mathematics

Article

Identifying the Shortest Path of a Semidirected Graph and Its Application

Rakhi Das [1], Laxminarayan Sahoo [1,*], Sovan Samanta [2], Vladimir Simic [3] and Tapan Senapati [4]

[1] Department of Computer and Information Science, Raiganj University, Raiganj 733134, India
[2] Department of Mathematics, Tamralipta Mahavidyalaya, Tamluk 721636, India
[3] Faculty of Transport and Traffic Engineering, University of Belgrade, Vojvode Stepe 305, 11010 Belgrade, Serbia
[4] Department of Mathematics, Padima Janakalyan Banipith, Jhargram 721517, India
* Correspondence: lxsahoo@gmail.com

Abstract: The basic goal of this research is to find the shortest path of a semidirected graph and apply it to the road network system. In the field of graph theory, networks are described as directed graphs, undirected graphs, or a combination of both. However, in the modern era of computing, several networks, such as social media networks, granular networks, and road transport networks, are not linked to any of the aforementioned network categories and in reality are a hybrid of networks with both directed and undirected interconnections. To better understand the notion of these types of networks, semidirected graphs have been developed to represent such networks. In a semidirect graph, both the directed and undirected edges concepts have been introduced together in a graph. In reality, it has been noted that for every node (source/destination) which is connected to another, some links/connections are directed—i.e., one-way—and some links/connections are undirected—i.e., two-way. Considering that there is no specific direction provided and that two nodes are connected, we have established the concept of an undirected edge as two-way connectivity since this provides for nodes in both ways. In this study, the road network system has been modelled using the concept of a semidirected graph, and the shortest path through it has been determined. For the purposes of illustration, we have used a real transportation road network in this case, and the computed results have been displayed.

Keywords: Semidirected graph; weighted graph; shortest path; mixed graph; road networks; Floyd–Warshall algorithm

MSC: 05C90; 90C35

check for updates

Citation: Das, R.; Sahoo, L.; Samanta, S.; Simic, V.; Senapati, T. Identifying the Shortest Path of a Semidirected Graph and Its Application. *Mathematics* **2022**, *10*, 4807. https://doi.org/10.3390/math10244807

Academic Editor: Andrea Scozzari

Received: 19 November 2022
Accepted: 15 December 2022
Published: 17 December 2022

1. Introduction

Since the development of the computer, optimal path selection for networks has been given more attention, and computation upgrading and optimization also have progressed over time. Researchers somewhere at time attempting continuously to put the most effective path selection algorithm in operation. For the graphical presentation of any network system, graph theory has taken on an important role. The fundamental idea of graph theory was first introduced by Swiss mathematician Leonhard Euler in the 18th century. Since then, the branch of graph theory has been rapidly evolving. Now, graph theory has several branches of research [1]. When representing a network graphically, the notation of both nodes and edges is important. Although it is presented by an undirected graph, investigating brain networks is one of the key research areas for scientists. The idea that brain networks may be a mixed directional combination of undirected and directed has been established by network researchers and scientists [2]. Another such example is networks of roads; some of them have a direction from source to destination and others have no direction at all. As a result, when a vehicle follows the directed path, it has a specific direction known as the one-way route. However, when a vehicle is moving in an arbitrary route from one point to another, it is logically travelling in both directions, making the arbitrary direction a bidirectional route. In order to represent the road network graphically, it may contain both directed and undirected edges, with the undirected edges logically being bidirectional edges (two-way routes). The mixed graph coloring problem was studied by Ries and Werra in 2008 [3]. It is necessary to keep in mind that mixed graphs usually include both directed and undirected edges. Hermitian adjacency matrices and Hermitian energies of mixed graphs have been studied by Liu and Li [4]. Using the idea of monographs, Abudayah et al. [5] explored mixed graphs and generalized the outcomes from earlier studies on the topic. Mixed adjacency matrixes of a mixed graphs have been studied by Adiga et al. [6]. Notable contributions made by different researchers to the analyses of different graph-related problems are presented in Table 1.

Table 1. Notable contributions implemented by different researchers to the analyses of different graph-related problems.

References	Year	Notable Contributions
Orlin and Madduri [7]	1999	A faster algorithm for the single-source shortest path problem with a limited number of distinct positive lengths
Thorup [8]	1999	Shortest path from an undirected single source in linear time
Katerinis and Tsikopoulosk [9]	2004	Edge connectivity and graph orientation
Bonato et al. [10]	2010	Characteristics of Semidirected Graphs and Their Automorphisms
Wang et al. [11]	2012	An effective technique for restricted shortest paths in traffic
Jicy and Mathew [12]	2014	New connection metrics for weighted graphs
Mohammed [13]	2017	Mixed graph isomorphism and its representation
Sotskov [14]	2020	A Historical Review of Mixed Graph Colorings
Deen et al. [15]	2021	Extending of Edge Even Graceful Labeling of Graphs to Strong r-Edge Even Graceful Labeling
Li and Li [16]	2021	Algorithms for directed graphs that use semi-dynamic shortest-path trees and adjustable weights

In this study, we took into account semidirected graphs, whose edges can be visualized as both directed and undirected graphs. All undirected edges are first converted into two bidirectional edges for the purpose of determining the shortest path in a semidirect graph, and all pair search techniques are then applied using the Floyd–Warshall algorithm. As we know, the Floyd–Warshall algorithm uses dynamic programming approaches to obtain the optimal solution with a coherent decision. Therefore, for multiple possible solutions, this algorithm is mostly used. Some research, shown in Table 2, discusses the use of the Floyd–Warshall algorithm for determining the best path.

Table 2. Use of the Floyd–Warshall algorithm for determining the best path.

