RAIRO-Oper. Res. 60 (2026) 2043–2064
https://doi.org/10.1051/ro/2026051

RAIRO Operations Research
www.rairo-ro.org

INTEGRATING DBSCAN CLUSTERING AND DIJKSTRA'S ALGORITHM FOR ENHANCED ROUTE PLANNING: A CASE STUDY OF ORAN, ALGERIA

Seddik Reguieg[1,*], Youcef Omari[2], Miloud Dahane[3], Simon Collart-Dutilleul[4] and Mohammed Amine Mami[5]

Abstract. This paper presents a novel approach to optimizing urban transport in Oran, Algeria, by addressing the issue of fragmented connectivity through the integration of clustering techniques with Dijkstra's algorithm. The study begins with the collection and management of geo-referenced data for urban transport lines and bus stops. Using a customized DBSCAN clustering method, bus stops are grouped based on geographic and traffic data, simplifying the network for more efficient analysis. Dijkstra's algorithm is then applied to calculate the shortest paths within the clustered network, optimizing routes and reducing travel times, which improves overall system efficiency. Additionally, a web-based application has been developed for Oran's residents, with initial testing indicating promising usability for practical implementation. This integrated approach provides significant advancements in urban transport planning, offering valuable tools and insights for transport practitioners and decision-makers.

Mathematics Subject Classification. 90B06, 68R10, 68T10, 90C35.

Received December 15, 2024. Accepted May 8, 2026.

1. Introduction

Tourism has gained increasing importance in recent years with the rise of sporting and cultural events in Oran, Algeria. This growth has led to a rising demand for more sophisticated and efficient urban transport means, particularly for bus and tramway networks, as BACHIRI and TRIKI [3] note that the current transport infrastructure in Oran creaks under the pressure of increased usage. Addressing this demand requires focused attention from authorities to effectively manage the transportation system, underscoring the need for advanced routing optimization techniques. Implementing such methods can enhance the efficiency and reliability of these systems, ultimately improving the overall travel experience for both residents and visitors. Technology plays an essential role in achieving these objectives. A study by Levinson *et al.* [21] highlights how Transportation

Keywords. Urban transport, route planning, optimization, clustering, path-finding.

[1] University of Oran 2 Mohamed Ben Ahmed, Department of Earth Sciences, 31000 Oran, Algeria.
[2] National School of Artificial Intelligence (ENSIA), Sidi Abdellah Campus, Algiers, Algeria.
[3] University of Oran 1 Ahmed Ben Bella, Department of Biology, 31000 Oran, Algeria.
[4] Gustave Eiffel University, COSYS-ESTAS, F-59650 Villeneuve d'Ascq, France.
[5] University of Oran 1 Ahmed Ben Bella, Department of Computer Sciences, 31000 Oran, Algeria.
*Corresponding author: `reguieg.seddik@univ-oran2.dz`

Network Companies have reshaped urban mobility, demonstrating the substantial potential of technology in advancing urban transport.

Previous work on urban route planning, particularly the integration of Dijkstra's algorithm with Density-Based Spatial Clustering of Applications with Noisex (DBSCAN) for traffic management by Zhang *et al.* [30], serves as a foundation for this study's approach in Oran. Their findings show how combining Dijkstra's path-finding with DBSCAN clustering can improve traffic flow and safety by managing incidents on large road networks, they focused on improving road traffic management by clustering incidents and rerouting around them, ideal for real-time adjustments, while our contribution adapts this method to optimize urban public transit in Oran, Algeria, clustering bus stops and using Dijkstra's algorithm to improve route efficiency.

This work primarily focuses on an applied decision-support perspective, aiming at demonstrating how clustering and graph-based optimization techniques can be effectively deployed in a real-world urban transport context.

Before outlining the contributions of this work, it is crucial to address the following research questions:

- *What are the primary limitations of Oran's current transport network?*
- *What challenges do users encounter when planning their routes within Oran's urban transport system?*
- *How can we effectively address the issue of unlinked transport networks by identifying transfer areas and achieving a fully connected network?*

This study leverages advanced algorithms to address the specific urban mobility challenges of Oran. Its primary objective is to develop a robust route optimization framework that significantly improves public transport efficiency and enhances the travel experience for both residents and visitors. The key contributions of this work are as follows:

(1) *Extensive analysis, comprehensive data collection, and detailed modeling of Oran's transport network.*
(2) *A novel approach to resolve unlinked cycle movements in transport networks by integrating DBSCAN clustering and Dijkstra's algorithm.*
(3) *A route planning framework that seamlessly combines logistical efficiency with a user-centric design.*

The document is structured as follows: Section 2 provides an extensive theoretical and empirical background. Section 3 outlines the methodologies employed in this research, including Subsection 3.1, which describes the Oran transport network, its unique specifications, and challenges; Subsection 3.2, which introduces the overall approach; Subsection 3.3, which covers data collection and processing; and Subsection 3.4, which details network processing and modeling, including the initial transport graph and the techniques used. Section 4 presents the key findings, starting with the obtained graph, followed by an illustrative path-finding scenario and a performance evaluation of the utilized techniques. Section 5 analyzes the outcomes, addresses persisting issues, and proposes solutions, while Section 6 summarizes the results and suggests directions for future research.

2. Background

This section provides the theoretical and empirical background for the current study, organized into three subsections: modeling of public transport networks, a review of key literature on route planning, and studies on the integration of DBSCAN and Dijkstra's algorithm.

2.1. Modeling of public transport networks

This section offers a comprehensive overview of the various modeling approaches used in the management of public transport networks, highlighting their complexities and the necessity of developing appropriate models to enhance customer satisfaction and operational efficiency.

By providing urban transport services that consider operational constraints such as adherence to theoretical timetables, connection guarantees, and minimized waiting times, these models aim to enhance the overall efficiency of transport systems [8].

Modeling begins with gathering comprehensive data on passenger flow, stops, network lines, vehicles, and the distances between stops. This data underpins the graphical representations of transportation networks, which are essential for visualizing vehicle movements and aiding decision-making in the event of disruptions. The graphical model varies across studies but serves as a fundamental tool for understanding the behavior of network components and facilitating effective management [31].

One popular method involves the use of **Petri nets**, which have been applied in numerous studies to depict various aspects of transport networks, such as the operations at a bus stop illustrated in the accompanying figure.

Another approach is the **Multi-Agent System (MAS)**, which focuses on the behavior of networks. Agents within these systems interact in a shared environment to address complex issues that surpass individual capabilities, proving particularly useful in urban traffic regulation and management [18].

The use of **Unified Modeling Language (UML)** also contributes to the modeling process by providing tools that simplify real-world complexities, which is especially beneficial for managing and understanding the intricate operations of tram networks before their implementation [2].

Similarly, another method involves creating a **System of Equations** to model bus traffic within urban networks, taking into account passenger distribution at stops. This approach not only helps in predicting line coverage and passenger flow but also assists in estimating revenue and managing operational costs, thereby aiding in profitability assessments [19].

Graph theory remains a dominant tool in transport modeling. It describes stops as nodes and movements as oriented, weighted arcs, where weights represent distances between stations. This method is exemplified by a network diagram featuring nodes and arcs that connect various points, illustrating the comprehensive path routes within the network [5]. The arcs' weights correspond to the distance between the two successive stations symbolized by the vertices. Thus, each line is represented by a series of nodes and arcs which connect the departure terminus to the arrival terminus [25].

Race modeling, **flow** modeling, and **route** modeling are specialized approaches that cater to specific aspects of transport network management. Race modeling addresses vehicle scheduling by assigning each vehicle to a specific route, critical for solving Multi-Depot Vehicle Scheduling Problems. Flow modeling allocates flow rates to routes based on costs and travel times, useful in managing congestion and planning static traffic in multimodal networks. Route modeling, the most comprehensive approach, involves detailing potential routes through sequences of nodes and arcs, optimizing the demand and placement of transfer points across the network [12].

As these models collectively advance our understanding of urban transport phenomena [28], they lay the groundwork for the next crucial phase: route planning. The subsequent section will explore how these theoretical models are applied practically, focusing on optimizing transit routes to improve public transport efficiency and user satisfaction in Oran. This not only bridges the gap between theory and practice but also highlights the direct benefits of these models in real-world applications, paving the way for innovative developments in urban transit systems.

2.2. Survey of route planning works

Route planning involves systematically finding an optimal route from a starting point to a destination. It has long been a critical component of navigation systems, primarily implemented through graph theory algorithms and various optimization approaches. Key algorithms for determining the shortest or fastest paths across complex networks include Dijkstra's algorithm, A*, and other heuristic search methods that dynamically integrate real-time data, such as traffic conditions and roadblocks [24].

