Intelligent Systems Research and Applications Journal

www.israj.org
eISSN: 3120-0826

Optimal Path Selection for a Weighted Semidirected Network and its Application

Laxminarayan Sahoo[1], Rakhi Das[1,*]

[1] Department of Computer and Information Science, Raiganj University, Raiganj-733134, India

ARTICLE INFO

Article history:
Received 21 October 2025
Received in revised form 25 November 2025
Accepted 9 December 2025
Available online 22 December 2025

Keywords:
Semidirected graph; Weighted semidirected graph; Shortest path; Congestion

ABSTRACT

Abstract This paper deals with optimal path selection within the context of weighted semidirected graphs, which combine both directed and undirected edges. In real life, it is seen that road networks are sometimes laid out in a unidirectional manner. Sometimes, it is also seen as a combination of unidirectional, bidirectional, or semidirected road structures. In a semidirected path, vehicles traverse both sides, resulting in congestion. Due to congestion, the travel time increased comparatively for semidirected paths relative to undirected paths. The study aims to develop efficient algorithms for identifying the most advantageous routes in scenarios where edges possess varying weights, reflecting diverse costs or distances between nodes. The chapter begins by introducing weighted semidirected graphs, establishing the shortest-path/optimal-path algorithm, and discussing its significance in real-world applications. The proposed algorithm is developed to evaluate the congestion of a semidirected graph, considering both the directionality and weights associated with each edge. The research emphasizes developing optimal path algorithms that enhance computational efficiency while identifying paths that minimize specific criteria, such as total weight, traversal time, or resource utilization. A real-world case study involving transport networks illustrates how the suggested algorithm can be applied in practice. This exploration contributes valuable knowledge to graph theory, operations research, and computer science, offering a sophisticated way to address complex routing challenges in dynamic and interconnected systems.

1. Introduction

Within the ever-changing field of network optimisation, the effective identification of optimal paths is crucial for addressing real-world problems across logistics, telecommunications, and transportation. This chapter further focuses on best-path selection in the complex framework of a Weighted Semidirected Graph (WSDG). By combining directed and undirected edges, WSDGs provide optimal path algorithms in innovative and challenging settings that account for the unique characteristics of interconnected systems with different edge weights. Though the study of traditional graphs has benefited greatly from theoretical advances in graph theory and network optimization, semidirected graphs require specific attention due to their distinct properties. Drawing

[*] *Corresponding author.*
E-mail address: rakhidas2004@gmail.com

https://doi.org/10.59543/f7k2ky14

on foundational studies by researchers such as Dijkstra [1], Floyd [2], and Ford Jr. [3], we first clarify the framework of weighted semidirected graphs to establish a solid foundation. These core algorithms serve as a baseline for understanding conventional path-finding techniques and form the basis for extending their relevance to the specific challenges posed by WSDGs.

The work of Prathik et al. [4] provides an overview of the application of graph theory. Ding et al. [5] proposed a method for computing time-dependent shortest paths on large graphs. A thorough analysis and evaluation of shortest-path algorithms may be found in the work of Magzhan & Jani [6]. Tretyakov et al. [7] investigated fast, fully dynamic estimation of the shortest-path distance between landmarks in very large graphs. Chakraborty et al. [8] applied graph theory to social media. Reijneveld et al. [9] examined the use of graph-theoretic analysis of brain networks. Fu et al. [10] explored heuristic shortest-path methods for transportation applications. Shu-Xi [11] has improved the use of Dijkstra's shortest-path algorithm. Cota-Ruiz et al. [12] developed a recursive shortest-path routing technique with application for wireless sensor network localization. The undirected single-source shortest-path problem in linear time has been examined by Thorup [13]. Madkour et al. [14] surveyed shortest-path methods. The structure and automorphisms of semi-directed graphs have been investigated by Bonato et al. [15]. Investigations on semi-directed graphs for social media networks were conducted by Samanta et al. [16, 17]. Duval & Iourinski [18] carried out studies on semidirect product constructions of directed strongly regular graphs. Wu et al. [19] have solved the routing problem on a set of Clayey graphs that are semidirect products of finite groups. Semi-dynamic shortest-path tree methods have been investigated by Li & Li [20] for directed graphs with adjustable weights. Additionally, Jicy & Mathew [21] examined several additional weighted graph connection factors. Subramani & Kumaresan [22] conducted traffic studies on road networks in Salem and identified necessary transport enhancement projects. Jayaweera et al. [23] have conducted a case study on Sri Lanka to illustrate the use of centrality indicators to identify traffic congestion on road networks. Urban road network traffic prediction and dynamic modelling based on graphs have been investigated by Liu et al. [24].