References	Year	Contributions
Singh and Mishra [17]	2014	Performance Comparison of the Floyd–Warshall and Rectangular Algorithms
Ramadiani et al. [18]	2018	Floyd–Warshall algorithm based on Android to find the shortest path

Table 2. *Cont.*

References	Year	Contributions
Triana and Syahputri [19]	2018	Implementation Shortest Path of Garage Using the Floyd–Warshall Algorithm
Brodnik et al. [20]	2022	Floyd–Warshall algorithm modifications with approximately quadratic expected-time
Sahoo et al. [21]	2022	Maximum Connectivity in Wireless Sensor Networks with Uncertainty Using a Modified Floyd–Warshall algorithm

We have modified the Floyd–Warshall algorithm that is currently in use to establish the optimal path (result) of a semidirected graph. The modified Floyd–Warshall algorithm usually equates the distances (intersections) between points from the initial point to the destination point and compares each route alternative to its measuring distance. Every edge in a semidirect graph/network must have a direction in order to apply the proposed methodology. Therefore, we converted the undirected edges into two directed edges before implementing the procedure. In this paper, the road network system has been modelled using the concept of a semidirected graph, and the shortest path through it has been determined. In this case, a practical transportation road network has been adopted for example purposes, and the computed results have been presented.

2. Contributions and Objectives

The contributions and objectives of the proposed research are listed below.

- We have studied the semidirected graph and implemented it on road/transport networks.
- We have investigated the optimal shortest path of the semidirected graph.
- We have modified the Floyd–Warshall algorithm to find a semidirected graph's shortest path.
- We have converted the undirected graph into a bidirectional one using the Floyd–Warshall algorithm.
- We have implemented the Floyd–Warshall algorithm for the bidirectional graph.
- A real case study has been considered and solved for illustration purposes.
- Conclusions and future direction of the proposed research have been made.

2.1. The Majority of the Work Conducted for This Research Can Be Summarized as Follows

(i) We have studied the semidirected graph and implemented it on road/transport networks. (ii) We have modified the Floyd–Warshall algorithm for finding the shortest path of a semidirected graph. (iii) We used the Floyd–Warshall algorithm for a bidirectional graph to find the optimal shortest path. (vi) A real case study has been considered and solved for experiment purposes.

The rest of the work is presented as follows. Section 2 presents Definition and Preliminaries to develop the entire paper. In order to accomplish the study, Section 3 provides conversion of undirected edges to directional edges. Different types of shortest path are discussed in Section 4. An algorithm for finding shortest path of a semidirected graph is presented in Section 5. Numerical example and illustration with case study have been presented in Section 6. The concluding remarks of the research are included in Section 7. We provide a list of notations and their meanings in a tabular format as below (see Table 3), which are used in the paper to help the reader.

Table 3. Notation and meanings.

Notations	Meaning
$G = (V, E_d, E_u)$	Semidirected graph G with vertices V, directed edge E_d, and undirected edge E_u.

Table 3. *Cont.*

Notations	Meaning
$v_1e_1, v_2e_2, v_3e_3, \ldots e_nv_{n+1}$	Undirected sequence of walk.
$v_1\vec{e}_1, v_2\vec{e}_2, v_3\vec{e}_3, \ldots \vec{e}_nv_{n+1}$	Directed sequence of walk.
$A(G)$	Adjacent matrix.
e_1, e_2, e_3, e_4, e_5	Edges.
E_b	Bidirectional edge.
$E_b = (\{A, B\}, \{B, A\})$	$(\{A, B\}, \{B, A\})$ are the bidirectional edges.
$\left(\left\{\vec{A}, \vec{B}\right\}, \left\{\overleftarrow{B}, \overleftarrow{A}\right\}\right)$	Symbol of bidirectional edge.
$w(e)$	Weight of edge.
D_j	$n \times n$ Square matrix and j is the number of stages and n no. of a node.
D_0	D_0 is the initial stage when j = 0.

2.2. Definition and Preliminaries

In order to develop the entire paper, we have included some definitions and explanations in this section.

Definition 1. *(Semidirected graph) A graph* $G = (V, E_d, E_u)$ *is called a semidirect graph, where* V *is a set of nonempty sets of vertices,* E_d *is a set of directed edges, and* E_u *is a set of undirected edges.*

Example 1. *In Figure 1, let* V *is the set of nonempty vertices or node,* $V = (A, B, C, D)$*, and* $E_d = (e_1, e_2, e_3)$*,* $E_u = (e_4, e_5)$

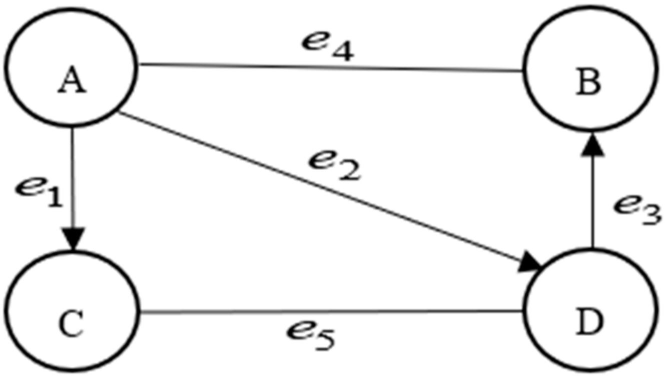


Figure 1. Semidirected graph.