Modern route planning tools, like Google Maps, not only consider current traffic events but also temporary roadblocks, providing users with the best routes and alternatives, complete with turn-by-turn directions, estimated arrival times, and travel contingencies [6]. These features are essential in urban logistics and transportation management, guiding users effectively to their destinations.

Service quality attributes significantly impact customer satisfaction in bus transit systems. According to Eboli and Mazzulla [10], factors such as reliability, comfort, safety, and convenience influence passenger satisfaction, with punctuality and service frequency being the most critical attributes. Public transport authorities should prioritize these elements to enhance customer experience and encourage greater use of bus services.

Modern route planning systems improve city mobility by optimizing travel itineraries based on time, distance, or cost [20]. However, focusing solely on these factors can lead to routes that are unsafe or environmentally harmful. Thus, integrating safety and environmental conditions into user profiles is essential for better route recommendations [4].

These systems also facilitate seamless transit connections through multi-modal integration, resulting in high user satisfaction with efficient, comfortable travel experiences across different transport modes [22]. Real-time transit status information, such as delays and timetable changes, further enhances reliability and efficiency, crucial for managing unpredictability in urban travel [8].

By promoting public transport through comprehensive planning tools, these systems enhance safety, convenience, and sustainability, which are vital for improving public perception and acceptance of public transit as a preferred mode of urban mobility.

In summary, detailed, timely, and accurate travel information empowers users to select the best routes to meet their needs while improving overall travel efficiency and satisfaction. This holistic approach underscores the need for adaptive, user-centric planning systems that address contemporary mobility demands and urban transit challenges [9].

2.3. Use of DBSCAN and Dijkstra algorithms in traffic management and route optimization

The integration of clustering methods, such as DBSCAN, with path-finding algorithms like Dijkstra's has been demonstrating its potential in various areas, particularly in traffic and route optimization. This hybrid approach addresses the limitations of both methods, providing a sophisticated solution to complex logistical challenges.

One of the pioneering works in this area is by Zhang *et al.* [30], who named their project "Dijkstra's-DBSCAN." They implemented this method for efficient traffic incident management across extensive road networks. Their approach significantly enhances traditional route planning systems by optimizing path-finding, estimating and controlling traffic densities, and managing recurrent congestion. This method is particularly applicable in urban settings, where traffic movement presents numerous challenges and patterns are often complex and highly variable, encompassing both emergencies and routine flows.

In 2020, Kiliç [17] introduced a tool for planning public transportation routes called ToTRoP. This tool employs a modified Dijkstra algorithm to identify optimal routes based on travel time, distance, and transfers. By applying DBSCAN clustering to eliminate less dense points, the combination significantly enhances route accuracy and efficiency, thereby improving public transport planning and user experience.

Subsequently, Huang *et al.* [13] utilized the DBSCAN clustering method in conjunction with Dijkstra's algorithm to address the challenge of identifying passenger pickup hotspots for taxis. By clustering areas of high passenger activity and routing these hotspots through Dijkstra's algorithm to determine the shortest paths for taxis, their method effectively enhanced service response and operational efficiency, which is crucial in busy downtown areas.

In 2022, Magnana, Rivano, and Chiabaut applied this integrated approach to model bicycle route preferences using GPS data [23]. Their research demonstrates how clustering common routes and employing path-finding algorithms can provide valuable insights for urban planners to improve bicycle infrastructure, thereby enhancing user experience in daily mobility.

The most recent study, conducted by Wan, Pu, Schonfeld, and their team in 2023, focused on optimizing railway alignments in areas crowded by physical obstacles [27]. This application identifies and clusters obstacles using the DBSCAN algorithm, subsequently finding possible railway paths around these obstacles with Dijkstra's algorithm.

Together, these studies highlight the power and versatility of integrating DBSCAN with Dijkstra's algorithm. Their findings underscore the significance of such integrated approaches across various domains, including traffic management, environmental planning, and urban infrastructure development, demonstrating their practical applications and relevance in current research.

From an operations research perspective, alternative optimization approaches such as Mixed Integer Programming (MIP) have been widely used to address transportation network design and hub location problems [7, 15]. These formulations are particularly powerful when detailed cost structures, demand patterns, and temporal constraints are available. However, in data-scarce and highly fragmented urban transport contexts such as Oran, their effectiveness becomes limited, as overly complex models may introduce unnecessary modeling overhead without providing proportional practical benefits. In such settings, graph-based and clustering-oriented approaches offer a more flexible and pragmatic solution, enabling effective connectivity enhancement and route planning while remaining aligned with available data and operational constraints.

2.4. Route planning within Oran transport network

To the best of our knowledge, no prior work has addressed the route planning problem within the city of Oran. Additionally, no public or private web platforms, such as Google Maps, currently offer such services for this region. This makes our work the first of its kind in the Algerian context.

3. Methodology

This section presents the comprehensive methodology employed in this study to optimize urban transport in Oran, Algeria. The methodology utilizes advanced computational algorithms to create a user-centric transport application grounded in a practical system design framework. This holistic approach ensures that the proposed solution is both theoretically sound and practically applicable, contributing to efforts to enhance urban mobility.

3.1. Urban transport network of Oran

As discussed in previous sections, the primary motivation for this case study is to optimize route planning within a multi-modal road network (Buses and Tram). Achieving this goal hinges on accurately modeling the network itself. The following sections provide an in-depth examination of the graph modeling approach and subsequent analysis applied to Oran's urban transportation network.

3.1.1. Challenges and specifications

Transport networks differ based on the unique characteristics of the areas they serve, and these specifics greatly influence the design and efficiency of network modeling. Oran's transport network, like any other, is structured around key components such as modes of transport (buses and trams), lines, and stops. Within this system, the tram operates on its own dedicated line but interconnects with various bus lines and stops.

Below are the key features of Oran's transport network:

- Independent Bus Operators: Each bus line is managed by different operators without a centralized authority coordinating all routes.
- Fare Structure: Passengers pay for each bus trip individually, and transferring between buses requires additional fare payments.
- Route Variability: Outbound and inbound routes differ, although they both connect a primary departure station to an arrival station.
- Lack of GPS Data: The model excludes travel times, schedules, and real-time tracking information, as the buses are not equipped with GPS technology.
- Distinct Stop Coordinates: Each bus stop is uniquely geolocated for its respective route, with no designated transfer hubs between different lines. Moreover, shared stops between lines often have distinct coordinates, varying by tens to hundreds of meters.

The absence of centralized coordination leads to inefficiencies, such as overlapping routes, inconsistent service quality, and difficulties in optimizing the network as a whole. This lack of integration inconveniences passengers

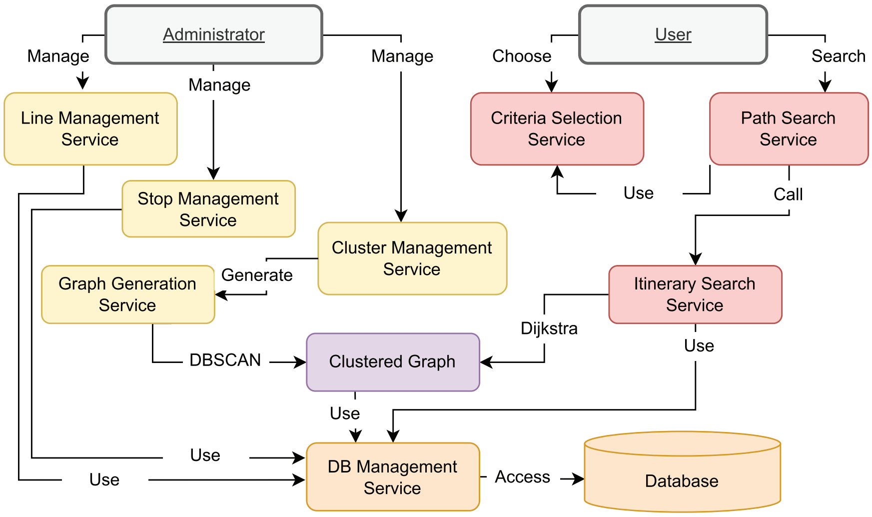


FIGURE 1. High-level functional overview of our approach.

and hinders effective urban transit planning. Furthermore, the need to pay separate fares for each bus trip discourages multi-bus journeys, increasing travel costs for passengers. It also complicates the adoption of unified ticketing systems, reducing the overall affordability and attractiveness of public transport.