A sequential graph neural network was developed by Xie et al. [25] to forecast traffic speed on metropolitan roads. One might refer to the work of Chen et al. [26], Zhou et al. [27], Wang et al. [28], and others for additional information regarding traffic networks applying graph theory. The concept of mixed graphs is additionally prevalent in the literature, and work by Ries & Werra [29], Liu & Li [30], Abudayah et al. [31], and others provides applications for mixed graphs. Numerous researchers, including Ryu & Rump [32], Wang et al. [33], Bhatele et al. [34], Barrera et al. [35], Long et al. [36], Erhardt et al. [37], Gu et al. [38], Stopher [39], Duranton& Turner [40], Hensher [41], Wei & Hong-Ying [42], and others, have also studied congestion problems, such as road networks, along with a range of network design problems employing graphs theory concept. To the best of our knowledge and belief, none of the previously cited literature has dealt with the problem of determining the best path for a weighted semidirected graph. To fill this gap, this chapter presents state-of-the-art techniques for handling the complications arising from weighted semidirected edges. The goal of the study is to improve the capacity to identify and improve paths in situations involving directionality and edge weights by combining theoretical underpinnings with real-world applications. This study introduces an algorithm for evaluating the optimal path of a semi-directed graph. The algorithm can compute the shortest or minimum-distance path with low congestion and also minimize traveling time. The concept of a semidirected graph is used to represent the road network system.

To illustrate the proposed approach, a real-world transportation route network is used as an example, and the results are presented.

The significant contribution and objective of the proposed paper are

(i) studied and investigated the semidirected graph and implemented the algorithm for the road/transport networks.
(ii) to minimize the travelling time by evaluating the minimum distance and congestion
(iii) an augmented matrix to represent the undirected and directed edges of the semidirected graph
(iv) for illustration purposes, a real-life road network system has considered and solved
(v) a conclusion drawn to suggest the future scope of research.

1. Some notations, definitions, and preliminaries

Here, we provide a list of notations and their meanings in tabular format (see Table 1), which is used throughout the paper. We have also incorporated some definitions and explanations in this section.

Table 1

Notation and its meanings

Notation	Meaning
$G=(V,E_d,E_u)$	A semidirected graph with V vertices, E_d directed edges, E_u and undirected edges.
e_i	edges
E_d	Directed edges
E_u	Undirected edges
a_{ij}	Adjacency matrix
A_{Gm}, A_{Gu}, A_{Gd}	Augmented matrix, undirected augmented matrix and directed augmented matrix.

Definition 6: (Edge) One of the connections between a network's nodes (or vertices, in a graph) is called an edge (or link). Edges that share a node are known as incident edges. Let *e* be referred to as an edge if and only if it connects two nodes.

Example 1: e_2, e_3, e_4, e_5 are called edges of $(A,C),(A,D),(D,B),(A,B),(C,D)$ pairs of vertices(see Figure 1)

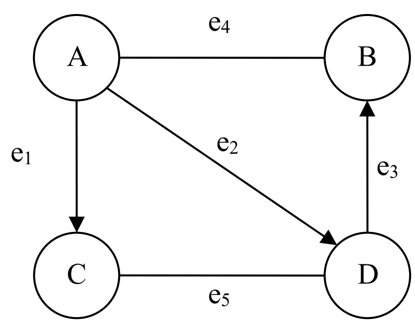


Fig. 1. Semidirected graph.

Definition 1: (Adjacency matrix*)* An adjacency matrix is a method to represent the connection of vertices that are adjacent to each other in a finite graph. Therefore, for a finite graph, if there is a connection between a pair of vertices, then it is represented by the binary value 1. If there is no connection between a pair of vertices, then it is represented by the binary value 0.

2.1 Adjacency matrix for an undirected graph

Let, $G_u = (V, E_u)$ is an un-directed graph with a set of vertices $V = \{v_0, v_1, v_2, ..., v_n\}$. Then the adjacency matrix represented by A_{Gu} of G_u for the listing of the vertices is given by *n,* x_n zero-one matrix with the entries.

$$a_{ij} = \begin{cases} 1 \text{ if} \left(v_i, v_j\right) is \text{ an edge of a graph} \\ 0 \text{ otherwise} \end{cases}$$

Example 2: Let us consider a graph (See Figure 2). The adjacency matrix of this graph with the arbitrary ordering of vertices as follows:

$$A_u = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

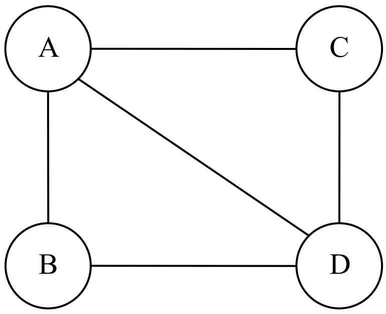


Fig. 2. Undirected graph

2.2 Adjacency matrix for a directed graph

Let, $G_d = \left(V, E_d\right)$ is a directed graph with a set of vertices $V = \{v_0, v_1, v_2, ..., v_n\}$ Then the adjacency matrix represented by A_{Gd}, of G_d for the listing of the vertices is given by *n* by *n* zero-one matrix with the entries.

$$a_{ij} = \begin{cases} 1 \text{ if there } is \text{ an edge from } \mathrm{v}_i \text{ to } \mathrm{v}_j \text{of a graph} \\ 0 \text{ otherwise} \end{cases}$$