Definition 2. *(Undirected walk) In a semidirected graph* $G = (V, E_d, E_u)$ *an undirected walk is a sequence of* $v_1e_1, v_2e_2, v_3e_3, \ldots e_nv_{n+1}$*, where* $v_1, v_2, v_3, \ldots v_{n+1}$ *are vertices of the semidirected graph and* $e_1, e_2, e_3, \ldots e_n$ *are undirected edges.*

Definition 3. *(Directed walk) In a semidirect graph,* $G = (V, E_d, E_u)$ *an undirected walk is a sequence of* $v_1\vec{e}_1, v_2\vec{e}_2, v_3\vec{e}_3, \ldots \vec{e}_nv_{n+1}$*, where* $v_1, v_2, v_3, \ldots v_{n+1}$ *are vertices of the semidirect graph and* $\vec{e}_1, \vec{e}_2, \vec{e}_3, \ldots \vec{e}_n$ *are directed edges.*

Definition 4. *(Adjacency matrix of a semidirected graph) An adjacent matrix is a tool that is used to represent the connection of vertices that are adjacent to each other in a finite graph. Therefore, in a finite graph, if there is a connection between a pair of vertices, then it is represented by binary value*

1, and if there is no connection between a pair of vertices, then it is represented by binary value 0. The finite graph can be represented by a square matrix $A(G)$ where $A(G) = \left(a_{ij,} b_{ij}\right)$ such that

$$a_{ij} = \begin{cases} 1, & \text{if there is an edge of } E_d \text{ between } i\text{-th and } j\text{-th vertices} \\ 0, & \text{if there is no edge between them.} \end{cases}$$

$$b_{ij} = \begin{cases} 1, & \text{if there is an edge of } E_u \text{ directed from } i\text{-thvertex to } j\text{-th vertex} \\ 0, & \text{if there is no edge between them.} \end{cases}$$

Definition 5. *(Edge) An edge (or link) of a network (or graph) is one of the connections between the nodes (or vertices) of the network. Incident edges are edges that share a node. Let,'e' be called the edge if and only if it is a link between a pair of nodes or vertices.*

Example 2. *From Figure 1, e_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.*

Definition 6. *(Shortest path) According to graph theory, the shortest path is one that connects two vertices of a graph such that the total weights of the individual edges are kept to a minimum.*

3. Conversion of Undirected Edges to Directional Edges

We have now reviewed how to convert a graph's undirected edges into directed edges.

Definition 7. *(Undirected edges to directed edge) An edge E_b is called the directional edge if and only if $\left(\left\{\vec{A}, \vec{B}\right\}, \left\{\overset{\leftarrow}{B}, \overset{\leftarrow}{A}\right\}\right)$ instead of $\{A, B\}$, where $E_b = (\{A, B\}, \{B, A\})$, the edge or link between the vertices A, B and B, A. Therefore, if two vertices or nodes are connected by a link or edge and if the edge has no direction, then one can convert the undirected link or edge into a directed edge. Figure 2 represents a semidirected graph with bidirectional edges.*

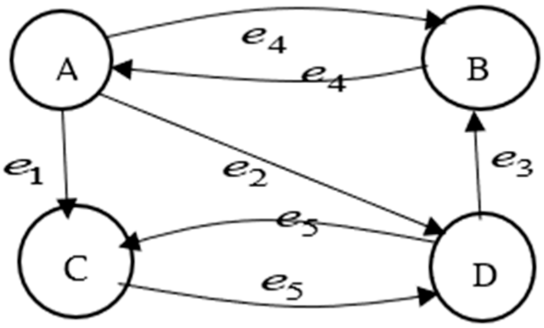


Figure 2. Semidirect graph (bidirectional edges).

Example 3. *Let, $(A, B) \in V$ and if e_4 be the undirected edge between vertices A, B, then the edge e_4 is called a bidirectional edge if and only if $\left(\left\{\vec{A}, \vec{B}\right\}, \left\{\overset{\leftarrow}{B}, \overset{\leftarrow}{A}\right\}\right)$ instead of $\{A, B\}$. In a similar manner, e_5 can be represented as a bidirectional edge (see Figure 2).*

Definition 8. *(Weighted semidirected graph) A semidirect graph $G = (V, E_{d,} E_u)$ with vertex set $V(G)$, directed edge set $E_d(G)$, and undirected edge set $E_u(G)$ is called a weighted graph if every edge e (directed or undirected) of graph G contains a numerical value $w(e)$. $w(e)$ is the weight of the graph.*

Example 4. *From Figure 3 , the weights of the graph are $\{4,7,3,5,5\}$, where$E_b(G) = \{5,5\}$ are sets of bidirectional weights and $E_d(G) = \{4,7,3\}$ are sets of directed edges.*

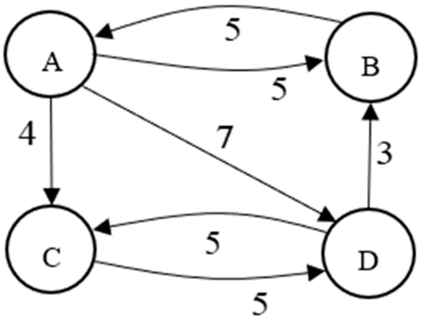


Figure 3. Weighted semidirected graph.

4. Different Types of Shortest Path

Numerous shortest path issues have been addressed in the literature with regard to theory. Some of these are as follows:

(i) Single-pair shortest path (SPSP) problem.
(ii) Single-source shortest path (SSSP) problem
(iii) Single-destination shortest path (SDSP) problem
(iv) All pairs' shortest path (APSP) problem

For the single-pair shortest path (SPSP) problem, we have mentioned the following points.

- The single-pair shortest path (SPSP) finds the shortest path either between a single pair of nodes or the shortest path between a particular pair of nodes.
- Find the shortest path from u to v.
- The well-known A * search algorithm has been used to solve the shortest path problem of a single pair.

We have identified the following for the single source shortest path (SSSP) problem:

- This is the shortest-path problem, in which the shortest path from a particular source node to all other remaining nodes has been computed.
- Find the shortest path from source s to all vertices $v \in V$.
- Dijkstra's algorithm and the Bellman–Ford algorithm are well-known algorithms used to solve the problem for finding the shortest path.