Additionally, differing outbound and inbound routes can confuse passengers, particularly those unfamiliar with the system. This inconsistency makes route planning and navigation more challenging for both users and system optimization efforts. Another significant limitation is the lack of GPS technology, which prevents the system from providing real-time tracking, estimated arrival times, or schedule adherence. This restricts passengers' ability to plan their trips efficiently and reduces opportunities for data-driven improvements to the network.

Finally, the absence of designated transfer hubs results in inconvenient and time-consuming transfers for passengers. Moreover, the lack of centralized hubs hinders the optimization of intermodal connectivity within the network.

3.2. Overview of the study approach

The methodology adopted in this study is a systematic review, emphasizing the importance of each detail in data collection, data processing, and algorithm application for developing an optimized transport model for the city of Oran. This methodology is inherently modular, allowing for independent updates to any component as new data or techniques become available. Figure 1 presents a high-level functional architecture of the proposed system. The diagram is inspired by UML use-case and service-oriented modeling principles. Its purpose is to illustrate the interactions between users, services, data management components, and optimization algorithms (DBSCAN and Dijkstra), rather than to provide a formal software design specification. Two distinct user profiles are identified:

The Administrator Profile is tasked with managing the transport network through various services. These include overseeing different transport means, lines, and stops that constitute the initial transport network. Additionally, the Administrator has the capability to refine the network to ensure its proper utilization by end users. This refinement process is achieved using the Clustering Service, which employs the DBSCAN algorithm for network optimization.

The Administrator manages two critical services: the **Line Management Service** and the **Stop Management Service**. These services ensure that information about transport lines and stops is accurate and

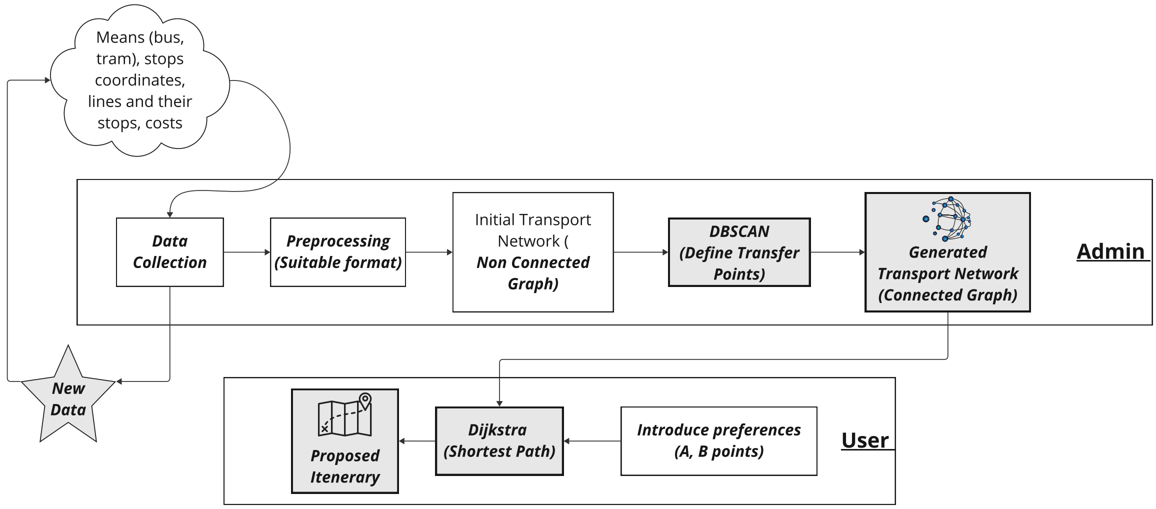


FIGURE 2. General flowchart of our approach.

regularly updated. **The Cluster Management Service** utilizes the DBSCAN algorithm to transform the urban transit network into a cohesive and connected graph, enabling efficient route optimization. **The Graph Generation Service** works in tandem with the Clustering Service to generate a connected graph that serves as the foundation for the Dijkstra Algorithm. The resulting graph facilitates the computation of shortest paths for route planning.

The User Profile empowers users to define search criteria and select departure and arrival points. User interaction is facilitated by the **Criteria Selection Service**, which connects to the **Itinerary Search Service**. This service leverages the Dijkstra Algorithm to calculate the shortest path within the connected graph generated by the Graph Generation Service. Additionally, **the Database Management Service** (DB Management Service) ensures efficient querying and reinforces a user-centric approach to urban mobility.

Figure 2 provides a detailed flowchart of our approach, illustrating the interconnected steps. The process consists of two main parts: *Transport Graph Construction* and *Itinerary Search*.

To build a usable and optimized graph, data about transport means, stop coordinates, lines, and costs must be collected continuously to maintain an up-to-date network. The process begins with a data preprocessing step, which cleans and formats the collected data. This step results in an initial graph containing many non-connected cycles, which create inconsistencies and ambiguity, rendering route planning ineffective.

The DBSCAN clustering algorithm addresses these challenges by identifying clusters and defining transfer points that connect previously disconnected areas. This process transforms the initial graph into a fully connected graph suitable for effective route planning. Once the connected graph is generated, users can search for the shortest path by specifying their starting and destination coordinates. The Dijkstra Algorithm calculates the most efficient route within the graph.

The connected graph is generated only once and can be reused multiple times unless significant new data necessitates an update. When new information is added, the graph construction process is reinitiated to update and refine the transport network.

3.3. Data collection and pre-processing

The data collection and preprocessing stage is critical in route planning, forming the foundation for the entire methodology. In this study, data was collected to encompass the entire network of bus and tram lines, including

TABLE 1. Sample of Oran bus lines.

Line Id	Line name	Departure	Arrival
5	C	578	552
9	51	88	119
17	P1	197	214
22	4G	251	795
28	G3	526	551
36	Tramway	580	645

TABLE 2. Sample of Oran bus stops.

Stop Id (name)	Next stop (name)	Line Id	Stop position
552 (Sports-Palace)	554 (Chakouri)	5	(35.6896, −0.646526)
104 (Gambetta)	106 (Kavgui)	9	(35.7068, −0.620619)
197 (Lotfi)	199 (Hamou)	17	(35.7051, −0.630447)
251 (Yaghmoracen)	253 (La Glaciére)	22	(35.6692, −0.658906)
528 (Museum)	530 (Place Roux)	28	(35.6958, −0.645576)
581 (Hai Sabah)	582 (Hai yasmine)	36	(35.6966, −0.571863)

their routes, timetables, and geolocations (longitude and latitude) for each stop. This comprehensive dataset captures the current state of the urban transport network and highlights areas requiring improvement.

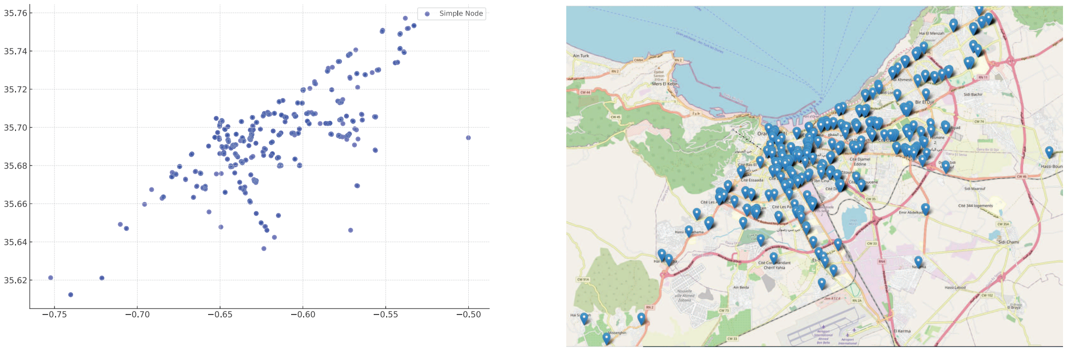
The data collection process posed significant challenges, particularly due to time constraints. The network comprises 25 bus lines and one tram line, with a total of 592 stops. Data was collected manually, geolocating all stops for each transport mode, including the tram line. Buses were identified using alphanumeric codes of up to three characters derived from their line names. Similarly, stop names were manually standardized to enhance usability, particularly for non-local users such as tourists and visitors. This meticulous approach ensures the dataset accurately represents real-world conditions, improving both convenience and usability for end-users.

Following data collection, extensive preprocessing was performed to ensure data quality and consistency. This included cleaning missing or irrelevant values, such as outdated stop geolocations or names, guided by domain experts. Additionally, buses were linked to their respective lines, and stops were assigned to each line. The data was then stored in a structured tabular format for subsequent graph-based modeling. These steps ensure a consistent and accurate representation of Oran's transport network.