Example 3: Let us consider a directed graph (see Figure 3). The adjacency matrix of this graph with the arbitrary ordering of vertices *a, b, and c* is as follows:

$$A_d = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

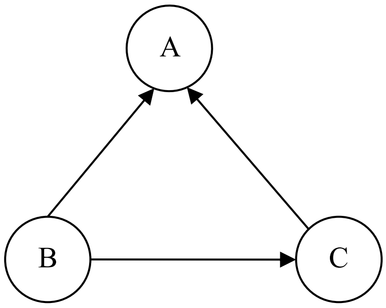


Fig. 3. Directed graph

Definition 2: (Augmented Adjacency Matrix of mixed graph) An augmented adjacency matrix $A_{Gm} = (A_{Gd} / A_{Gu})$ represents the semidirected graph $G_m = (V, E_u, E_d)$ where A_{Gu} and A_{Gd} are n x n representing adjacency matrices of the edge-disjoint sub-graphs and V represents the set of vertices, E_u representing the undirected edges, E_d is representing the directed edges.

Example 4: Considered a mixed graph (see Figure 4). The augmented adjacency matrix of this graph is as follows:

$$A_{Gm} = (A_{Gd} / A_{Gu}) = \left(\begin{matrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{matrix} \middle/ \begin{matrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{matrix} \right)$$

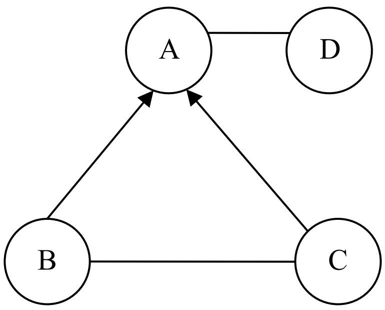


Fig. 4. Mixed graph

Definition 3: (Congestion*)* In Transportation, traffic congestion is a condition of transport that is characterized by slower speeds, longer trip times, and increased vehicular queuing. A traffic jam or (informally) a traffic snarl-up occurs when all moving cars are completely stopped for an extended time.

Definition 4: (**Semidirected graph**) A graph $G = (V, E_d, E_u)$ is named a semidirect graph, where V represents a set of nonempty sets of vertices, E_d as a set of directed edges, and E_u as a set of undirected edges.

Example 5: In Figure 1, if we consider V as a set of nonempty vertices or nodes, $V = (A, B, C, D)$ and $E_u = (e_4, e_5)$ and $E_d = (e_1, e_2, e_3)$ then it is a semidirected graph.

2. Weighted semidirected graph

Here, we introduce the concept of a weighted semidirected graph and present some definitions.

Definition 5: (**Weighted Semidirected Graph**) A semidirected graph $G = (V, E_d, E_u)$ is called a weighted semidirected graph if and only if all edges (directed or undirected) of the graph G have a numerical value $W(e)$ or the weight of the graph G.

Example 6: We have considered a graph (see Figure 5). The weights of the edges are {6,7,4,5,9}, where E_d = {6,7,5} are sets of directional weights and E_u ={4,9} are sets of undirected edges.

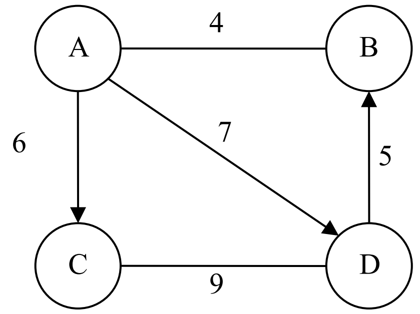


Fig. 5. Weighted Semidirected graph

3. Different Types of Shortest Paths

There have been numerous theoretical discussions of shortest-path problems in the literature. Here are a few examples as follows:

Single-pair shortest path problem (SPSP):

The single-pair shortest path (SPSP) algorithm determines the shortest route between a single or specific pair of nodes. This algorithm determines the fastest route to travel from vertex u to vertex v. The shortest path problem for a single pair has been resolved using the well-known A^* search technique.

Single-source shortest path problem (SSSP):

The shortest path between a specific source node and all other remaining nodes has been calculated in this instance of the shortest-path problem. It finds the shortest route between source s and all vertices $v \in V$. Two well-known algorithms for finding the shortest path are Dijkstra's and Bellman-Ford.

Single-destination shortest path problem (SDSP):

This algorithm computes the shortest path from all vertices to a single target vertex. This algorithm can be reduced to a single-source shortest path problem by reversing the direction of each edge in the graph. A well-known Dijkstra algorithm has been applied to solve the problem of finding the shortest path to the destination.

All pairs' shortest path problem (APSP):

The algorithm computes the shortest path between each pair of vertices. It helps to find the shortest path v for all u. The Johnson and Floyd–Warshall algorithms are well-known for finding the shortest path for all pairs.

Here, we studied various shortest-path algorithms and observed that many algorithms can be applied to both directed and undirected graphs. They are not applicable for semidirected graphs or those associated with directed and undirected edges. In this situation, our proposed algorithm is particularly effective for computing the shortest or optimal path in a semidirected graph. A detailed review of algorithms, their applications, and their time complexities is provided in Table 2.

