Dijkstra's Shortest Path Algorithm and Its Application on Bus Routing

Ruiting Chen*

Walnut High School, Walnut, 91789, United States.
**Corresponding author. Email: whsrchen327019@gmail.com*

ABSTRACT
The shortest path problem is common in computer science and has wide application in life. The shortest path algorithm is designed to solve these problems and the most classic and useful one is Dijkstra's shortest path algorithm which Dijkstra introduced in 1959. This paper presents and analyzes normal shortest path problems starting from the explanation of graphs and how they are divided into their practical examples. Common shortest path algorithms are also explained including their basic mechanism. Focusing on Dijkstra's algorithm, the paper analyzes its specific process and mechanism in a detailed way. Plus, the paper also analyzes the possibility of usage of Dijkstra' algorithm on public transit and roads. Considering the characteristic of bus transportation in Walnut, the paper concluded that Dijkstra's Algorithm can be used on Walnut bus transportation and set up a list of preconditions for the system to succeed. Based on the above work, a bus query system is implemented for the City of Walnut. "Walnut Bus Transportation Query System" serves as a reference to help residents choose the shortest road from the starting bus station to the ending bus station, thus saving time for the user.

Keywords: *Computer Science, Shortest Path, Dijkstra's Algorithm, Bus Routing, Algorithm*

1. INTRODUCTION

The Shortest Path Problem SPP has vast application and usage in real life. It took place in many industrial systems' production, organization, and management. For example, the subway is one of the vital essential travel tools. When we input the departure and destination in the subway ticket office or software, the mobile phone software will give us the optimal travel scheme with the least cost or the shortest time. The realization of the function of navigation software is a specific application of the shortest path algorithm. In management, making maximum profits out of the least cost is a Shortest Path Problem, and in production, deciding which step goes first that results in maximum efficiency is also a Shortest Path Problem.

The Shortest Path Problem is usually represented in each weighted network with vertices and edges, in which you are asked to find the shortest distance or path between the starting node and ending node. It has crucial usage in Computer science, Operations Research, and Geographic Information science.

This paper will focus on algorithms that solve the Shortest Path Problem and specifically on Dijkstra's Algorithm proposed by Dijkstra in 1959. This paper will analyze Dijkstra's Algorithm and realize java code for bus routing to find the shortest path. Based on those and public transport data from the City of Walnut, this paper develops a Walnut Bus Routing System, which finds the shortest bus path and serves as a reference resource for Walnut residents.

2. COMMON SHORTEST PATH PROBLEM AND ALGORITHMS

2.1. Shortest Path Problem in Reality

For example, if a passenger in the New York subway system wants to go from A to B destination, finding the shortest path and transfer route would be the shortest path problem related to time and distance.

For another example, during the postbellum industrial expansion in America, the cost of laying a railroad was the concern of many companies and even the government. Based on the terrain, soil, and structures like farms and cities, those railroad companies needed to estimate the cost between distances. Finding the cheapest

route from A to B is the shortest path problem related to cost.

In addition, shortest path problems could also relate to route arrangement and urban planning.

2.2. Shortest Path in Weighted Graphs

In a weighted graph G= (V, E), V represents the vertices or nodes or points in the graph that serve as a stop or station. E represents edges or sides like roads that link to those vertices. Each edge would have a corresponding weight representing the length it takes to travel between those nodes. It can be defined as L (Vi, VJ). For example, the length of travel from V1 to V3 could be L (V1, V2) + L(V2, V3).

The weight of each edge, depending on the different cases you are working on, could simply be the distance, cost, and time that it takes to travel.

In this kind of graph G, the shortest path problem means to find the smallest sum of length L possible to go from a starting node to an ending node. The sequence of nodes of this least length would be the shortest path of this shortest path problem.

The following figure 1, which we think of as a graph representing rooms, is a weighted graph. The weighted edges will represent hallways to each room, and weights will represent the distance in meters between each room. If a person wants to go from room 0 to room 3, he/she will consider the shortest distance. The shortest path of this case would be 0 -> 2 -> 1 -> 3. The shortest distance then would be 2+1+2=5 meters.

