Dijkstra's Algorithm on Semigraph

PRANJAL SEN[1], DHRUBA JYOTI NATH[2], ABDUR ROHMAN[1, a], SURAJIT KR. NATH[1,b,*], MOUSHUMI MITRA[3]

[1]Department of Mathematical Sciences,
Bodoland University,
Kokrajhar, 783370,
INDIA

[2]Department of Economics,
Kokrajhar Govt. College,
Kokrajhar, 783370,
INDIA

[3]Department of Information Technology,
Gauhati University,
Guwahati, 781014,
INDIA

[a]ORCiD: https://orcid.org/0009-0006-1353-6009
[b]ORCiD: https://orcid.org/0000-0002-2142-393X

Corresponding Author

Abstract: - The study of graph theory helps us in understanding the relationship between two nodes. Semigraphs are generalizations of graphs where the relationship occurs between more than two nodes. It provides solutions to issues with layout, matching, networking, optimization, etc. To determine the shortest route on a road or network, graph theory can be applied. Google Maps also uses graph theory, and it appears that semigraphs are a more realistic description of the same. In this paper, we investigate how semigraphs determine the shortfall distance using the Google Maps application. Here, we'll use a semigraphic model together with Dijkstra's Algorithm to determine the shortest route between Bodoland University, India, and Basugaon, Assam, India.

Key-Words: - Semigraph, Dijkstra's Algorithm, Google map, weighted semigraph, Shortest path, Road network.

Received: March 9, 2023. Revised: October 6, 2023. Accepted: December 15, 2023. Published: February 8, 2024.

1 Introduction

In today's digital age, we all use apps and applications to acquire various services that make our lives smooth and convenient. But have we ever thought about how these applications are executed or on what algorithms these applications are based? With few exceptions, the answer is no. To draw your attention to the use of mathematical algorithms, let us take a look at one of the most commonly used services, Google Maps, and understand the algorithm behind it.

The Google Maps, [1], application was first introduced in 2005 as a desktop solution that provides a bird's eye view of the world's road maps with GPS location and road navigation to help people easily move from one point to another. With its dynamic nature and ability to immediately respond to user needs, the Google Maps program has emerged as a crucial service in the modern world. The languages used to build the Google Maps framework are C++, JavaScript, XML, and Ajax.

Dijkstra's algorithm is one of the crucial Graph algorithms that Google Maps utilizes to determine the shortest path between two points: a source and a destination. Here, we'll talk about Dijkstra's algorithm and how it is used to determine the

Semigraph to find shortage distance from one point to another.

Why Semigraph?, [2], [3], [4], [5].
In graphs, [6], vertices (or points) enjoy more properties which is not true for edges (or lines). For example, consider the following statements:

1) Any number that mutually non-adjacent vertices may be adjacent to the same vertex. For edges, this is not true.
2) Similar to the concept of block graph B(G) of graph G, where every vertex (or point) represents a block of G, and two vertices (or points) in B(G) are adjacent if and only if, the corresponding blocks in G are adjacent. However, we do not have a concept of the graph where each edge (or line) represents a block of G.
3) Similar to the concept of a line graph of a graph G, we don't have a concept of a point graph where each edge (or line) represents a vertex (or point) of G.

In semigraph, all properties of vertices (or points) are equally enjoyed by edges (or lines).

The Dijkstra algorithm, [7], is a solution used by the Google Maps application for a shortest path problem in graph theory that can also be used in semigraph. It is a successful and productive algorithm that was proposed in the year 1956 and was published three years later. It is utilized to determine the shortest path between any two points. It selects the unvisited node with the lowest distance, then calculates the distance through it to each unvisited neighbor, updates the neighbor's distance if smaller, and marks the visited node when done with neighbors.

The purpose of this paper is to use the concept of semigraphs, [8], [9], in Dijkstra's algorithm to determine the shortest path between any two points in the Google Maps application.

2 Preliminaries

Definition 2.1: Semigraph, [2], [3], [4], [5].
A **semigraph** S is a pair of (V, X) where V is a non-empty set whose elements are vertices of S and X is a set of ordered n-tuples $n \geq 2$, called edges of S satisfying the following properties:

i. The components of an edge E in X are distinct vertices from V.
ii. Any two edges have at most one vertex in common.
iii. In semigraphs, two edges $E_1 = (e_1, e_2, \dots, e_m)$ and $E_2 = (u_1, u_2, \dots, u_n)$ are said to be equal if and only if
   a) $m = n$, and
   b) Either $e_i = u_i$ or $e_i = u_{n-i+1}$ for any $1 \leq i \leq n$.