Table 2

Review of algorithms and their applications, together with their time complexity

Algorithm	**Directed edge graph**	**Undirected edge graph**	**Semidirected graph**	**All source graph**	**Determine Congestion**	**Time complex**
Dijkstra's Algorithm	Applicable	Applicable	Not Applicable	Not applicable	Not applicable	$\Theta(\lvert V\rvert + \lvert E\rvert)$
Bellman–Ford Algorithm	Applicable	Not applicable	Not applicable	Not applicable	Not applicable	$\Theta(VE)$
Floyd–Warshall	Applicable	Not applicable	Not applicable	Applicable	Not applicable	$\Theta(n^3)$
The study of an optimal path selection for a weighted semidirected network	Applicable	Applicable	Applicable	Applicable	Applicable	$O(n^2)$

3. Algorithm for a semidirected graph

Here, we propose an algorithm for finding the shortest path and computing the congestion of a semidirected graph. Different steps of this algorithm are as follows:

Step 1: Initially, take a graph $G = (V, E_d, E_u)$.

Step 2: Create an Augmented matrix (A_{Gm}) for the graph G .

Step 3: Select the initial vertex (v_0).

Step 4: Select the final vertex (F).

Step 5: v_0 is incidence with E_d and E_u together

$$[E_d] = 1 \text{ then go to } \mathrm{A}_{E_d} \left[\mathrm{Q}\, \mathrm{A}_{E_d} \rightarrow \text{Augmented matrix of } \mathrm{E}_d \right]$$

$$[E_d] = 0 \text{ then go to } \mathrm{A}_{E_u} \left[\mathrm{Q}\, \mathrm{A}_{E_u} \rightarrow \text{Augmented matrix of } \mathrm{E}_u \right]$$

and $dist[E_u] \rangle dist[E_d]$

$$\text{then P=W}\left[E_d\right] \qquad \left[\text{Q}\in \rightarrow 0\right]$$

$$\left[E_d\right]=0 \text{ then go to } \mathrm{A}_{E_u}\left[\text{Q } \mathrm{A}_{E_u} \rightarrow \text{Augmented matrix of } \mathrm{E}_u\right]$$

$$\text{and } dist\left[E_u\right]\langle dist\left[E_d\right]$$

$$\text{then P=W}\left[E_u\right]$$

$$\in=\in_1+\in_2+\ldots+\in_n \quad \left[\text{Q}\in \rightarrow \text{congestion}\right]$$

Step 6: update the vertex. Go to step 5 until you get the final vertex.

The time complexity of the above algorithm is $O(n^2)$. And $\in$ is considered as 0 while there is no traffic or no congestion, mainly it is found in a one-way route or a directed path, and $\in$ is belongs within $(0.0 \text{ to } 0.5)$ depends upon traffic or congestion.

5.1. Real case study and survey result

We have considered a real case study to illustrate our algorithm implemented on a road network (see Figure 7). This road network is part of the Grand Trunk (GT) road in the Durgapur area of Paschim Bardhaman District, West Bengal, India. We can determine the optimal path from Durgapur Gandhi More to Ukhra, West Bengal, India. To form the network, we collected the distances between places (nodes) from Google Maps (see Figure 6). After collecting the distances for each node, we present a real-world case study based on the network (see Figure 7). All node descriptions are mentioned in Table 4, and the distances of all nodes are mentioned in Table 5. The proposed algorithm is implemented on this network (see Figure 6). For the successful execution of the algorithm, we used the Python (version 3.6.15) software and considered the congestion $\in$ within [$(0.0 \text{ to } 0.5)$] and the configurations of the computing machine are mentioned in Table 3.

Table 3

Details of hardware configurations of computing machine

Processor	**Intel(R) Core (TM) i3-7020U CPU @ 2.30GHz 2.30 GHz**
Installed RAM	12.0 GB (10.4 GB usable)
System type	64-bit operating system, x64-based processor
Windows edition	10 Home Single Language
Version	21H2
Device's name	Asus Vivobook15 Intel Core i3 7th Gen