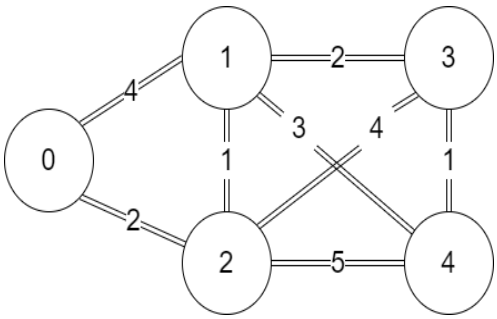

Figure 1: Weighted graph of room situation.

2.3. Shortest Path Algorithms

So, to solve such shortest path problems, as shown above, many computer scientists developed different algorithms to deal with these specific problems. Dijkstra ’ s Algorithm, which Dijkstra proposed in 1959, is a type of shortest path algorithm based on Breadth-first search and the concept of greedy algorithm [1]. In other words, It will check all the nodes, find the optimal in each check and produce a result after all the nodes are checked. Bellman-Ford, similar to Dijkstra Algorithm, is another algorithm to solve shortest path problems. It works on the principle of relaxation, and each edge will go through this process |V|-1 (number of nodes-1) times [2][3]. This brings more leniency during application in return for greater time complexity. Floyd-Warshall is another algorithm to solve shortest path problems. It specifically aims to find the shortest distance or path in random two nodes, and it is based on dynamic programming [4][5].

2.4. Dijkstra's algorithm

Out of all the different kinds of algorithms listed above, Dijkstra's algorithm is the one that is considered to be the most classic and commonly used algorithm. It is good in terms of time complexity which is O(|E|+|V|log|V|). Its produced result not only finds the shortest path from the start node to the end node but also finds the shortest path from the start node to every node in the graph. The basic idea of Dijkstra's algorithm is a loop of comparison. Set each node with a label of (dj, pj). dj is the length of the shortest path from starting node s to node j. pe is the previous node of the shortest path from s to j [6][7].

Initialize: s is set to be (ds=0, ps=empty); all other nodes are set to be di=∞, pi=? (undefined); Mark starting node s as visited and set k=s. All other nodes are unvisited.

Check the distance from all visited nodes k to its unvisited node e, set:

dj=min[dj,dk+lke]

lkj is the distance from node k to j.

From all the unvisited nodes, select the smallest i from dj:

di=min [dj,all the unvisited nodes j]

node i is included as one of the nodes in the shortest path and marked as visited.

From all the visited nodes, find node j* that is directly connected to node i. j* would be the previous node.

i=j*

Mark node i as visited, if all nodes are visited, the algorithm is done. Otherwise, set k=i, continue from 2).

The process of Dijkstra's algorithm is visualized by graphs below in Figures 2, 3 and 4, which solve the problem in figure 1. Nodes row represents nodes from 0-4, and Distance represents the current shortest distance from starting node to the ending node. Visited or not is marked by F(false) and T(true), and Parent refers to the previous node of the current node.

Nodes	0	1	2	3	4
Visited	F	F	F	F	F
Distance	inf	inf	inf	inf	inf

Parent	-1	-1	-1	-1	-1

Figure 2: Dijkstra's algorithm in initialization

Nodes	0	1	2	3	4
Visited	T	T	T	F	F
Distance	0	3	2	6	7
Parent	/	2	0	2	2

Figure 3: Dijkstra's algorithm in process

Nodes	0	1	2	3	4
Visited	T	T	T	T	T
Distance	0	3	2	5	6
Parent	/	2	0	1	1

Figure 4: Dijkstra's algorithm ended

3. APPLICATION OF DIJKSTRA'S ALGORITHM ON PUBLIC TRANSPORTATION

3.1. Bus Query System

Traffic and public transportation like buses and subways grow more critical and occupy a more significant proportion of our lives these days as cities become more crowded and overwhelming worldwide. People who cannot afford cars, place cars, or bypass the dense traffic tend to choose buses as their daily drivers. Though many bus companies post their information about bus stations on their website, this information is usually too fragmented and makes it hard for people to decide on their travel routes.

Dijkstra's algorithm can help with this situation and reference people which station they should travel in between. In this case, Dijkstra's algorithm could either use time or distance as the standard to measure. Stations and roads would be imported as the nodes and edges and assume people at each station have the possibility of changing or transferring buses. The database will then process using Dijkstra's Algorithm according to user inputs of starting station and ending station as they are the s and j. A rough design is shown in figure 5.

Referring to the above ideas, we saw the possibility of future development of many Bus query systems that can give users an accurate, fast, and relevant suggestion to choose their bus lines and stations.