The single-destination shortest path (SDSP) problem can be viewed as follows:

- The shortest path problem that computes the shortest path from all vertices to one of the target vertices.
- This problem reduces to a single-source shortest path problem by reversing the direction of each edge in the graph.
- A well-known Dijkstra's algorithm has been applied to solve the problem of finding the shortest path to the destination.

The all-pair shortest path (APSP) problem can be summarized as follows:

- The shortest path problem that computes the shortest path between each pair of vertices.
- The Floyd–Warshall algorithm and the Johnson algorithm are known algorithms used to solve the problem of finding the shortest path for all pairs.
- Find the shortest path from u to v for all u.

From Table 4, we analyzed the different types of algorithms, and we found that the different types of shortest path algorithms are applicable to either undirected edges or directed edges. However, our proposed algorithm can apply to both the undirected and directed edges or semidirected edges. In real life, it is found that all networks are not only associated with the only undirected edge or directed edge, but sometimes, the network is also associated with both undirected and directed edges. Our proposed algorithm is very fruitful in evaluating the shortest path in that situation. It can also calculate the shortest path of all pairs of vertices and the time complexity $\Theta\left(n^3\right)$.

Table 4. Describing different SPAs (shortest path algorithms).

Algorithm	Applied for Undirected Edge	Applied for Directed Edge	Applied for Semidirected Edge	All Source	Time Complexity
Dijkstra's algorithm	Applicable	Applicable	Not Applicable	Not Applicable	$\Theta\left(\begin{array}{c}\lvert E\rvert+\\ \lvert V\rvert\log\lvert V\rvert\end{array}\right)$
Bellman–Ford Algorithm	Applicable	Not applicable	Not applicable	Not applicable	$\Theta(VE)$
Floyd–Warshall	Not Applicable	Applicable	Not Applicable	Applicable	$\Theta\left(n^3\right)$
Identifying the shortest path of a semidirected graph and its application	Applicable	Applicable	Applicable	Applicable	$\Theta\left(n^3\right)$

5. Algorithm for Finding Shortest Path of a Semidirected Graph

Here, we have proposed an algorithm for finding the shortest path of a semidirected graph. In our proposed algorithm, there are two phases. In Phase 1, we have converted unidirectional edge to bidirectional edge, and we applied the Floyd–Warshall algorithm in Phase 2.

Phase 1:
# Algorithm for conversion of unidirectional edge to bidirectional edge
Step 1: Initially choose a Semidirected graph $G = (V, E_d, E_u)$.
Step 2: Create a set of all unvisited set (V, E_d, E_u).
Step 3: select all E_u
Step 4: update all E_u in E_b
(i) $E_u = (v_{i,}v_j)$ then $E_u = \left(\{v_{i,}v_j\}, \{v_{j,}v_i\}\right) = E_b$ instead of $(v_{i,}v_j)$
(ii) $G = (V, E_d, E_u) \approx G = (V, E_d, E_b)$
Phase 2:
# Implementation of Floyd-Warshall algorithm for Bidirectional graph
Step 1: Initially take a graph $G = (V, E_d, E_b)$
Step 2: Set D_j as a $n \times n$ square matrix where j is the number of stages and n no of nodes.
Step 3: Compute D_0 when $j = 0$ and $D_0 = [d_{ik}]$

$$\text{where } d_{ik} = \begin{cases} d_{ik} \quad \text{if there is a directed edge between } i \text{ to } k \text{ node} \\ \\ \alpha \quad \text{if there is no directed edge between } i \text{ to } k \text{ node} \\ \qquad\qquad 0 \quad \text{if } i = k \end{cases}$$

Step 4: Repeat the following step for D_n
For $j = 1, 2, \ldots, n$ calculate the D_j and $D_j = [d_{ik}]$

$$\text{where } d_{ik} = \begin{cases} d_{ik} \qquad \text{if } i = k,\ i = j\ , k = j \\ \min(d_{ik}, d_{ij} + d_{ik}) \quad \text{otherwise} \end{cases}$$

Step 5: Return D_n and exit

5.1. Pseudocode for Conversion of Unidirectional Edge to Directional Edge

```
Algorithm bidirectional edge $B = (G, V, E_d, E_u)$
Initialized $V, E_d, E_u$
$V$ [source] = 0;
$E_d$ = directed edge;
$E_u$ = undirected edge;
$E_b$ = bidirectional edge;
if $e = E_d$ then print ("draw a directed edge from $(v_i, v_j)$");
else
```

```
for $v = i$ to j
$E_u = (\{v_i, v_j\}, \{v_j, v_i\})$
$E_u = E_b$
print ("draw a bi -directed edge between $(v_i, v_j)$");
```

5.2. Pseudo-Code for Shortest Path of a Semidirected Graph

```
Algorithm of shortest path $W(G, V, E_d, E_b)$
$n = row\,[W]$;
$D\,[0] = W$;
for $k = 1$ to n
for $i$= 1 to $n$
for $j$= 1 to $n$
do $\mathrm{d}^{(k)}{}_{ij} = \min\left(d^{(k-1)}{}_{ij}, d^{(k-1)}{}_{ik} + d^{(k-1)}{}_{kj}\right)$
return $D^n$
```

The Floyd–Warshall algorithm takes time $O(|V|^3)$, where $(|V|)$ is the number of vertices in graph G.

6. Numerical Example and Illustration

For the implementation of shortest path algorithm on a semidirected graph we have considered a network (see Figure 4). From Figure 4, we can write $E_d = (\{1,2\}\{2,3\}\{3,4\})$ and $E_u = (\{1,4\}\{2,4\})$. Now, we have applied our proposed algorithm. After execution of Phase 1, we have an equivalent bidirectional graph, which is depicted in Figure 5.