Tables 1 and 2 provide samples of the collected and preprocessed data. Figure 3a presents a scatter plot of the 592 geolocated bus stops, offering visual insights into their spatial distribution. The plot reveals densely served areas and sparsely covered regions, highlighting disparities in transit accessibility.

In densely clustered areas, stops are so closely packed that distinguishing them visually becomes difficult. This density makes it impractical to manually define transfer points, as overlapping and proximate stops hinder the identification of optimal transfer areas. Neighboring stops often have minimal coordinate differences, increasing the complexity of manual assignments and risking redundancies. Without computational assistance, such manual definitions may lead to inaccuracies and inefficiencies in transfer hub placement, ultimately reducing the overall effectiveness of the network.

This analysis underscores the importance of automated approaches to handle the complexities of dense transit networks, ensuring accurate and efficient route planning.



(A) Stops represented by Geolocated points

(B) Map of Urban Bus Stops in Oran

FIGURE 3. Stops distribution.

3.4. Graph modeling and analysis

Graph modeling involves representing a network as a graph where nodes are connected by weighted edges. In this framework, nodes signify bus or tram stops, while edges link adjacent stops along the same bus or tram route. These edges are defined by key parameters, including the distance between stops and the frequency of service across the Oran transport network. Notably, the travel time parameter is not considered in this model, as the current infrastructure lacks data to accurately track such information.

This graph-based model for Oran's transportation network is designed to reflect the city's layout and distinct operational factors, outlined as follows:

Definition 1. We define the `transport network model` I as follows:

$$I = (G, R)$$

where:

- $G = (S, C)$ is a *directed graph* representing the bus stop network:
  - The *vertex set* S denotes the set of bus stops.
  - The *arc set* $C \subseteq S^2$ represents the connections between stops.
- R denotes the *set of routes* within the network.

Definition 2. For each route $x \in R$:

- S_x is the set of bus stops along route x.
- $C_x \subseteq S_x^2$ is the set of connections specific to route x.

These sets satisfy the following conditions:

$$S = \bigcup_{x \in R} S_x \quad \text{and} \quad C = \bigcup_{x \in R} C_x.$$

Definition 3. A *route* $x \in R$ is defined by an *acyclic path* through graph G, denoted by:

$$x = \langle S_1, S_2, \ldots, S_k \rangle,$$

where $(S_i, S_{i+1}) \in C_x$ for all $i \in \{1, \ldots, k-1\}$. This indicates that:

- Each route x begins at an initial stop S_1,

– Proceeds sequentially through intermediate stops $S_2, \ldots, S_{k-1}$,
– And ends at stop S_k.

In this model, a set of buses on route x will initiate from S_1, proceed through the defined sequence of stops, and terminate at S_k. Our data is represented in two main forms: *lines* and *stops*.

Definition 4. Each *line* L_i in the urban network is characterized by the following elements:

$$L_i = (Li\_id,\ Li\_name,\ Li\_dep,\ Li\_arr)$$

where:

– *Li_id*: A unique identifier for the urban line.
– *Li_name*: The name of the urban line.
– *Li_dep*: The departure bus stop for the line.
– *Li_arr*: The arrival bus stop for the line.

Definition 5. Each *bus stop* S_i is represented as a set of attributes, given by:

$$S_i = (Si\_id,\ Si\_name,\ Si\_next,\ Si\_Li\_id,\ Si\_type,\ Si\_position)$$

where:

– *Si_id*: A unique identifier for the bus stop.
– *Si_name*: The name of the bus stop.
– *Si_next*: The next bus stop on the line.
– *Si_Li_id*: The identifier of the line to which the bus stop belongs.
– *Si_type*: Terminus, Inbound or Outbound stop.
– *Si_position*: The geographic position of the bus stop, given in longitude and latitude.

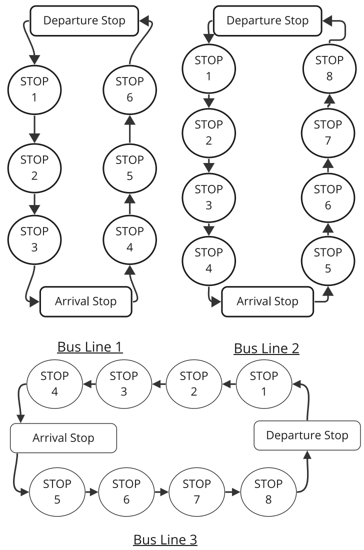
The transportation network of Oran is modeled as a graph comprising 592 nodes (bus stops) and 592 edges (connections), represented as 26 **cyclic subgraphs** corresponding to individual bus lines. Each bus line forms a cycle of nodes, initiating and terminating at terminal stops, reflecting the bidirectional nature of the routes (see Fig. 4a). This cyclical structure arises from the specific arrangement of stops, ensuring comprehensive coverage of service areas. However, the lack of connectivity between these cycles results in **isolated** subgraphs (see Fig. 4b).

Analysis of these representations reveals significant fragmentation within the network, where cycles of individual bus lines remain isolated. This **structural disconnect** prevents seamless transfers between lines, a critical feature for an integrated urban transport network. For passengers, the **poor connectivity** means journeys often require intermediary transfer points, significantly increasing travel time and reducing network efficiency.

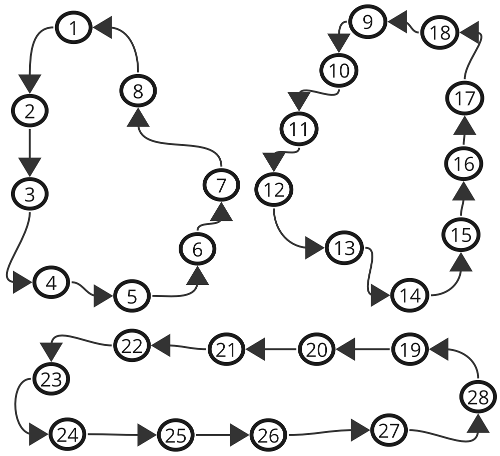
Connectivity is a fundamental property of transport networks, enabling the existence of at least one path between any two nodes [29]. In the case of Oran's network, the fragmentation not only disrupts passenger travel but also limits the applicability of key graph-theoretic metrics, such as the diameter. To quantify the level of connectivity, we used the density value D_v (see Formula 1), where higher values indicate stronger connectivity and lower values reflect greater fragmentation. The calculated density value of $D_v = 0.0017$ highlights the **sparse connectivity** of the initial graph, underscoring the need for structural improvement.

$$D_v = \frac{C}{S(S-1)}. \tag{1}$$

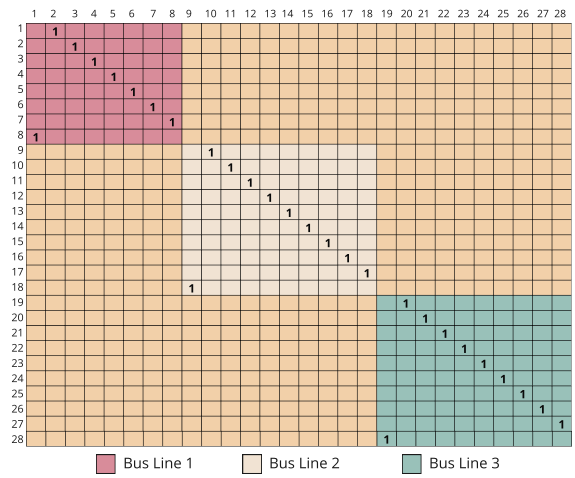
Connectivity between locations is represented in an adjacency matrix M (see Fig. 4c), where each entry $M(S_i, S_j)$ indicates the strength of the connection between nodes S_i and S_j. For nodes not directly linked, $M(S_i, S_j)$ is assigned a value of 0.