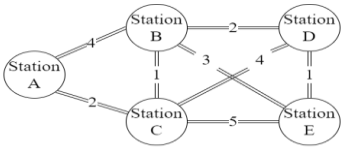

Figure 5: A rough design for the bus query system

3.2. Characteristics of Walnut Bus Transportation Route

Since Walnut is a relatively new city founded in 1959, its bus transportation routes are not as complicated as those of old international cities regarding the number of stations, bus lines, and duplicated naming caused by the legacies of their urban planning. Walnut's routes are less in number and less complicated. Here is how the data of Walnut's routes are handled.

Each line, station and distance data are made into a weighted graph to represent the walnut bus transportation route.

Due to the good city design of the City of Walnut, stations came in pairs for two-way purposes and were usually specified by direction words such as N, S, W, and E. To take advantage of this characteristic, bus routes and stations are all marked as unidirectional in our graph. This decreases our work when drawing the graph and reduces the amount of data to process while keeping the same level of precision [8].

Stations with similar or the exact name might appear in different locations. To handle this problem, we numbered the stations so the duplicate name would not affect the output.

3.3. Walnut Bus Transportation Query System

According to the above ideas, a Walnut Bus Transportation Query System is developed using Java programming language which is efficient for data operation [9]. A weighted graph of walnut bus stations is created during the process and serves as a reference to the system shown in figure 6. The system provides the shortest path and distance according to the input of starting station and ending station. It serves as a reference for traveling citizens who seek to reduce time.

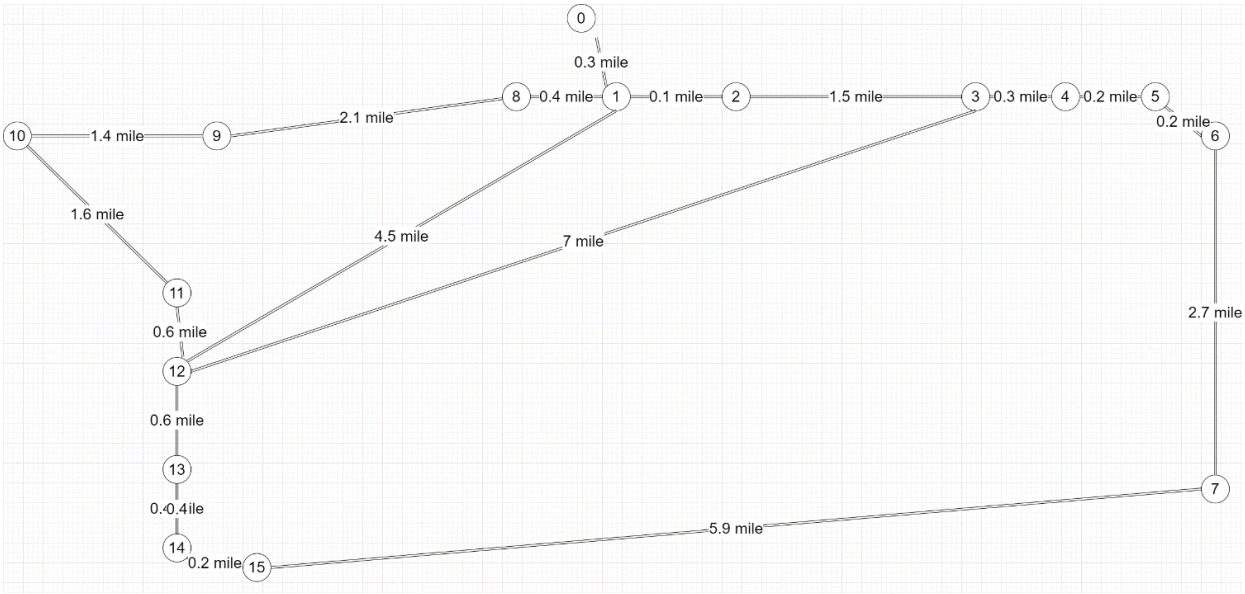

Figure 6: A weighted graph for walnut bus stations

The system's data of buses and stations are drawn from Google and the official sites of Foothill transit which are accurate and reliable.