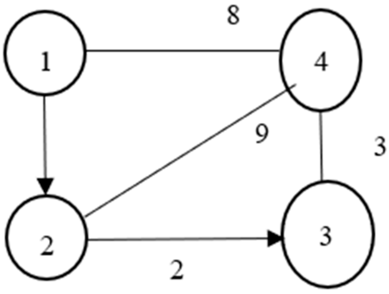


Figure 4. Semidirect graph of a road network system.

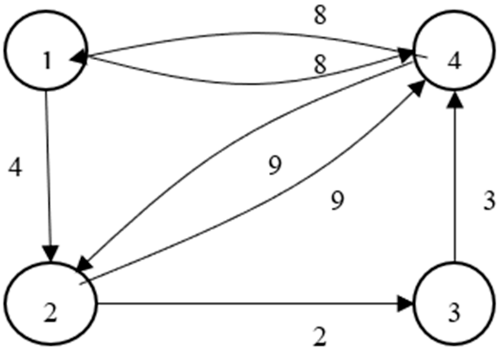


Figure 5. Equivalent bidirectional graph of semidirected graph.

After Phase 1, proceeded through Phase 2 of our proposed algorithm. Finally, we obtained the sequence of matrices as follows:

$$D^0 = \begin{bmatrix} 0 & 4 & \alpha & 8 \\ \alpha & 0 & 2 & 9 \\ \alpha & \alpha & 0 & 3 \\ 8 & 9 & \alpha & 0 \end{bmatrix}$$

D_0 defines the initial distance matrix of the bidirectional graph of a semidirected network, and this matrix represents the direct connection of the edge between two nodes. Undirected connections are represented by α.

$$D^1 = \begin{bmatrix} 0 & 4 & \alpha & 8 \\ \alpha & 0 & 2 & 8 \\ \alpha & 4 & 0 & 3 \\ 8 & 9 & \alpha & 0 \end{bmatrix}$$

D^1 defines the distance matrix via node 1, compares each node's distance with the distance of the direct node and via node 1, and replaces it with the minimum distance.

$$D^2 = \begin{bmatrix} 0 & 4 & 6 & 8 \\ \alpha & 0 & 2 & 8 \\ \alpha & 4 & 0 & 3 \\ 8 & 9 & 11 & 0 \end{bmatrix}$$

D^2 defines the distance matrix via node 2, compares the distance of each node with the distance of the direct node and via node 2, and replaces it with the minimum distance.

$$D^3 = \begin{bmatrix} 0 & 4 & 6 & 8 \\ \alpha & 0 & 2 & 5 \\ \alpha & 4 & 0 & 3 \\ 8 & 9 & 11 & 0 \end{bmatrix}$$

D^3 defines the distance matrix via node 3, compares the distance of each node with the distance of the direct node and via node 3, and replaces it with the minimum distance.

$$\text{And } D^4 = \begin{bmatrix} 0 & 4 & 6 & 8 \\ 17 & 0 & 2 & 5 \\ 11 & 4 & 0 & 3 \\ 8 & 9 & 11 & 0 \end{bmatrix}$$

D^4 defines the distance matrix via node 4, compares each node's distance with the distance of the direct node and via node 4, and replaces it with the minimum distance. D^4 gives the final shortest path between every pair of vertices.

A Real Case Study (Survey Results)

In this research, we have considered a case study which has been implemented on a road network (see Figure 6). This road is situated at Durgapur Express Way of the Paschim Burdwan district, West Bengal, India. This is a transport/road network system, and finding of shortest path of this network is based on finding a shortest path of a semidirected graph. For successfully execution of our proposed algorithm, we collected data from five places (the five places are listed in Table 5). These five places have been denoted as a node of a semidirected graph. The network diagram of this semidirected graph are shown in Figure 6. Descriptions of path type and distance are given in Table 6. Descriptions of path name with distance using Google Maps are shown in Figure 7.

Table 5. Description of node.

Node	Place Name
1	Durgapur City Center
2	Andal More
3	Andal Station Road

Table 5. *Cont.*

Node	Place Name
4	Ukhra Railway Station
5	Raniganj Panjabi More

Table 6. Description of path type and distance.

Path Type	Node	Path Name	Distance
One-way	1–2	Durgapur City Center—Andal More	11 km
One-way	2–5	Andal More—Raniganj Panjabi More	11 km
Bidirectional	3–2	Andal Station Road—Andal More	3 km
Bidirectional	4–2	Ukhra Railway Station—Andal More	8 km