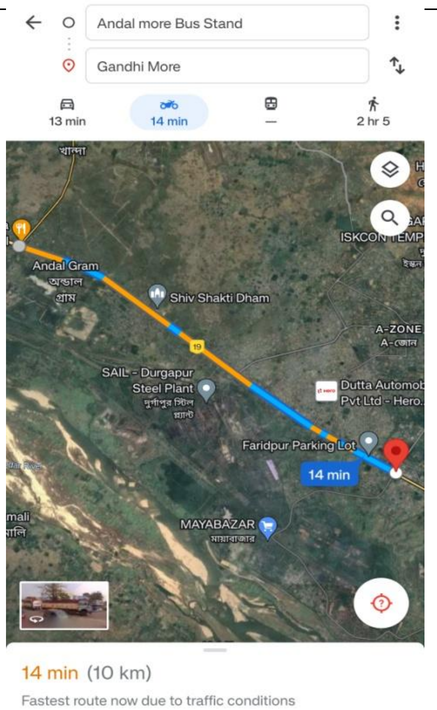
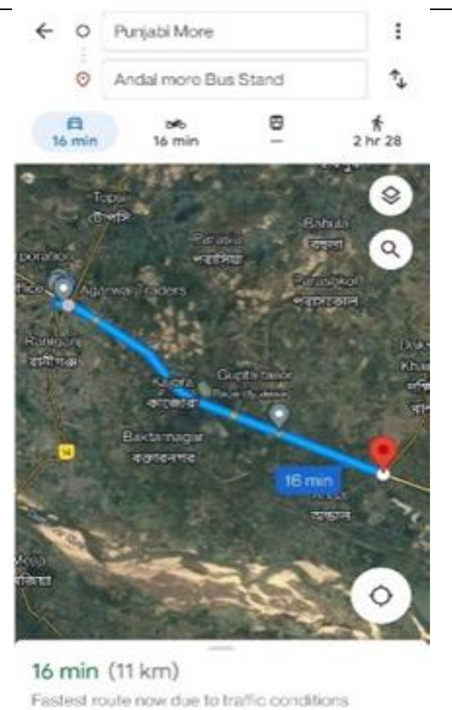
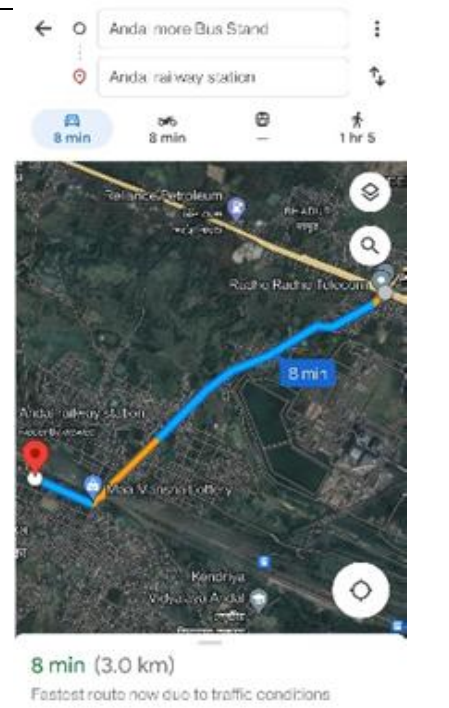
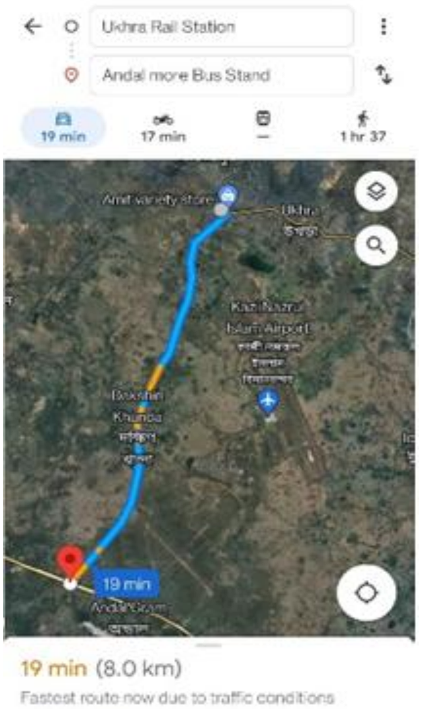
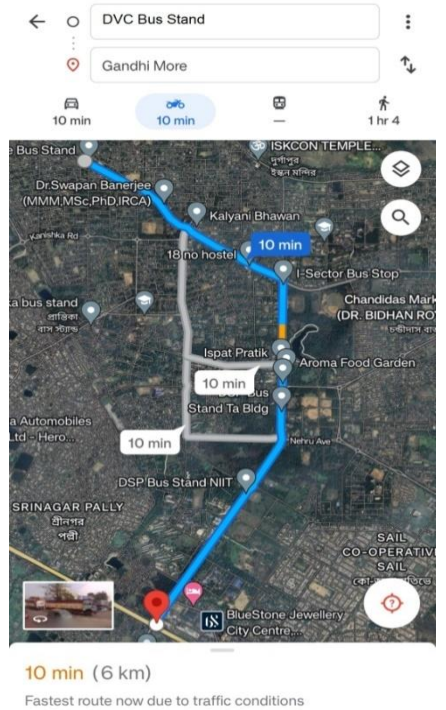
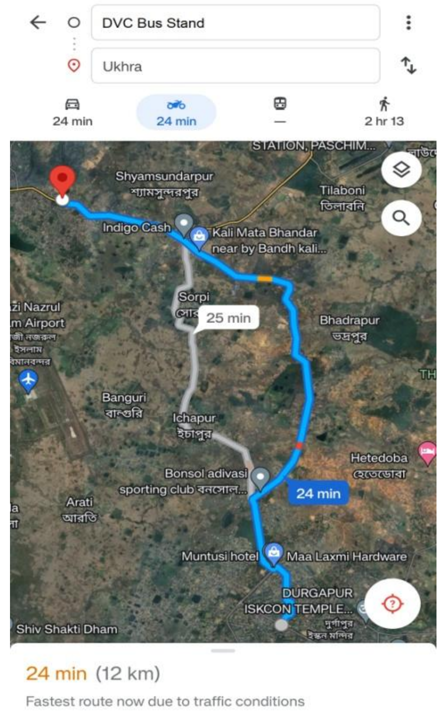
		
Durgapur Gandhi More-Andal More - 10 Km	Andal More -Raniganj Panjabi More - 11 km	Andal Station Road -Andal More - 3 km
		
Ukhra – Andal More - 8 km	Durgapur Gandhi More - DVC Bus Stand Shivaji Road - 6 km	DVC Bus Stand Shivaji Road – Ukhra - 12 km

Fig. 6. Google map of different nodes

We collected real data from Google Maps (see Figure 6) and implemented our proposed algorithm provided in Section 5. Table 4 contains the node (place) descriptions, and Table 5 provides the distances between nodes.

