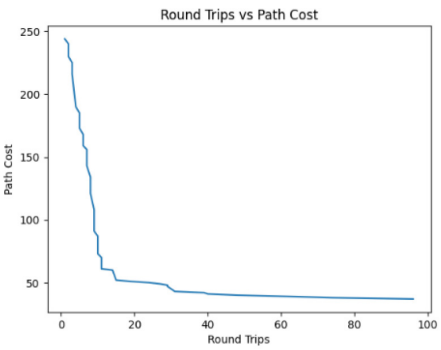


Figure 12

Round Trips vs Path Cost

Path cost represents the length of the best path, and round trips indicate complete trips through all nodes. Initially, path costs are high as ants explore, gradually decreasing as improved paths are found. Small fluctuations are normal as ants refine the solution.

5.3 *Case Study*

This research functions as a trip planner, enabling users to visit chosen locations in the shortest time and return to the starting point. For example, if a user in Alexandria wants to visit Cairo, Suez, Aswan, Sharm al Sheikh, and Ismailia, the input dataset is processed by the proposed algorithm, and the results are displayed (Fig. 13)

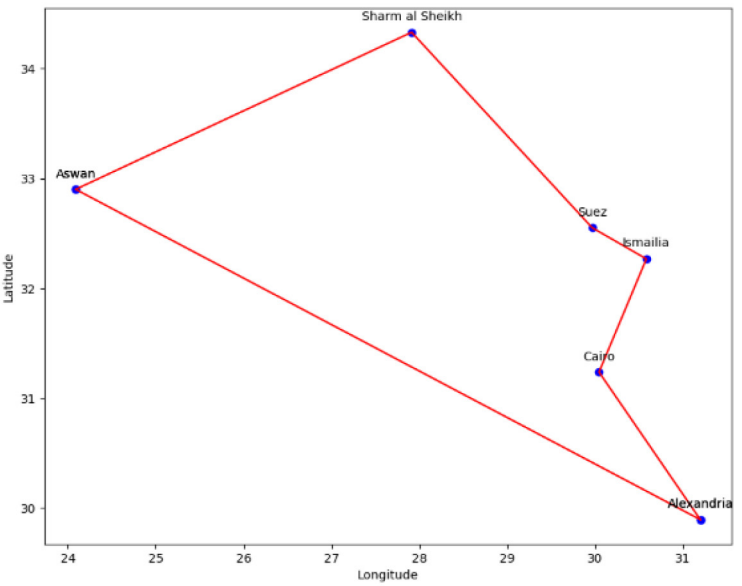


Figure 13

Case Study Results

6. Conclusion

This paper has mentioned all the stages of implementing the hybrid proposed system that was integration between both the Clustering technique using K-means, and the Ant Colony Optimization. It started by introducing the Clustering technique by implementing two dissimilar experiments by initializing two values for (k) clusters in order to improve that the error decreases while increasing the number of clusters. In addition, the Ant Colony Optimization was also mentioned with its phases, while taking the output of the clustering technique using the K-means as its input, as shown in the case study previously.

Compliance with Ethical Standards

This research followed the ethical standards and licenses. Throughout the research process, we have secured the preservation of human subjects' rights, privacy, and confidentiality. All study participants provided informed consent, and their identity was maintained in the publication of the findings. Any possible conflicts of interest have been declared and properly addressed.

Competing Interests

The authors declare that they have no competing interests that could influence how the research findings are interpreted or presented. There are no financial, personal, or professional ties that may be seen as a conflict of interest. The study was done objectively and without any outside pressures that may jeopardize its objectivity or integrity.

Research Data Policy and Data Availability:

We understand the significance of data openness and open scientific concepts. This research has analyzed the Egyptian Cities Dataset which is available online. The data repository may be found at [Egypt Cities Database | Simplemaps.com]. The data are made available in a way that assures their integrity and use by other researchers. In accordance with applicable data protection and privacy rules, we followed best practices in data handling in the presented prototype and took precautions to ensure the long-term preservation and accessibility of the study data as we are currently working on transforming this prototype model into a high-end product.