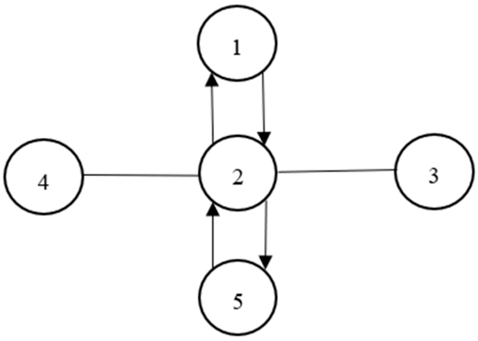
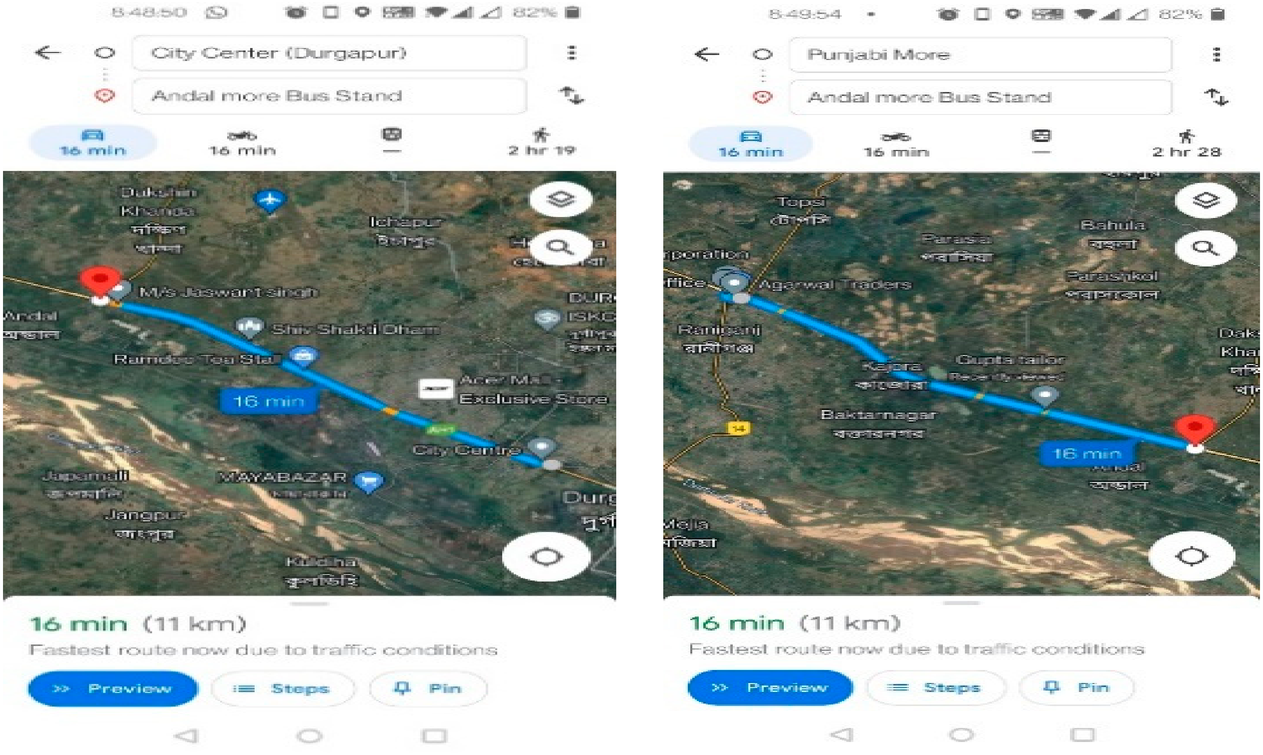


Figure 6. Semidirected network of a road network system.

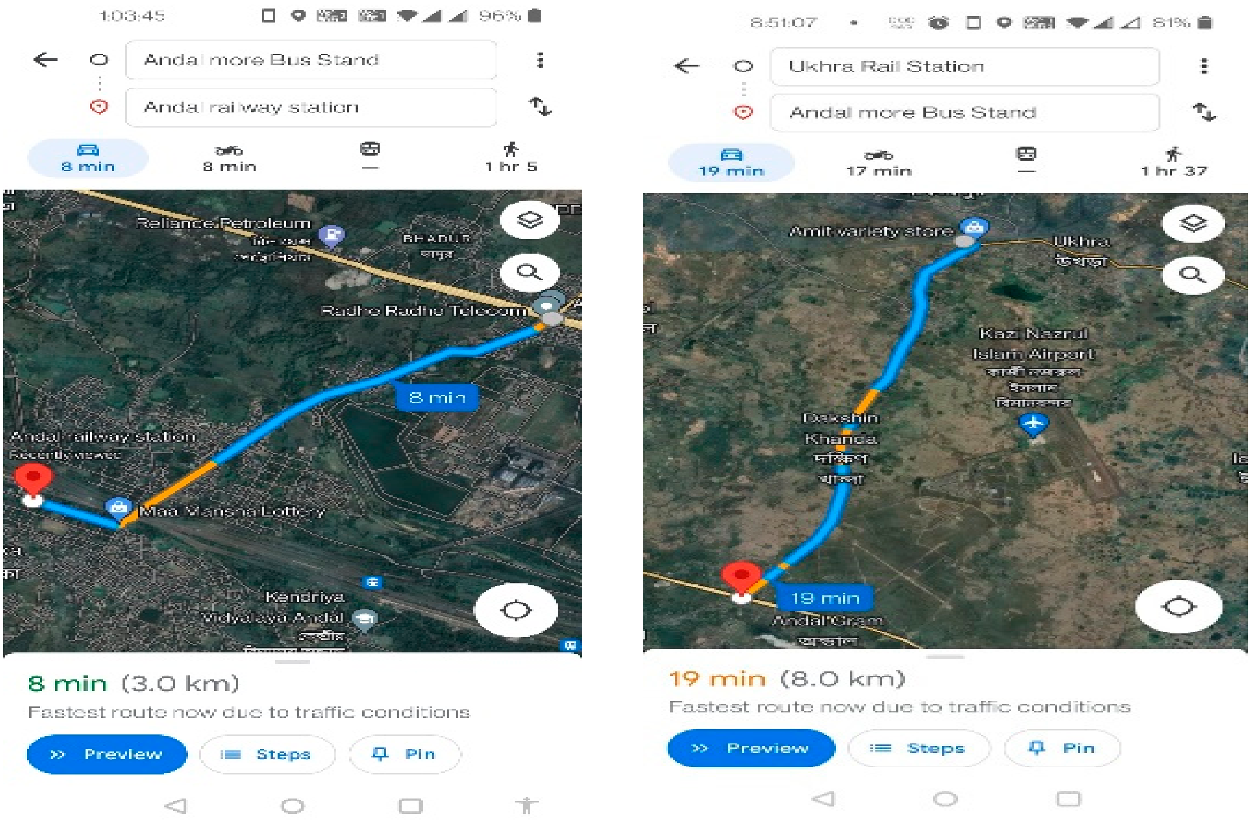

8:48:50
82%
City Center (Durgapur)
Andal more Bus Stand
16 min
16 min
—
2 hr 19
16 min (11 km)
Fastest route now due to traffic conditions