   Clearly, the edge $(e_1, e_2, \dots, e_m)$ is the same as $(e_m, e_{m-1}, \dots, e_1)$.

Depending upon their positions in an edge, the vertices in a semigraph are divided into four types namely, **end vertices**, **middle vertices**, **middle-end vertices** and **isolated vertices**. The end vertices and isolated vertices are represented by thick dark dots, middle vertices are represented by small circles, and a small tangent is drawn at the small circles to represent middle-end vertices.

Example 2.1: Let $S = (V, X)$ be a semigraph where $V = (v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9)$ and $X = ((v_1, v_2, v_3, v_4); (v_2, v_5, v_6); (v_4, v_8); v_9)$ then

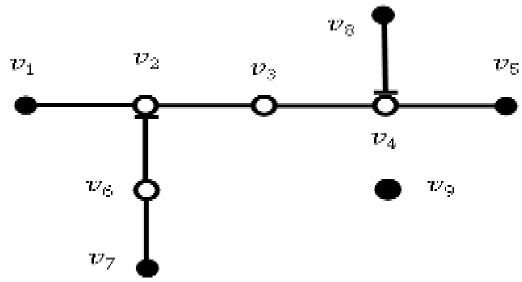

Fig. 1: Semigraph Example

Here from Figure 1, v_1, v_5, v_7, v_8 are end vertices, v_2, v_4 are middle-end vertices, v_3, v_6 are middle vertices, and v_9 is isolated vertex.

Definition 2.2: Adjacency of two vertices in a Semigraph, [2]
In a semigraph, if two vertices belong to the same edge then they are said to be **adjacent**. Again, two vertices are said to be **consecutively adjacent** if they are also consecutive in order. Similarly, two vertices are said to be **e-adjacent** if they are the end vertices of an edge.

Definition 2.3: Weighted Semigraph
A **weighted semigraph** is a semigraph in which each edge is assigned a numerical weight. Thus, it is a special type of semigraph (in which the labels are numbers taken to be positive).

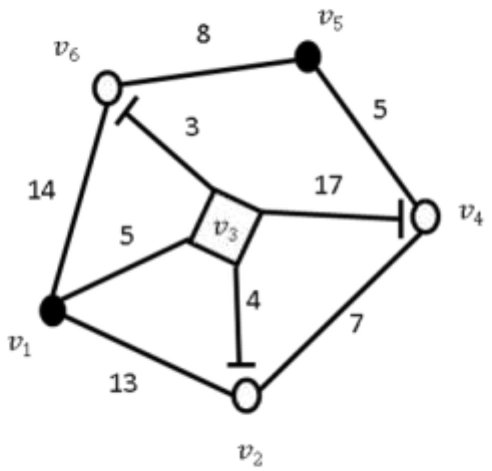

Fig. 2: Weighted Semigraph Example

Here from Figure 2, we get, 3, 4, 5, 7, 8, 13, 14, and 17 are the numerical weight assigned to its edges.

A graph is a mathematical model that can be applied to any system involving binary relations between objects. We can consider a city's road map as a graph in which the vertices are the important or prominent places of the city and two vertices are joined by an edge if they are along the road. But there can be other places of interest along the road. To represent such systems that involve a relationship that is not binary, there is a requirement for the generalization of graphs.

There have been many attempts to generalize this 'graph model': Semigraph and Hypergraph are such generalizations. Hypergraph, [6], is a generalization in which an edge may contain more than two vertices initiated by C. Berge. Semigraph, introduced in 2004, [2], [3], is a graph in which an edge contains two or more vertices. But the main difference between Semigraph and hypergraph is that in semigraph, the edges are ordered tuples whereas in hypergraph, the order of vertices occurring in an edge is irrelevant.

As in the road, the appearance of the order of places of interest is important; road maps are better represented by semigraph. Thus, one can suitably represent the road map problems by semigraph more effectively.

Definition 2.4: Optimal Routing, [2]
Let V be the set of places of human dwellings (such as towns, villages, cities, etc.) in a state, interconnected by road networks. Let us consider, in that state, a state transport corporation running its buses to various destinations. To minimize traffic and for the economy, the corporation can plan its route in such a way that any two-route map may be planned as a semigraph on the available or existing road network.