(A) Sample of three bus lines cycles



(B) Graph representation of the bus lines sample

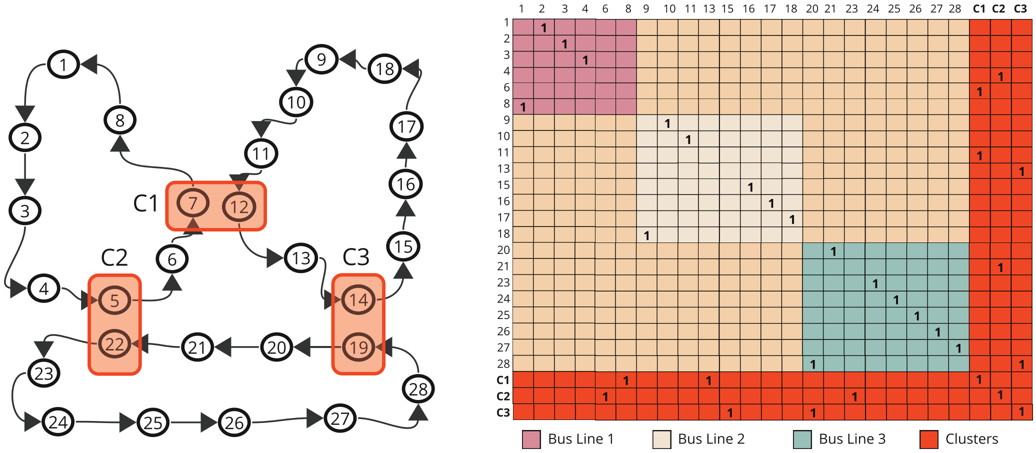


(C) Adjacency matrix of three bus lines

FIGURE 4. Non-connected samples of line cycles and their corresponding adjacency matrix.

The lack of interconnections between the graph's cycles presents challenges for traditional **shortest path algorithms**, such as Dijkstra's, which depend on **connected nodes** to explore possible routes. In this **disjointed** configuration, these algorithms cannot compute viable paths across isolated components, making **comprehensive route planning impossible**. This limitation highlights the necessity of introducing mechanisms to **connect the cycles** or rethinking the network structure to support integrated analysis.

To address this issue, we applied the Density-Based Spatial Clustering of Applications with Noise **(DBSCAN)** algorithm. DBSCAN effectively identifies high-density regions in spatial data, such as critical nodes where multiple bus line cycles converge. By clustering these independent cycles and designating **key**



(A) New generated graph with clusters as nodes

(B) The adjacency matrix of the new connected graph

FIGURE 5. Connected graph representation.

transfer stops, DBSCAN enables the creation of a cohesive network. These transfer stops act as strategic nodes, **bridging** the previously disconnected components and facilitating integrated route optimization. This enhanced connectivity transforms Oran's fragmented transport network into a unified system, improving accessibility and usability for passengers while enabling efficient path-finding methods to operate effectively.

3.4.1. DBSCAN: Clusterization of Oran transport graph

The utility of DBSCAN extends far beyond basic clustering; it fundamentally redefines the topology of Oran's transport network. By forming clusters that function as strategic hubs, DBSCAN transforms specific stops into vital connection points between previously disjointed cycles. These hubs significantly enhance the effectiveness of the route planning system, ensuring comprehensive coverage and improved navigability for commuters. This innovative application of DBSCAN contributes to the development of a more unified and user-friendly urban transport network.

To illustrate its impact on a non-connected graph, we utilized the example shown in Figure 4b. After applying DBSCAN, the graph was transformed into a connected network with three clusters – $C1$, $C2$ and $C3$ (transfer points) – as depicted in Figure 5a. The corresponding adjacency matrix, presented in Figure 5b, demonstrates how these clusters were added as new stops, serving as binding points between the previously disconnected cycles.

3.4.2. Why DBSCAN?

Several clustering methods, such as K-Means, Agglomerative Clustering, and Gaussian Mixture Models, were considered, but DBSCAN was chosen for its suitability for spatial data [1]. Unlike K-Means, which minimizes variance and requires a predefined number of clusters, DBSCAN identifies clusters based on density using two parameters: ϵ (maximum walking distance) and MinPts (minimum of two stops). This makes DBSCAN ideal for grouping irregularly distributed stops into meaningful clusters, such as transfer hubs while ignoring noise. Its ability to handle the spatial patterns of Oran's transport network ensures improved connectivity and practical integration of disconnected subgraphs.

3.4.3. DBSCAN methodology

The DBSCAN clustering method [11, 16] is one of the most widely utilized techniques among density-based classification methods. It enhances graph connectivity by grouping data points based on a distance metric and a minimum number of neighboring points. The algorithm requires only two input parameters:

– **EPS:** Defines the neighborhood radius around a data point. Two points are considered neighbors if the distance between them is less than or equal to EPS.
– **MinPts:** Specifies the minimum number of points required to form a dense neighborhood. The size of the dataset is proportional to the choice of MinPts.

In this study, we define the neighborhood radius (EPS) and the minimum number of points (MinPts) to suit the specific requirements of our problem: EPS reflects the maximum walking distance for passengers, and MinPts is set to a minimum of two stops. The general principle of density-based clustering is that each point in a cluster must have a neighborhood (within the EPS radius) containing at least MinPts points. DBSCAN iteratively collects density-reachable points starting from core objects, with the process terminating when no additional points can be added to a cluster.

The DBSCAN algorithm operates as follows:

(1) **Initialization**: Mark all points as unvisited.
(2) **Visit Points**: Select an unvisited point p, mark it as visited, and find all points in its ϵ-neighborhood.
(3) **Expand Cluster**: If the number of points in the ϵ-neighborhood is greater than or equal to MinPts, create a new cluster and add p and its neighbors to it.
(4) **Iterate**: For each point q in the cluster, if q is unvisited, mark it as visited and repeat the neighborhood query. If q is a core point, add its neighbors to the cluster.
(5) **Repeat**: Continue until all points are visited. Points not assigned to any cluster are classified as noise.

To adapt DBSCAN to our transport network use case, several modifications were made:

(1) **Shared line points**: The transfer spaces created by clusters should not include stops belonging to the same bus line. In standard DBSCAN, stops and their corresponding backstops on the same line might cluster if they are close in proximity. In our case, these stops are treated as distinct entities to preserve the clarity and functionality of transfer spaces.
(2) **Noise nodes**: The concept of noise is not applicable in this context, as our focus is not on clustering all points but rather on identifying transfer hubs. Points that are not part of a cluster are treated as regular stops rather than noise.

Algorithm 1 provides a detailed outline of the customized DBSCAN algorithm, incorporating these modifications to suit the requirements of our study.

We proceed with the following steps:

(1) **Condition for the Creation of Clusters:** We introduce a constraint to prevent stops on the same line from forming a cluster. This condition is mathematically expressed as:

$$\mathrm{CL} \in L \implies \exists n_1, n_2 \in \mathrm{CL} \text{ such that } n_1 \in L_i \text{ and } n_2 \in L_j \text{ with } i \neq j. \tag{2}$$

(2) **Regular Points Generation:** Noise points will be classified as regular points, as illustrated in the following formula 3:

$$\forall n \in \hat{N}, \implies n \in N. \tag{3}$$

ϵ-Neighborhood: The ϵ-neighborhood of a point p is the set of points q such that the distance between p and q is less than or equal to ϵ:

$$N_\epsilon(p) = \{q \in D \mid \text{Haversine}(p, q) \leq \epsilon\}$$

where D is the set of all points in the dataset.

Algorithm 1. Customized DBSCAN clustering algorithm.

```
procedure DBSCAN(D, ϵ, MinPts)
    C ← 0
    for each unvisited point p in dataset D do
        mark p as visited
        N ← RegionQuery(p, ϵ)
        if the number of points in N is ≥ MinPts then
            if HasDifferentLines(N) then
                C ← C + 1
                ExpandCluster(p, N, C, ϵ, MinPts)
            end if
        end if
    end for
end procedure
procedure ExpandCluster(p, N, C, ϵ, MinPts)
    add p to cluster C
    for each point q in N do
        if q is not visited then
            mark q as visited
            N̂ ← RegionQuery(q, ϵ)
            if the number of points in N̂ is ≥ MinPts then
                N ← N ∪ N̂
            end if
        end if
        if q is not yet member of any cluster then
            add q to cluster C
        end if
    end for
end procedure
function RegionQuery(p, ϵ)
    return all points within p's ϵ-neighborhood (including p)
end function
function HasDifferentLines(N)
    let lines ← {L(p) | p ∈ N}
    return size(lines) > 1
end function
```

Haversine distance: The Haversine distance d between two points with coordinates (ϕ_1, λ_1) and (ϕ_2, λ_2) is calculated as follows (Formula 4):

$$d = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos(\varphi_1)\cos(\varphi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right). \tag{4}$$

Where:

- d is the distance between two points,
- r is the radius of the Earth,
- φ_1 and φ_2 are the latitudes of the two points in radians,
- λ_1 and λ_2 are the longitudes of the two points in radians.

Typically, the DBSCAN algorithm relies on the Haversine formula to measure distances between coordinates specified by latitude and longitude. This approach calculates the shortest path between two points on the Earth's surface, treating the Earth as a sphere, which provides a reasonable approximation for geographic clustering. By applying the Haversine formula, each edge weight in the graph represents the geographic

TABLE 3. Complexity comparison of DBSCAN clustering algorithms.