8:49:54
82%
Punjabi More
Andal more Bus Stand
16 min
16 min
—
2 hr 28
16 min (11 km)
Fastest route now due to traffic conditions

Durgapur City Center to Andal More (11 km)

Andal More -Raniganj Panjabi More (11 km)

Figure 7. *Cont.*



1:03:45 96%

Andal more Bus Stand

Andal railway station

8 min (3.0 km)

Fastest route now due to traffic conditions

8:51:07 81%

Ukhra Rail Station

Andal more Bus Stand

19 min (8.0 km)

Fastest route now due to traffic conditions

Andal Station Road to Andal More (3 km)

Ukhra Railway Station to Andal More (8 km)

Figure 7. Description of path name with distance using Google Maps.

Using Phase 1 of our proposed algorithm, we have obtained an equivalent bidirectional graph, which is shown in Figure 8.

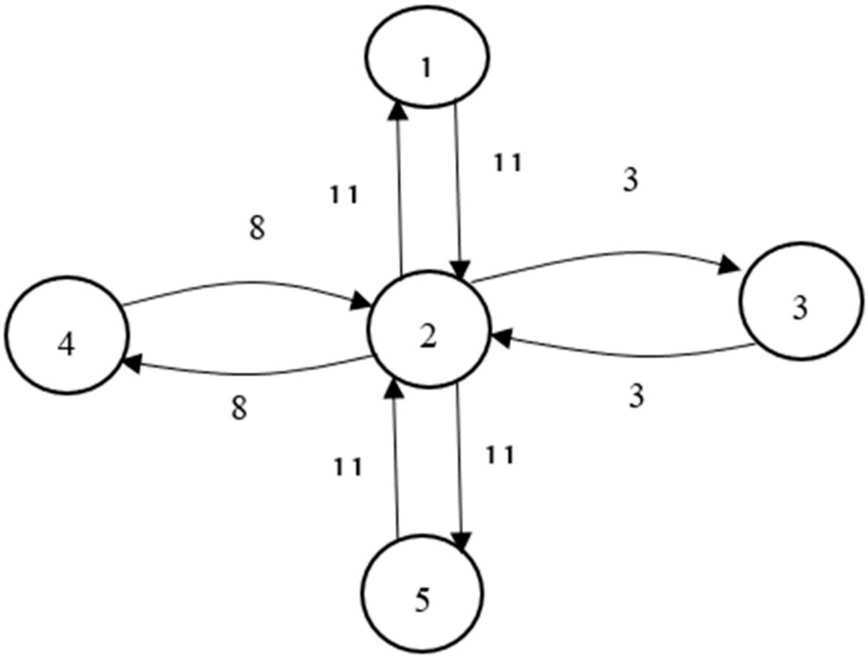


Figure 8. Equivalent bidirectional graph of a semidirected graph (Figure 6).

After Phase 1, we proceeded through Phase 2 of our proposed algorithm and obtained the sequence of matrices which is as follows:

$$D^0 = \begin{bmatrix} 0 & 11 & \alpha & \alpha & \alpha \\ 11 & 0 & 3 & 8 & 11 \\ \alpha & 3 & 0 & \alpha & \alpha \\ \alpha & 8 & \alpha & 0 & \alpha \\ \alpha & 11 & \alpha & \alpha & 0 \end{bmatrix}, D^1 = \begin{bmatrix} 0 & 11 & \alpha & \alpha & \alpha \\ 11 & 0 & 3 & 8 & 11 \\ \alpha & 3 & 0 & \alpha & \alpha \\ \alpha & 8 & \alpha & 0 & \alpha \\ \alpha & 11 & \alpha & \alpha & 0 \end{bmatrix},$$
$$D^2 = \begin{bmatrix} 0 & 11 & 14 & 19 & 22 \\ 11 & 0 & 3 & 8 & 11 \\ 14 & 3 & 0 & 11 & 14 \\ 19 & 8 & 11 & 0 & 19 \\ 22 & 11 & 14 & 19 & 0 \end{bmatrix} D^3 = \begin{bmatrix} 0 & 11 & 14 & 19 & 22 \\ 11 & 0 & 3 & 8 & 11 \\ 14 & 3 & 0 & 11 & 14 \\ 19 & 8 & 11 & 0 & 19 \\ 22 & 11 & 14 & 19 & 0 \end{bmatrix},$$
$$D^4 = \begin{bmatrix} 0 & 11 & 14 & 19 & 22 \\ 11 & 0 & 3 & 8 & 11 \\ 14 & 3 & 0 & 11 & 14 \\ 19 & 8 & 11 & 0 & 19 \\ 22 & 11 & 14 & 19 & 0 \end{bmatrix} \text{ and } D^5 = \begin{bmatrix} 0 & 11 & 14 & 19 & 22 \\ 11 & 0 & 3 & 8 & 11 \\ 14 & 3 & 0 & 11 & 14 \\ 19 & 8 & 11 & 0 & 19 \\ 22 & 11 & 14 & 19 & 0 \end{bmatrix}$$

where D^0 is the initial matrix of the network.

D^1 gives the shortest distance of all pairs' vertices via node 1, i.e., via Durgapur City center.

D^2 gives the shortest distance of all pairs' vertices via node 2, i.e., via Andal More.

D^3 gives the shortest distance of all pairs' vertices via node 3, i.e., via Andal Station road.

D^4 gives the shortest distance of all pairs' vertices via node 4, i.e., via Ukhra Station.

And D^5 gives the final shortest distance of all pairs' vertices via node 5, i.e., via Raniganj Panjabi More.