Table 4

Node descriptions

Node	Place name
A	Durgapur Gandhi More
B	Andal More
C	Raniganj Panjabi More
D	DVC Bus Stand Shivaji Road
E	Ukhra
F	Andal Station Road

Table 5

Distance of each node

Path type	Node	Path name	Distance
One way	A-B	Durgapur Gandhi More-Andal More	10 Km
One way	B-C	Andal More -Raniganj Panjabi More	11 km
Bidirectional way	F-B	Andal Station Road -Andal More	3 km
Bidirectional way	E-B	Ukhra – Andal More	8 km
One way	A-D	Durgapur Gandhi More - DVC Bus Stand Shivaji Road	6 km
Bidirectional way	D-E	DVC Bus Stand Shivaji Road - Ukhra	12 km

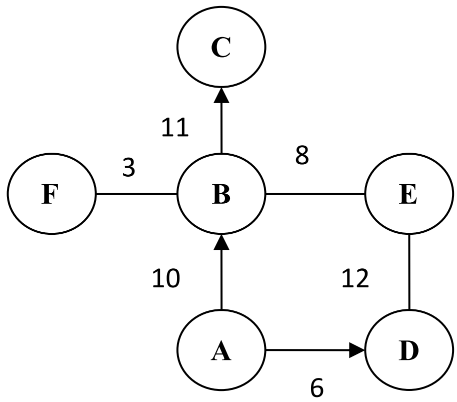


Fig. 7. Semidirected network of a road network system

We have now executed the proposed algorithm mentioned in Section 5, using all the parametric values provided in Tables 4 and 5. Here, we want to travel from Durgapur Gandhi More to Ukhra. So, in our problem, the initial node is Durgapur Gandhi More and the destination node is Ukhra. Here, we have assumed that the congestion from A to B is 0.05 and from A to D is 0.01. During the execution process, our algorithm generated an augmented matrix for the network (see Figure 7) as follows:

$$A_{Gm} = (A_{Gd} / A_{Gu})$$

$$A_{Gm} = \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \Bigg/ \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix}$$

After execution of the algorithm, we have obtained the optimal path as $A \to D \to E$ i.e., Durgapur Gandhi More to Ukhra, the path is optimal when we go via the route (Durgapur Gandhi More - DVC Bus Stand Shivaji Road) and then we reach Ukhra via the route (DVC Bus Stand Shivaji Road – Ukhra). There might be a second route that goes via Durgapur Gandhi More-Andal More and then Ukhra-Andal More in a bidirectional manner, according to the graph (see Figure 7). Due to strict traffic regulations, Durgapur Gandhi More-Andal More is often congested because it is part of the Grand Trunk Road. Thus, our algorithm selects the least-congested ideal path. As a result, the best path's total distance is 18, and its total congestion is 0.01. It should be noted that although the total distance is the same when travelling in both directions via Durgapur Gandhi More-Andal More and Ukhra - Andal More, there is always a 0.05 congestion on the route.

4. Conclusions

To sum up, the research focused on identifying the best path in a weighted semidirected graph. It showed that traffic congestion poses different difficulties for undirected paths than for directed paths in these graphs. The study effectively addressed congestion problems and proposed feasible solutions by using an augmented matrix and an algorithm for the road network. The results highlight how crucial it is to consider semidirected graph properties while improving path choices in practical settings. Transportation planners and engineers can gain significant insights by understanding the difference between directed and undirected paths, which is crucial for understanding and reducing congestion. By exploring and refining optimal strategies, the research lays the groundwork for future work and adds to the expanding body of knowledge in graph theory applied to transportation systems. Future work might focus on improving algorithms, incorporating real-time data, and examining the effects of dynamic traffic conditions, as this study lays the foundation for tackling traffic-related problems using semidirected graphs. The suggested method may be made more flexible by applying machine learning models to forecast traffic patterns. The accuracy and effectiveness of path selection algorithms may also be enhanced by combining innovative technology and data analytics, given the dynamic nature of transportation networks. As a result, the study offers a solution for semidirected graph congestion. It creates new opportunities for further investigation, creative thinking, and valuable applications in the dynamic fields of network design and transportation.

Author Contributions

Conceptualization, L.S. and R.D.; methodology, L.S.; software, R.D.; validation, R.D.; formal analysis, R.D.; investigation, L.S.; resources, R.D.; data curation, L.S.; writing—original draft preparation, R.D.; writing—review and editing, L.S.; visualization, R.D.; supervision, L.S. All authors have read and agreed to the published version of the manuscript.