References

[1] R. K. Sharma and M. Jailia, "Machine learning and IoT-based garbage detection system for smart cities," *Journal of Information and Optimization Sciences*, vol. 44, no. 3, pp. 393–406 (2023), doi: 10.47974/JIOS-1349.

[2] S. P. D, S. S. Rao, S. Prakash, and G. S. Madhan Kuma, "SMART CITY TRAVELLER," *International Research Journal of Engineering and Technology* (2018), [Online]. Available: www.irjet.net.

[3] A. B. Osmond, S. H. Supangkat, and F. Hidayat, "Design and Implementation of Smart Trip Planner," in *2019 International Conference on ICT for Smart Society (ICISS)*, IEEE, pp. 1–4, Nov. (2019). doi: 10.1109/ICISS48059.2019.8969814.

[4] K. Iqbal, M. Adnan, S. Abbas, Z. Hasan, and A. Fatima, "Intelligent Transportation System (ITS) for Smart-Cities using Mamdani Fuzzy Inference System," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 2 (2018), doi: 10.14569/IJACSA.2018.090215.

[5] M. Z. Rodriguez *et al.*, "Clustering algorithms: A comparative approach," *PLoS One*, vol. 14, no. 1, p. e0210236, Jan. (2019), doi: 10.1371/journal.pone.0210236.

[6] G. D. ' Amico, S. Ercoli, and A. Del Bimbo, "A framework for Itinerary Personalization in Cultural Tourism of Smart Cities," (2013). [Online]. Available: http://www.micc.unifi.it/

[7] Unary Team, "'What Are The Most Effective Software Development Life Cycle Models?,' ," https://unaryteam.com/software-development-life-cycle-models/.

[8] G. Suciu, C. Butca, C. Dobre, and C. Popescu, "Smart City Mobility Simulation and Monitoring Platform," in *2017 21st International Conference on Control Systems and Computer Science (CSCS)*, IEEE, pp. 685–689, May (2017). doi: 10.1109/CSCS.2017.105.

[9] B. Atasoy *et al.*, "SMART mobility via prediction, optimization and personalization," in *Demand for Emerging Transportation Systems*, Elsevier, pp. 227–265 (2020). doi: 10.1016/B978-0-12-815018-4.00012-7.

[10] D. J. Ratnaningsih, "Clustering with K-means Hybridization Ant Colony Optimization (K-ACO)," (2022). [Online]. Available: http://ijm2c.iauctb.ac.ir.

[11] S. S. Bamane, A. J. Umbarkar, and M. R. Gaikwad, "Improved Ant Colony Optimization in K-Means for Data Clustering," pp. 521–528 (2020). doi: 10.1007/978-981-15-0630-7_52.

[12] A. Oliazadeh, O. Bozorg-Haddad, A. Arefinia, and S. Ahmad, "Ant Colony Optimization Algorithms: Introductory Steps to Understanding," pp. 137–154 (2022). doi: 10.1007/978-981-19-2519-1_7.

[13] Punam Bedi; Neha Mediratta; Silky Dhand; Ravish Sharma; Archana Singhal, "Avoiding Traffic Jam Using Ant Colony Optimization - A Novel Approach," (2008). doi: 10.1109/ICCIMA.2007.61.

[14] Z. Li and J. Huang, "How to mitigate traffic congestion based on improved ant colony algorithm: A case study of a congested old area of a metropolis," *Sustainability (Switzerland)*, vol. 11, no. 4, Feb. (2019), doi: 10.3390/su11041140.

[15] dominodatalab, "K-means Clustering in Python: A Step-by-Step Guide." Accessed: Mar. 15 (2024). [Online]. Available: www.dominodatalab.com. https://www.dominodatalab.com/blog/getting-started-with-k-means-clustering-in-python.

[16] J. MCGUIGAN, "KMeans Ant Colony Optimization," Kaggle. Accessed: Mar. 15 (2024). [Online]. Available: https://www.kaggle.com/code/jamesmcguigan/kmeans-ant-colony-optimization/notebook#Visualization.

Received October, 2023