7. Conclusions

In this proposed paper, we have obtained the shortest path of a semidirected graph, and a real case study has been considered and solved. The proposed method for finding the shortest path of a semidirected graph is very useful for finding the shortest path of a transport network in any highway in the minimum time because all existing shortest path algorithms can evaluate the shortest path from either the directed or the undirected network. However, in our real-life road network system, networks are sometimes associated with unidirectional or directional edges. In these cases, it is very difficult to evaluate the shortest path. Hence, the proposed modified Floyd–Warshall algorithm can solve that problem.

The proposed algorithm is executed in two phases. First phase of the algorithm is able to convert all undirected edges into directed edges. Then, all undirected edges of the semidirected graph are converted into directed edges, and the semidirected graph is converted into a directed graph; then, the second phase of the proposed algorithm is able to identify the shortest path via Floyd–Warshall algorithm. Therefore, this proposed approach is a fruitful implementation for a modern road network system in which both directed and undirected roads have been found, and the proposal of this paper helps to determine the shortest path of a semidirected graph. It will be very helpful for automated driving systems (ADS) in the future. In future work, one could evaluate the congestion of the undirected edges or the bidirectional edges since traversal in both directions may lead to congestion.

Author Contributions: R.D., L.S., S.S., V.S. and T.S. formulated and studied the problem. R.D., L.S., S.S., V.S. and T.S. wrote the first draft of the manuscript. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Samanta, S.; Pal, M.; Mahapatra, R.; Das, K.; Bhadoria, R.S. A Study on Semi-directed Graphs for Social Media Networks. *Int. J. Comput. Intell. Syst.* **2021**, *14*, 1034–1041. [CrossRef]
2. Sporns, O. Graph theory methods: Applications in brain networks. *Dialogues Clin. Neurosci.* **2018**, *20*, 111–121. [CrossRef] [PubMed]
3. Ries, B.; De Werra, D. On two coloring problem of a mixed graph. *Eur. J. Comb.* **2008**, *29*, 712–725. [CrossRef]
4. Liu, J.; Li, X. Hermitian-adjacency matrices and Hermitian energies of mixed graphs. *Linear Algebra Its Appl.* **2015**, *466*, 182–207. [CrossRef]
5. Abudayah, M.; Alomari, O.; Sander, T. Incidence matrices and line graphs of mixed graphs. *arXiv* **2022**, arXiv:2205.05375.
6. Adiga, C.; Rakshith, B.R.; So, W. On the mixed adjacency matrix of a mixed graph. *Linear Algebra Its Appl.* **2016**, *495*, 223–241. [CrossRef]
7. Orlin, J.B.; Madduri, K.; Subramani, K.; Williamson, M. A faster algorithm for single source shortest path problem with few distinct positive length. *J. Discret. Algorithm* **1999**, *8*, 189–198. [CrossRef]
8. Thorup, M. Undirected single source shortest path in linear time. *J. ACM* **1999**, *46*, 362–394. [CrossRef]
9. Katerinis, P.; Tsikopoulosk, N. Edge-connectivity and the orientation of a graph. *SUT J. Math.* **2004**, *41*, 1–10. [CrossRef]
10. Bonato, A.; Delic, D.; Wang, C. The Structure and Automorphisms of Semi-directed Graphs. *J. Multi-Valued Log. Soft Comput.* **2010**, *27*, 161–173.
11. Wang, Z.H.; Shi, S.S.; Yu, L.C.; Chen, W.Z. An efficient constrained shortest path algorithm for traffic navigation. *Adv. Mater. Res.* **2012**, *356*, 2880–2885.
12. Jicy, N.; Mathew, S. Some new connectivity parameters for weighted graphs. *J. Uncertain. Math. Sci.* **2014**, *2014*, 1–9. [CrossRef]
13. Mohammed, A.M. Mixed graph representation and mixed graph isomorphism. *Gazi Univ. J. Sci.* **2017**, *30*, 303–310.
14. Sotskov, Y.N. Mixed Graph Colorings: A Historical Review. *Mathematics* **2020**, *8*, 385. [CrossRef]
15. Zeen El Deen, M.R.; Omar, N.A. Extending of Edge Even Graceful Labeling of Graphs to Strong r-Edge Even Graceful Labeling. *J. Math.* **2021**, *2021*, 1–19. [CrossRef]
16. Li, S.; Li, Y. Semi-dynamic shortest-path tree algorithms for directed graphs with arbitrary weights. *arXiv* **2021**, arXiv:1903.01756.
17. Singh, A.; Mishra, P.K. Performance Analysis of Floyd-Warshall Algorithm vs. Rectangular Algorithm. *Int. J. Comput. Appl.* **2014**, *107*, 23–27. [CrossRef]
18. Ramadiani; Bukhori, D.; Azainil; Dengen, N. Floyd-Warshall algorithm to determine the shortest path based on android. *IOP Conf. Ser. Earth Environ. Sci.* **2018**, *144*, 1–9. [CrossRef]
19. Triana, Y.S.; Syahputri, I. Implementation Floyd-Warshall Algorithm for the Shortest Path of Garage. *Int. J. Innov. Sci. Res. Technol.* **2018**, *3*, 871–878.
20. Brodnik, A.; Grgurovic, M.; Pozar, R. Modifications of the Floyd-Warshall algorithm with nearly quadratic expected-time. *ARS Math. Contemp.* **2021**, *22*, 1–22. [CrossRef]
21. Sahoo, L.; Sen, S.; Tiwary, K.S.; Samanta, S.; Senapati, T. Modified Floyd–Wars hall's Algorithm for Maximum Connectivity in Wireless Sensor Networks under Uncertainty. *Discret. Dyn. Nat. Soc.* **2022**, *2022*, 1–11.