Funding

This research received no external funding.

Data Availability Statement

In this section, please provide details regarding where data supporting reported results can be found, including links to publicly archived datasets analyzed or generated during the study. You might choose to exclude this statement if the study did not report any data.

Conflicts of Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This research was not funded by any grant.

References

[1] Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. Numerische Mathematik, 1(1), 269–271. https://doi.org/10.1007/bf01386390

[2] Floyd, R. W. (1962). Algorithm 97: Shortest path. Communications of the ACM, 5(6), 345. https://doi.org/10.1145/367766.368168

[3] Ford, L. R., Jr. (1956). Network flow theory. RAND Corporation.

[4] Prathik, A., Uma, K., & Anuradha, J. (2016). An overview of application of graph theory. International Journal of ChemTech Research, 9(2), 242–248.

[5] Ding, B., Yu, J. X., & Qin, L. (2008). Finding time-dependent shortest paths over large graphs. In Proceedings of the 11th International Conference on Extending Database Technology: Advances in Database Technology (pp. 205–216). https://doi.org/10.1145/1353343.1353371

[6] Magzhan, K., & Jani, H. M. (2013). A review and evaluations of shortest path algorithms. International Journal of Scientific and Technology Research, 2(6), 99–104.

[7] Tretyakov, K., Armas-Cervantes, A., García-Bañuelos, L., Vilo, J., & Dumas, M. (2011). Fast fully dynamic landmark-based estimation of shortest path distances in very large graphs. In Proceedings of the 20th ACM International Conference on Information and Knowledge Management (pp. 1785–1794). https://doi.org/10.1145/2063576.2063834

[8] Chakraborty, T., Dutta, S., Mondal, S., & Nath, A. (2018). Application of graph theory in social media. International Journal of Computer Sciences and Engineering, 6(10), 722–729. https://doi.org/10.26438/ijcse/v6i10.722729

[9] Reijneveld, J. C., Ponten, S. C., Berendse, H. W., & Stam, C. J. (2007). The application of graph theoretical analysis to complex networks in the brain. Clinical Neurophysiology, 118(11), 2317–2331. https://doi.org/10.1016/j.clinph.2007.08.010

[10] Fu, L., Sun, D., & Rilett, L. R. (2006). Heuristic shortest path algorithms for transportation applications: State of the art. Computers & Operations Research, 33(11), 3324–3343. https://doi.org/10.1016/j.cor.2005.03.027

[11] Wang, S.-X. (2012). The improved Dijkstra's shortest path algorithm and its application. Procedia Engineering, 29, 1186–1190. https://doi.org/10.1016/j.proeng.2012.01.110

[12] Cota-Ruiz, J., Rivas-Perea, P., Sifuentes, E., & Gonzalez-Landaeta, R. (2016). A recursive shortest path routing algorithm with application for wireless sensor network localization. IEEE Sensors Journal, 16(11), 4631–4637. https://doi.org/10.1109/jsen.2016.2543680

[13] Thorup, M. (1999). Undirected single-source shortest paths in linear time. Journal of the ACM, 46, 362–394. https://doi.org/10.1145/316542.316548

[14] Madkour, A., Aref, W. G., Rehman, F. U., Rahman, M. A., & Basalamah, S. (2017). A survey of shortest-path algorithms. arXiv preprint arXiv:1705.02044. https://doi.org/10.48550/arXiv.1705.02044

[15] Bonato, A., Delic, D., & Wang, C. (2010). The structure and automorphisms of semi-directed graphs. Journal of Multi-Valued Logic and Soft Computing, 27, 161–173.

[16] Samanta, S., Pal, M., Mahapatra, R., Das, K., & Bhadoria, R. S. (2021). A study on semi-directed graphs for social media networks. International Journal of Computational Intelligence Systems, 14(1), 1034–1041. https://doi.org/10.2991/ijcis.d.210301.001

[17] Das, R., Sahoo, L., Samanta, S., Simic, V., & Senapati, T. (2022). Identifying the shortest path of a semidirected graph and its application. Mathematics, 10(24), 4807. https://doi.org/10.3390/math10244807

[18] Duval, A. M., & Ourinski, D. I. (2003). Semidirect product constructions of directed strongly regular graphs. Journal of Combinatorial Theory, Series A, 104(1), 157–167. https://doi.org/10.1016/s0097-3165(03)00141-9

[19] Wu, F. L., Lakshmivarahan, S., & Dhall, S. K. (2000). Routing in a class of Cayley graphs of semidirect products of finite groups. Journal of Parallel and Distributed Computing, 60(5), 539–565. https://doi.org/10.1006/jpdc.2000.1624

[20] Li, S., & Li, Y. (2021). Semi-dynamic shortest-path tree algorithms for directed graphs with arbitrary weights. arXiv preprint arXiv:1903.01756. https://doi.org/10.48550/arXiv.1903.01756

[21] Jicy, N., & Mathew, S. (2014). Some new connectivity parameters for weighted graphs. *Journal of Uncertainty in Mathematics Science*, *2014*, 1-9. https://doi.org/10.5899/2014/jums-00002

[22] Subramani, T., & Kumaresan, P. K. (2012). Traffic study on road network and identify transport improvement projects required in Salem. International Journal of Management, IT and Engineering, 2(7), 190–200.