Algorithm	Operation	Complexity
Standard DBSCAN	RegionQuery	$O(n)$
	ExpandCluster	$O(n \cdot m)$
	Total	$O(n^2)$
Customized DBSCAN	RegionQuery	$O(n)$
	HasDifferentLines	$O(m)$
	ExpandCluster	$O(n \cdot m)$
	Total	$O(n^2)$

distance between two points, enabling an accurate and efficient method for path-finding within the transport network.

Definition 6. To accommodate the new constraints in our formulation, we expand our initial sets beyond S and C (as detailed in Def. 1) by defining the following additional sets:

- L: the set of clusters
- $\hat{N}$: the set of noise nodes
- N: the set of regular nodes within S.

Following these modifications, the new graph $I = (G, R)$ will be redefined as follows:

- $G = (S, C)$, where:
  - S is the set of vertices defined as $S = N \cup L$.
  - A cluster is defined as a subset of nodes.
  - C is the set of arcs (connections) between the vertices in S.

Table 3 compares the computational complexities of the classical DBSCAN algorithm and the customized DBSCAN algorithm with the line constraint. The results show that both algorithms exhibit the same complexity of $O(n^2)$, demonstrating that our proposed customization does not impact the overall computational efficiency of DBSCAN.

This approach ensures that the clusters formed are not only dense but also representatively diverse, making them more meaningful and potentially more useful for planning and analysis purposes in complex network environments.

3.4.4. DIJKSTRA: Finding shortest path between nodes

In the literature, numerous methods exist for finding the shortest paths in networks, each with unique advantages and limitations. The *Bellman–Ford* algorithm is versatile, capable of handling graphs with both negative and positive edge weights. However, its practicality diminishes for graphs without negative weights, where its slower performance compared to other methods becomes a drawback.

Another prominent method is the *Floyd–Warshall* algorithm, which excels in scenarios requiring all-pairs shortest paths from a specific node. Despite its theoretical elegance, its computational intensity makes it less suitable for large-scale networks, where performance and efficiency are critical. Similarly, the A^* algorithm is highly efficient when guided by an admissible heuristic, as the heuristic can significantly accelerate the search process. However, in cases where a reliable heuristic is unavailable, A* offers little advantage over alternative methods.

For large, positively weighted graphs without a reliable heuristic, ***Dijkstra's*** algorithm emerges as a robust choice. Known for its efficiency and simplicity, it is widely regarded as one of the most practical methods for shortest path-finding in various applications [14]. Its deterministic nature and optimal performance for graphs with non-negative weights make it especially suitable for transportation networks, where computational efficiency and accuracy are paramount.

The shortest path is a cornerstone of network analysis, particularly in transportation systems, where it directly impacts route optimization, cost minimization, and accessibility improvements. In the context of road networks, identifying the shortest path is essential not only for reducing travel time but also for minimizing fuel consumption and associated costs.

Dijkstra's algorithm is specifically designed to calculate the shortest paths from a source node to all other nodes in a graph with non-negative edge weights. Its core mechanism involves iteratively refining tentative distances until the shortest path to each node is determined.

The algorithm proceeds as follows:

(1) **Initialization:** Assign a distance of infinity to all nodes, except for the source node, which is set to zero.
(2) **Mark Distances:** Mark the source node's distance as permanent, while all other distances remain temporary.
(3) **Activate the Source Node:** Begin with the source node as the active node.
(4) **Calculate Tentative Distances:** For each neighbor of the active node, calculate a tentative distance by summing the active node's distance with the edge weight to the neighbor.
(5) **Update Distances:** If the calculated distance is shorter than the previously recorded distance for a neighbor, update it with the new value.
(6) **Select the Next Active Node:** Choose the node with the smallest temporary distance as the new active node, and mark its distance as permanent.
(7) **Repeat:** Repeat steps 4 to 6 until all reachable nodes have permanent distances, indicating that no further updates are possible.

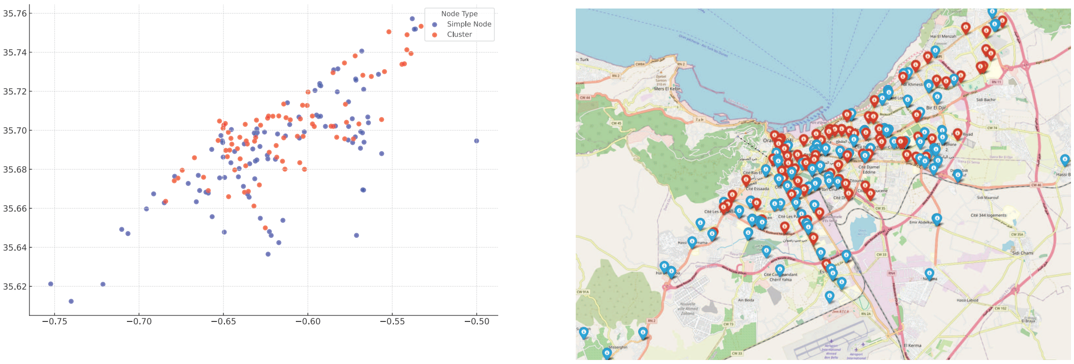
Dijkstra's algorithm is a cornerstone of this study, offering a reliable and efficient means to optimize route planning. Its practical adaptability to large-scale transportation networks underscores its relevance in improving urban mobility and accessibility.

4. Experimental results

This section presents the key findings of our approach to addressing the Oran transportation network, complemented by illustrative screenshots of the proposed framework. Following the data collection and preprocessing detailed in Section 3.3, the preprocessed data containing all critical information about the existing transportation network was prepared. Using this data, the administrator generated the initial graph, as shown in Figure 3, which illustrates the dispersion of stops across Oran's map.

Due to the disconnected nature of the initial graph, the proposed DBSCAN algorithm was applied to generate transfer hubs, significantly improving graph connectivity and preparing it for the route planning process. The resulting graph comprises 231 nodes, including 149 regular nodes and 82 clusters formed by merging stops, as illustrated in Figures 6a and 6b. In these figures, red points (and pins) represent the generated clusters. Table 4 highlights the improvements in key metrics after transitioning from the initial graph to the clusterized graph. The number of nodes is reduced from 592 to 231 due to the merging of stops into clusters, while the number of edges remains nearly the same, indicating that connectivity between nodes is preserved.

The average degree centrality increases significantly from 0.0034 to 0.0221, and the average degree rises from 2.00 to 5.09. These changes reflect a more connected and cohesive network structure, where nodes have more connections on average. Similarly, the graph density improves from 0.0017 to 0.0111, underscoring the enhanced compactness of the graph. The average clustering coefficient increases from 0.0000 to 0.0767, indicating the formation of local clusters within the network, which were absent in the initial graph. Furthermore, the number of connected components reduces from 26 to 1, transforming the graph into a fully connected network. The diameter of the clusterized graph is 21, indicating that the longest shortest path between any two nodes spans 21 stops.



(A) Stops and correspondences represented by Geolocated points

(B) Post-Clustering Urban Bus Stops Map of Oran

FIGURE 6. Post-clustering representation of Oran's transport network.

TABLE 4. Comparison of metrics between initial and clusterized graph.

Metric	Initial graph	Clusterized graph
Number of Nodes	592	231
Number of Edges	592	588
Avg. Degree Centrality	0.0034	0.0221
Avg. Degree	2.00	5.09
Graph Density	0.0017	0.0111
Average Clustering Coeff	0.0000	0.0767
Connected Components	26	1
Diameter	–	21