3 Shortest Path between Bodoland University and Basugaon

First of all, we draw a semigraph on the map between Bodoland University and Basugaon shown in Figure 3

Fig. 3: Map between Bodoland University and Basugaon

Semigraph of the map with weighted Semigraph (Figure. 4):

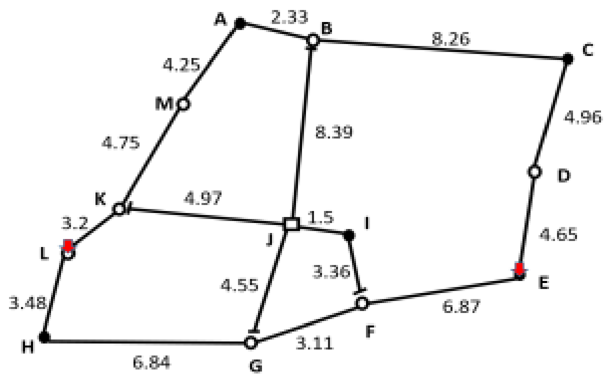

Fig. 4: Semigraph model map between Bodoland University and Basugaon with weight

Here, the distances between two nodes (in km) are found with the help of the Google Maps application. Also, the nodes **L** and **E** are respectively the positions of Bodoland University and Basugaon respectively whose shortest distance needs to be determined.

In the above weighted semigraph the edge sets are
(A, B, C); (C, D, E); (E, F, G, H); (B, J, G); (I, F); (I, J, K) and (A, M, K, L, H).
Also, in the above weighted semigraph,

Pranjal Sen, Dhruba Jyoti Nath,
Abdur Rohman, Surajit Kr. Nath, Moushumi Mitra

A, **C**, **E, H,** and **I** are the end vertices,
D, **J**, **L,** and **M** are the middle vertices,
B, **F**, **G,** and **K** are the middle-end vertices.

So, now we will follow Dijkstra's Algorithm mechanism to determine the shortest path between the nodes **L** and **E**.

The following steps are taken to find the shortest path between **L** and **E** as shown in Figure 5

STEP I: Mark node E with a distance of zero. After the nodes are named, record the distances in brackets ([]). For each node leading to E, we calculate the distance. We mark E as visited.

STEP II: Now, we mark node D with the smallest recorded distance as current. For each node leading to D (and not leading to a visited node), we determine the distance from the end. Since C is 4.96 km from D, then C is (4.96 + 4.65 =) 9.61 km from E. We mark D as visited.

STEP III: Now, node F has the smallest recorded distance from the end and we designate it as current. For each node leading to F (and not leading to a visited node), we determine the distance from the end.

i. Since G is 3.11 km from F, then G is (3.11 + 6.87 =) 9.98 km from E.
ii. Since I is 3.36 km from F, then I is (3.36 + 6.87 =) 10.23 km from E.

We mark F as visited.

STEP IV: Now, we mark node C with the smallest recorded distance as current. For each node leading to C (and not leading to a visited node), we determine the distance from the end. Since B is 8.26 km from C, then B is (8.26 + 9.61 =) 17.87 km from E. We mark C as visited.

STEP V: Now, we mark node G with the smallest recorded distance as current. For each node leading to G (and not leading to a visited node), we determine the distance from the end. Since H is 6.84 km from G, then H is (6.84 + 9.98 =) 16.82 km from E. We mark G as visited.

STEP VI: Now, we mark node I with the smallest recorded distance as current. For each node leading to I (and not leading to a visited node), we determine the distance from the end. Since J is 1.5 km from I, then J is (1.5 + 10.23 =) 11.73 km from E. We mark I as visited.

STEP VII: Now, we mark node B with the smallest recorded distance as current. For each node leading to B (and not leading to a visited node), we determine the distance from the end. Since A is 2.33 km from B, then A is (2.33 + 17.87 =) 20.2 km from E. We mark B as visited.

STEP VIII: Now, we mark the node J with the smallest recorded distance as current. For each node leading to J (and not leading to a visited node), we determine the distance from the end. Since K is 4.97 km from J, then K is (4.97 + 11.73 =) 16.7 km from E. We mark J as visited.

STEP IX: Now, we mark the node K with the smallest recorded distance as current. For each node leading to K (and not leading to a visited node), we determine the distance from the end. Since L is 3.2 km from K, then L is (3.2 + 16.7 =) 19.9 km from E.

STEP X: Now, we mark node H with the smallest recorded distance as current. For each node leading to H (and not leading to a visited node), we determine the distance from the end. Since L is 3.48 km from H, then L is (3.48 + 16.82 =) 20.3 km from E. Since this distance is longer than the previously calculated distance from E to L through K (which is through J), we do not replace it. We mark H as visited.