[23] Jayaweera, I. M. L. N., Perera, K. K. K. R., & Munasinghe, J. (2017). Centrality measures to identify traffic congestion on road networks: A case study of Sri Lanka. IOSR Journal of Mathematics, 13(2), 13–19. https://doi.org/10.9790/5728-1302011319

[24] Liu, T., Jiang, A., Miao, X., Tang, Y., Zhu, Y., & Kwan, H. K. (2021). Graph-based dynamic modeling and traffic prediction of urban road networks. IEEE Sensors Journal, 21(24), 28118–28130. https://doi.org/10.1109/jsen.2021.3124818

[25] Xie, Z., Lv, W., Huang, S., Lu, Z., Du, B., & Huang, R. (2019). Sequential graph neural network for urban road traffic speed prediction. IEEE Access, 8, 63349–63358. https://doi.org/10.1109/access.2019.2915364

[26] Chen, Z., Xu, J., Lin, Y., Feng, B., & Huang, Z. (2021). A traffic flow forecasting method regarding traffic network as a digraph. International Journal of Pattern Recognition and Artificial Intelligence, 35(15), 2159043. https://doi.org/10.1142/s0218001421590436

[27] Zhou, F., Yang, Q., Zhong, T., Chen, D., & Zhang, N. (2020). Variational graph neural networks for road traffic prediction in intelligent transportation systems. IEEE Transactions on Industrial Informatics, 17(4), 2802–2812. https://doi.org/10.1109/tii.2020.3009280

[28] Wang, Z. H., Shi, S. S., Yu, L. C., & Chen, W. Z. (2012). An efficient constrained shortest path algorithm for traffic navigation. Advanced Materials Research, 356, 2880–2885. https://doi.org/10.4028/www.scientific.net/amr.356-360.2880

[29] Ries, B., & De Werra, D. (2008). On two coloring problems of a mixed graph. European Journal of Combinatorics, 29, 712–725. https://doi.org/10.1016/j.ejc.2007.03.006

[30] Liu, J., & Li, X. (2015). Hermitian-adjacency matrices and Hermitian energies of mixed graphs. Linear Algebra and Its Applications, 466, 182–207. https://doi.org/10.1016/j.laa.2014.10.028

[31] Abudayah, M., Alomari, O., & Sander, T. (2023). Incidence matrices and line graphs of mixed graphs. *Special Matrices*, *11*(1), 20220176. https://doi.org/10.1515/spma-2022-0176

[32] Ryu, S., Rump, C., & Qiao, C. (2003). Advances in internet congestion control. IEEE Communications Surveys & Tutorials, 5(1), 28–39. https://doi.org/10.1109/comst.2003.5342228

[33] Wang, Y., Liu, H., Han, K., Friesz, T. L., & Yao, T. (2015). Day-to-day congestion pricing and network resilience. *Transportmetrica A: Transport Science*, *11*(9), 873-895. https://doi.org/10.1080/23249935.2015.1087234

[34] Bhatele, A., Titus, A. R., Thiagarajan, J. J., Jain, N., Gamblin, T., Bremer, P. T., & Kale, L. V. (2015). Identifying the culprits behind network congestion. In Proceedings of the IEEE (pp. 113–122). https://doi.org/10.1109/ipdps.2015.92

[35] Barrera, J., & Garcia, A. (2014). Dynamic incentives for congestion control. IEEE Transactions on Automatic Control, 60(2), 299–310. https://doi.org/10.1109/tac.2014.2348197

[36] Long, J., Gao, Z., Ren, H., & Lian, A. (2008). Urban traffic congestion propagation and bottleneck identification. Science in China Series F: Information Sciences, 51(7), 948–964. https://doi.org/10.1007/s11432-008-0038-9

[37] Erhardt, G. D., Roy, S., Cooper, D., Sana, B., Chen, M., & Castiglione, J. (2019). Do transportation network companies decrease or increase congestion? Science Advances, 5(5), e2670. https://doi.org/10.1126/sciadv.aau2670

[38] Gu, Y., Jiang, C., Zhang, J., & Zou, B. (2021). Subways and road congestion. American Economic Journal: Applied Economics, 13(2), 83–115. https://doi.org/10.1257/app.20190024

[39] Stopher, P. R. (2004). Reducing road congestion: A reality check. Transport Policy, 11(2), 117–131. https://doi.org/10.1016/j.tranpol.2003.09.002

[40] Duranton, G., & Turner, M. A. (2011). The fundamental law of road congestion: Evidence from U.S. cities. American Economic Review, 101(6), 2616–2652. https://doi.org/10.1257/aer.101.6.2616

[41] Hensher, D. A. (2018). Tackling road congestion—What might it look like in the future under a collaborative and connected mobility model. Transport Policy, 66, A1–A8. https://doi.org/10.1016/j.tranpol.2018.02.007

[42] Wei, L., & Hong-Ying, D. (2016). Real-time road congestion detection based on image texture analysis. Procedia