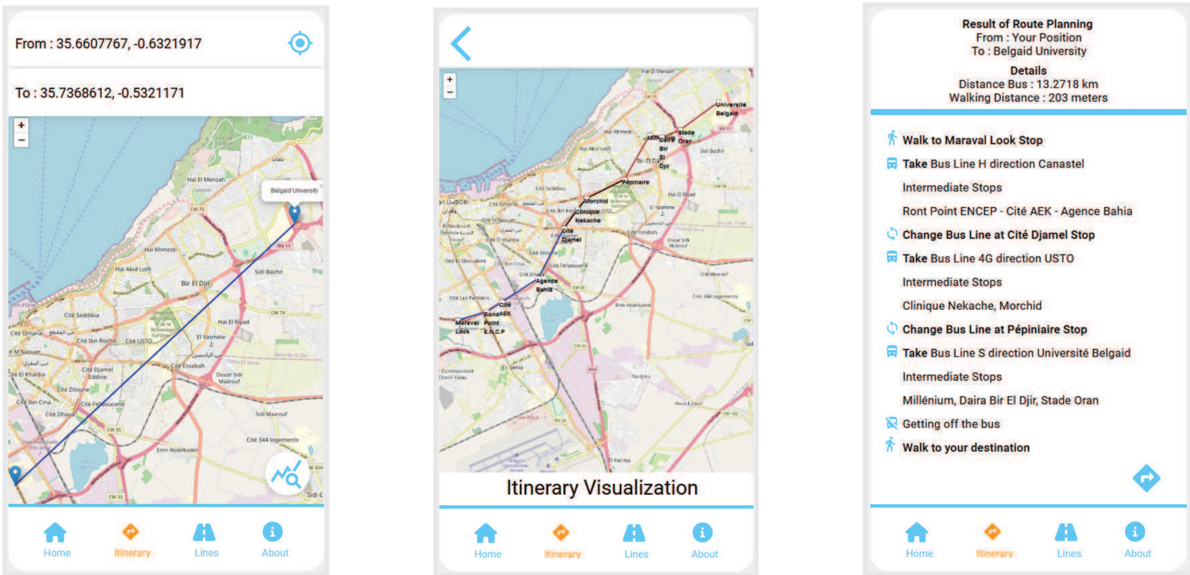
These metrics collectively demonstrate the effectiveness of the DBSCAN clustering algorithm in addressing the initial graph's disconnection, improving both connectivity and network coherence.

The newly constructed graph is tailored for path-finding algorithms, enabling efficient computation of the shortest routes between user-specified starting and destination points. Users can view all bus lines and their full itineraries, including corresponding stops. To search for an optimal itinerary, users enter their starting and destination locations as illustrated in Figure 7a.

Once the clustered graph is generated, Dijkstra's algorithm is employed to find the shortest path between two specified stops. The algorithm provides users with a detailed path page describing the optimal route, including transfer points that indicate which transfer stop the user should proceed to (see Fig. 7c). Users can also visualize the resulting path on a map, as shown in Figure 7b.

When updates to network data are required, such as adding new stops or buses, it is essential to regenerate a connected graph along with its dependent features, including visualizations and the adjacency matrix. To evaluate the performance of the proposed approach, the execution time and memory efficiency of both the customized DBSCAN and Dijkstra's algorithms were assessed using a single route simulated over multiple runs.

The customized DBSCAN was applied to the initial graph, with execution time and memory usage measured over 10 iterations. The execution time remained consistent across runs, varying slightly between 0.03 and 0.09 s, demonstrating stable computational complexity. Similarly, memory usage was stable, averaging between 1.8 and 2.25 MB, indicating a predictable and efficient memory footprint suitable for the graph's size and structure.



(A) Visualization of Line B

(B) Itinirary Visualization Page

(C) Itinirary Result Page

FIGURE 7. Shortest path results.

Dijkstra's algorithm, applied to the clustered graph produced by the customized DBSCAN, exhibited comparable consistency and efficiency. Its execution time ranged from 0.003 to 0.0062 s for the longest path (diameter 21), reflecting its suitability for this graph of 231 nodes and 588 edges. Memory usage was similarly efficient, averaging 0.6 to 0.76 MB across runs, highlighting its predictable and lightweight resource requirements.

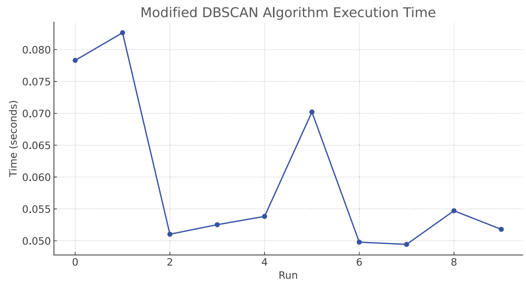
The comparison between the algorithms is visualized in Figure 8, which combines four subfigures: execution time and memory usage for both the customized DBSCAN and Dijkstra's algorithm. These results emphasize the reliability and efficiency of the proposed approach in handling updates to the transport network and facilitating effective path-finding within the clustered graph.

5. Discussion

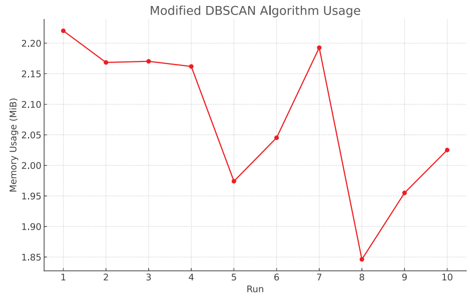
This study marks the first comprehensive attempt to optimize Oran's transportation network for route planning, addressing its inherent challenges through graph modeling and clustering techniques. Prior to this work, the network suffered from fragmented connectivity, making seamless travel difficult and inefficient. The proposed approach, which integrates DBSCAN clustering and Dijkstra's algorithm, has significantly improved network cohesion, providing a robust foundation for accessible and user-friendly urban transit.

Key findings reveal that the initial graph exhibited low connectivity, with a graph density of 0.0017, zero clustering coefficient, and 26 disconnected components. This structure prevents the calculation of graph diameter and makes any path-finding techniques unfeasible. These characteristics rendered the network unsuitable for effective route planning. By introducing transfer hubs through DBSCAN clustering, the network was transformed into a fully connected graph, reducing nodes from 592 to 231 while maintaining almost the same number of edges. Metrics such as graph density (0.0111), average degree (5.09), and clustering coefficient (0.0767) highlight substantial improvements in network structure and usability.

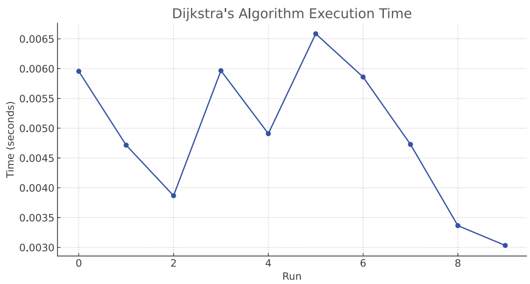
From an operations research perspective, Mixed Integer Programming (MIP) approaches are commonly used to address network design and hub location problems and could, in principle, be adapted to the present context.



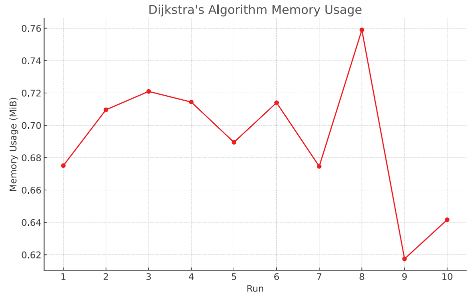
(A) Customized DBSCAN Execution Time



(B) Customized DBSCAN Memory Usage



(C) Dijkstra Execution Time



(D) Dijkstra Memory Usage

FIGURE 8. Comparison of execution time and memory usage for the customized DBSCAN and Dijkstra's algorithm over 10 iterations.

However, such formulations typically require detailed cost structures, demand matrices, and temporal information, which are not available for Oran's public transport network. In addition, solving large-scale MIP models may involve significant computational overhead, making them less suitable for interactive and scalable route-planning applications. For these reasons, this work deliberately favors a graph-based and clustering-oriented approach, which enables the identification of transfer hubs and efficient path computation while remaining compatible with real-world data availability and operational constraints.

However, the study also uncovered several challenges within Oran's transport network:

Structural and Operational Challenges:

- Independent Bus Operators: Each bus line is managed independently without centralized coordination, leading to overlapping routes, inconsistent service quality, and inefficiencies in network optimization.
- Fare Structure: The lack of an integrated ticketing system requires passengers to pay separately for each bus trip, discouraging multi-bus journeys and increasing travel costs.
- Route Variability: Differing inbound and outbound routes create inconsistencies, making navigation challenging for passengers and complicating route planning processes.

Data and Connectivity Limitations:

- Lack of GPS Data: The absence of real-time tracking and scheduling data limits the model's ability to provide accurate travel times, dynamic routing, and real-time updates to passengers.
- Distinct Stop Coordinates: Shared stops between lines often have slightly different geolocations, making manual identification of transfer points difficult and prone to errors. This issue was mitigated by clustering but remains a fundamental limitation of the existing infrastructure.

Graph-Specific Challenges:

- Peripheral Nodes: Some nodes have low degree centrality (0.0087), indicating limited integration into the network and reduced accessibility for certain areas.
- Static Parameter Dependency: DBSCAN's reliance on predefined ϵ and MinPts parameters requires careful manual tuning, which may not adapt well to varying conditions or larger networks.