3.4. System Testing

For example, if you would like to find the shortest path from Temple Ave and University Dr to Nogales St and Valley Blvd, you need to enter "3" in the "starting station" text box and "12" in the "ending station" text box. The system then would provide the shortest path through all the bus stations in the database as shown in figure 7.

```
====================================
0: Grand Ave and San Jose Hills Rd
1: Temple Ave and Grand Ave
2: Temple Ave and Mt SAC Way
3: Temple Ave and University Dr
4: Temple Ave and S Campus Dr
5: Temple Ave and Valley Blvd
6: Temple Ave and Pomona Blvd
7: Diamond Bar Blvd and Diamond Bar Park & Ride
8: Amar Rd and Sunset Bluff Rd
9: Amar Rd and Creekside Dr
10: Amar Rd and Westport St
11: Nogales St and Amanda St
12: Nogales St and Valley Blvd
13: Nogales St and Walnut Dr
14: Nogales St and Daisetta St
15: Colima Rd and Nogales St
====================================
Please input the start station:
3
Please input the end station:
12
The shortest path from Temple Ave and University Dr to Nogales St and Valley Blvd is: 3-->2-->1-->12
====================================
The shortest path from Temple Ave and University Dr to Nogales St and Valley Blvd is: 6.1M
```

Figure 7: Output of the system from Temple Ave and University Dr to Nogales St and Valley Blvd.

4. CONCLUSION

The shortest path problem is a typical type of problem in graph theory and computer science. Dijkstra's algorithm is also a classical algorithm used to solve the shortest path problem.

This paper analyzed Dijkstra's algorithm and described its mechanisms in graph theory. The paper also analyzed the possibility of Dijkstra's algorithm in the bus query system and introduced the characteristics of bus routes of the City of Walnut. A set of solutions were explicitly given for these characteristics.

Based on the above work, developed a "Walnut Bus Transportation Query System" that gives the shortest path from the starting station to the ending station. It serves as a reference for citizens to choose their traveling routes and reduce traveling time.

However, this paper also has many shortcomings. First, the practicability of Dijkstra's algorithm in large and international cities with more complicated bus routes needs further research and proving. Second, the time to execute Dijkstra's algorithm is long for large cities. Since cities like New York City, Paris, and Beijing have many more complicated bus routes than the City of Walnut, execution time under large amounts of data is slow and unacceptable. Third, the output of shortest path might not be realistic under more complicated bus routes due to the particularity of transit of different bus lines.

So, the direction of future research would focus on improving Dijkstra's algorithm as well as its practicability in large and international cities and improving the time complexity and efficiency of Dijkstra's algorithm. Since Dijkstra's algorithm has wide application, future research would seek and discuss application of Dijkstra's algorithm other than bus routing.

ACKNOWLEDGMENTS

Special thanks to Prof. V.G from UC Berkeley who introduced me to the world of algorithms during the winter break of 2021. He was passionate and nice to learning and welcomed every question I had on computer science. Also, thanks to Prof. Bin Liu from the Dalian

University of Technology and Mr. Nicholas Blackford from Walnut High School for concepts on computing programming in C++ and Java that helped me complete this project.

REFERENCES

[1] Dijkstra, Edsger W. "A note on two problems in connection with graphs." Numerische mathematik 1.1 (1959): 269-271.

[2]. Bellman, Richard. "On a routing problem." Quarterly of applied mathematics 16.1 (1958): 87-90.

[3] Ford Jr, Lester Randolph, and Delbert Ray Fulkerson. "Solving the transportation problem." Management Science 3.1 (1956): 24-32.

[4] Floyd, Robert W. "Algorithm 97: shortest path." Communications of the ACM 5.6 (1962): 345.

[5] Warshall, Stephen. "A theorem on boolean matrices." Journal of the ACM (JACM) 9.1 (1962): 11-12.

[6] Cormen, Thomas H., et al. Introduction to algorithms. MIT press, 2009.

[7] Yue Yang, and Gong Jianya. An Efficient Implementation of Shortest Path Algorithm Based on Dijkstra Algorithm. Diss. 1999.

[8] Chen Xiao-feng, Cai Xiu-yun, and Tang De-qiang. Shortest Path Algorithm Analysis and Its Application to Bus Route Query. Diss. 2001.

[9] Wang Shu-xi, and Wu Zheng-xue. (2012) Improved Dijkstra Shortest Path Algorithm and its Application. Computer Science, Vol.39 No.5: p223-228.