STEP XI: Now, we mark node A with the smallest recorded distance as current. For each node leading to A (and not leading to a visited node), we determine the distance from the end. Since M is 4.25 km from A, then M is (4.25 + 20.2 =) 24.45 km from E. We mark A as visited.

STEP XII: Now, we mark the node M with the smallest recorded distance as current. For each node leading to M (and not leading to a visited node), we determine the distance from the end. Since K is 4.75 km from M, then K is (4.75 + 24.45 =) 29.2 km from E. Since this distance is longer than the previously calculated distance from E to L through K (which is through J), we do not replace it. We mark M as visited. Hence K is also visited.

Since all the nodes are visited.

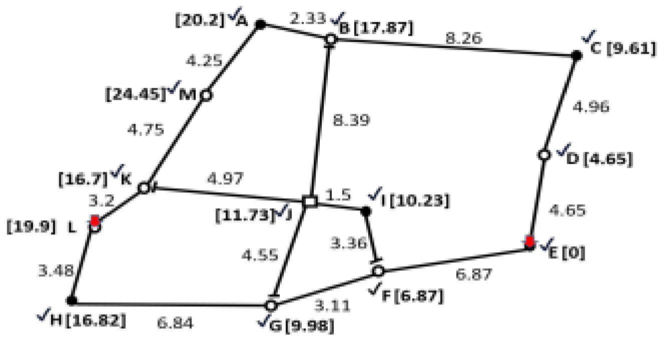

Fig. 5: Visited nodes

Therefore, the shortest path from E to L through K (which is through J) is found to be 19.9 km.

The shortest route between Bodoland University and Basugaon is given below (Figure 6).

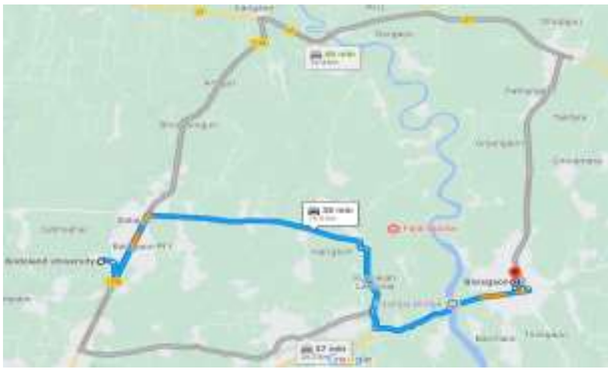

Fig. 6: Shortest Path

4 Conclusion

This work gives an insight into how the concept of Semigraph and Dijkstra's Algorithm is utilized to determine the shortest distance between Bodoland University and Basugaon with the help of the Google Maps application. The concept of Dijkstra's Algorithm allows the Google Maps application to calculate and display the shortest path between two points in a short amount of time. These are the strengths of this work. However, the weakness of this work is that it fails to consider parameters like time requirement, volume of traffic, weather conditions, etc.

5 Compliance with Ethical Standards

Conflict of Interest: The authors declare that they have no conflict of interest or other ethical conflicts concerning this research article.

References:

[1] P. Kanani, H. Mehta, P. Lande, *Google Maps*, *International Journal of Computer Applications*, Vol. 178-No. 8, May 2019.

[2] E. Sampathkumar, *Semigraphs and Their Applications*. *Academy of Discrete Mathematics and Applications*, India, 2019.

[3] E. Sampathkumar, "Semigraphs", *SERC Research Highlights*, New Delhi, 270-389, 2003

[4] S.S. Kamath, R.S. Bhat, "Domination in Semigraphs", *Electronic Notes in Discrete Mathematics*, Vol.15, 106-111, 2003

[5] B.Y. Bam, C.M. Deshpande, N.S. Bhave, "Line semigraph of a semigraph", *Korean Journal Adv. Studies Contem. Math.*, Vol.18(2), 2009

[6] C. Berge, *Graphs and Hypergraphs*, North Holand.

[7] Javaid, Adeel, "Understanding Dijkstra Algorithm", *SSRN Electronic Journal*, 2013, https://dx.doi.org/10.2139/ssrn.2340905.

[8] S. Gomathi, *Studies in Semigraphs and Domination*, Ph.D Thesis, Madurai KamarajUniversity, 2008.

[9] K. Bhattarai, A. Rohman, S.K. Nath, "On Transmission of COVID-19 in terms of Semigraph", *WSEAS Transactions on Systems and Control*, Vol. 18, 2023, Art. #29, https://doi.org/10.37394/23203.2023.18.29.

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors contributed in the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

No funding was received for conducting this study.

Conflict of Interest

The author has no conflict of interest to declare.