5.1. Insights and perspectives

To address these challenges, we propose the following solutions:

Centralized Coordination: Establish a centralized transport authority to manage all bus lines, optimize routes, and implement consistent operational guidelines. This would reduce inefficiencies and improve the overall quality of service.

Integrated Fare System: Introduce a unified ticketing system that enables seamless transfers between buses without additional fare payments, increasing affordability and passenger satisfaction.

Route Standardization: Develop standardized inbound and outbound routes to reduce variability and improve the consistency of service for both passengers and route optimization algorithms.

Real-Time Data Integration: Equip buses with GPS technology to enable real-time tracking, schedule adherence monitoring, and dynamic route updates. This would significantly enhance the accuracy and usability of the route planning system.

Enhanced Clustering and Transfer Points: Improve the identification of transfer points by incorporating additional parameters such as passenger flow data and stop usage patterns. This would further refine cluster definitions and reduce ambiguities caused by distinct stop coordinates.

Dynamic Parameter Optimization: Automate the selection of ϵ and MinPts parameters in DBSCAN using optimization techniques like grid search or machine learning models, ensuring scalability and adaptability to different network conditions.

6. Conclusion

This study presents a novel approach to optimizing urban transport in Oran by addressing the issue of non-connected cycles in the transportation network through the integration of DBSCAN clustering and Dijkstra's algorithm. A primary focus of this research is the identification of strategic transfer stops and the connection of previously isolated cycles, enabling a comprehensive analysis and optimization of the transport network.

The methodology began with an overview of existing models and graphical techniques commonly used in public transport systems. By leveraging foundational concepts in graph theory and spatial clustering, we tackled the specific challenges of unidirectional transport in Oran. Our customized DBSCAN technique successfully identified transfer hubs, refining the Dijkstra algorithm to deliver efficient on-demand route planning.

This integrated approach not only addresses Oran's unique challenges but also offers a scalable solution for other urban environments facing similar transportation issues. The framework facilitates improved network connectivity, reduced travel times, and enhanced system efficiency. The low mean execution times and minimal memory requirements demonstrated in this study confirm its practical feasibility for urban transport planning.

Future work should focus on incorporating additional factors into the system, such as bus schedules, valid travel periods, waiting times, transfer opportunities, walking distances, and passenger comfort. These enhancements will further refine route optimization and improve the overall functionality and user experience of the transport network. This pioneering work lays the groundwork for more efficient, user-centric public transport systems, benefiting both practitioners and decision-makers in urban mobility.

Data availability statement

The dataset used in this study is publicly available on Zenodo under the DOI: https://zenodo.org/records/18602691 [26].

References

[1] M. Ahmed, R. Seraj and S.M. Shamsul Islam, The k-means algorithm: a comprehensive survey and performance evaluation. *Electronics* **9** (2020) 1295.

[2] J. Axelsson, Real-world modeling in UML, in 13th International Conference on Software and Systems Engineering and their Applications, Paris, France (2000).

[3] H. Bachiri and N. Triki, Les modes de transport urbain à oran aux défis d'une ville et de sa population. *Erriwak, Revue des Recherches Sociales et Humaines* **8** (2022) 837–853.

[4] M.J. Baena-Toquero, J.L. Muros-Cobos, S. Rodríguez-Valenzuela and J.A. Holgado-Terriza, Towards sustainability in multi-modal urban planners, in 2014 International Conference on Connected Vehicles and Expo (ICCVE). IEEE (2014) 492–497.

[5] H. Bast, D. Delling, A. Goldberg, M. Müller-Hannemann, T. Pajor, P. Sanders, D. Wagner and R.F. Werneck, Route planning in transportation networks, in Algorithm Engineering: Selected Results and Surveys (2016) 19–80.

[6] K.L. Brown, Enhanced route planning and optimization for navigation systems. *J. Navig.* **71** (2018) 666–681.

[7] J.F. Campbell and M.E. O'Kelly, Twenty-five years of hub location research. *Transp. Sci.* **46** (2012) 153–169.

[8] A. Ceder, S. Chowdhury, N. Taghipouran and J. Olsen, Modelling public-transport users' behaviour at connection point. *Transp. Policy* **27** (2013) 112–122.

[9] G. Cui, J. Luo and X. Wang, Personalized travel route recommendation using collaborative filtering based on GPS trajectories. *Int. J. Digit. Earth* **11** (2018) 284–307.

[10] L. Eboli and G. Mazzulla, Service quality attributes affecting customer satisfaction for bus transit. *J. Public Transp.* **10** (2007) 21–34.

[11] A. Ghosal, A. Nandy, A. Kumar Das, S. Goswami and M. Panday, A short review on different clustering techniques and their applications, in Emerging Technology in Modelling and Graphics: Proceedings of IEM Graph 2018 (2020) 69–83.

[12] S. Hoogendoorn-Lanser, R. van Nes and P. Bovy, Path size modeling in multimodal route choice analysis. *Transp. Res. Rec.* **1921** (2005) 27–34.

[13] Z. Huang, S. Gao, C. Cai, H. Zheng, Z. Pan and W. Li, A rapid density method for taxi passengers hot spot recognition and visualization based on dbscan$^+$. *Sci. Rep.* **11** (2021) 9420.

[14] M. Kairanbay and H. Mat Jani, A review and evaluations of shortest path algorithms. *Int. J. Sci. Technol. Res.* **2** (2013) 99–104.

[15] A. Khaleghi, Multi-period hub location problem: a review. *RAIRO-Oper. Res.* **56** (2022) 2431–2466.

[16] K. Khan, S. Ur Rehman, K. Aziz, S. Fong and S. Sarasvady, DBSCAN: Past, present and future, in The Fifth International Conference on the Applications on Digital Information and Web Technologies (ICADIWT 2014), IEEE (2014) 232–238.

[17] F. Kiliç, A public transportation route planning tool: ToTRoP. *J. Public Transp.* (2020). https://library.cu.edu.tr/tezler/8153.pdf.

[18] J. Kozlak, J.-C. Créput, V. Hilaire and A. Koukam, Multi-agent environment for modelling and solving dynamic transport problems. *Comput. Inform.* **28** (2009) 277–298.

[19] M.V. Kravtsova, The formulation of the economic and mathematical model of serving system on the example of public transport bus traffic management, in *Series: Economics and Management*. Science Vector of Togliatti State University (2018) 11–16.

[20] K. Kwakye, Y. Seong and S. Yi, Travel time, distance and costs optimization for paratransit operations using graph convolutional neural network. Preprint arXiv:2205.10507 (2022).

[21] D. Levinson, H.X. Liu and M.G.H. Bell, Transportation and network analysis: current trends: Miscellanea. *Netw. Spat. Econ.* **12** (2012) 293–295.

[22] H. Liu, J. Hu and M. Rauterberg, AIRSF: a new entertainment adaptive framework for stress free air travels, in Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology (2008) 183–186.

[23] L. Magnana, H. Rivano and N. Chiabaut, Implicit GPS-based bicycle route choice model using clustering methods and a LSTM network. *PLoS ONE* (2022).

[24] M.E. Newman, Dynamic route planning algorithms in transportation networks. *Transp. Res. Part C Emerg. Technol.* **100** (2019) 1–17.

[25] A.G.N. Novaes, J.E. Souza de Cursi, A.C.L. da Silva and J.C. Souza, Solving continuous location–districting problems with Voronoi diagrams. *Comput. Oper. Res.* **36** (2009) 40–59.

[26] S. Reguieg, Urban transport network dataset of Oran, Algeria (2026). https://zenodo.org/records/18602691.

[27] X. Wan, H. Pu, P. Schonfeld, T. Song, W. Li and L. Peng, Railway alignment optimization in regions with densely-distributed obstacles based on semantic topological maps. *ICAE* **31** (2023).

[28] P. Xu, A. Pulung Sasmito, B. Yu and A. Sadashiv Mujumdar, Transport phenomena and properties in treelike networks. *Appl. Mech. Rev.*, **68** (2016) 040802.

[29] Z. Xu, Q. Zhang, D. Chen and Y. He, Characterizing the connectivity of railway networks. *IEEE Trans. Intell. Transp. Syst.* **21** (2019) 1491–1502.

[30] Y. Zhang, L.D. Han and H. Kim, Dijkstra's-DBSCAN: Fast, accurate, and routable density based clustering of traffic incidents on large road network. *Transp. Res. Rec.* **2672** (2018) 265–273.

[31] K. Zidi, *Système Interactif d'Aide au Déplacement Multimodal (SIADM)*. Theses, Ecole Centrale de Lille ; Université des Sciences et Technologie de Lille - Lille I, December 2006. https://theses.hal.science/tel-00142159.