Universiti Teknologi MARA

MATHEMATICS AND STATISTICS

UNDERGRADUATE RESEARCH PROCEEDINGS 2025

UiTM CAWANGAN NEGERI SEMBILAN

Application Of Dijkstra's Algorithm To Find The Minimum Travel Time For Public Transport In A City

Nurin Aina Sofhia Binti Zuraidi[1,*], Noor Hidayah Mohd Zaki[1] and Noraimi Azlin Mohd Nordin[2]

*[1*12]Mathematical Sciences Studies, College of Computing Informatics and Mathematics, Universiti Teknologi MARA Cawangan Negeri Sembilan, Kampus Seremban, Persiaran Seremban Tiga/1, Seremban 3, 70300 Seremban, Negeri Sembilan.*

Corresponding: hidayahzaki@uitm.edu.my

Abstract

Public transportation systems are essential to sustainable urban mobility, particularly in densely populated cities. To achieve this, Dijkstra's Algorithm has been applied to two important objectives: to find the minimum travel time using Dijkstra's Algorithm and to recommend the shortest route for bus transportation services. The transportation network is represented as a graph with nodes representing some key landmarks and edges weighted by travel times and penalties including traffic signals and junctions. Manual calculations in the area of study has been calculated and in order to effectively implement the algorithm, Microsoft Excel and Open Solver were used to provide an expandable, user-friendly platform for computational analysis and visualization. Based on the finding, the shortest path is achieved in 9.30 minutes starting from Node 3 to Node 10. Overall, the results show that Dijkstra's algorithm is a simple yet competent approach in finding the shortest path for bus services within a short time period.

Keywords*:* Transportation, Dijkstra's Algorithm, Microsoft Excel and Open Solver

Introduction

Nowadays, discussing the issue of finding the quickest path in daily life is highly fascinating. As the world's economy grows and the middle class is becoming more prevalent in places like China and India, it is anticipated that the number of vehicles would also rise globally (Indrajaya et al., 2015). Even if the number of transports is growing and the growing middle-class and upper-class population has a good effect, there has not been enough development of vehicle tracking systems, which are crucial components of transportation networks (Indrajaya et al., 2015). Numerous factors, such as economy and safety in private automobiles, public transit systems, commercial vehicles, and so on, will benefit greatly from the deployment of tracking systems on the mode of transportation.

One of the most popular forms of transportation in cities is public transportation. Due to the numerous benefits that public transportation offers its customers, a sizable portion of city residents as well as tourists regularly utilise this urban transportation service. Firstly, compared to other modes of transportation, it is more cost-effective. Furthermore, in some situations such as peak hours, using a taxi may be quicker and less stressful than utilising a personal vehicle. As a result of the large number of individuals who regularly use public transport, multiple route-recommending applications have been developed across various platforms.

One of the common public transportation used are bus services. In Malaysia, this service is considered important and widely used throughout the country. It provides accessible, affordable services for all sections of society, especially for those living in urban and suburban areas. As a flexible form of public transport, buses can travel on various routes and destinations, including areas that may not be accessible by other types of public transport such as trains or MRT.

The process of figuring out the optimal path for a car to take, either before or during travel, to get to its destination is known as path planning. This specific application of the shortest path issue in automobile navigation systems is based on the topological structure of the road network. Bogdan (2015) said, Edsger Dijkstra, a computer scientist, developed Dijkstra's algorithm in 1959 which the idea is to link graph's shortest path problem (SPP). Until the criteria are met, this algorithm will iterate over each point on the graph and determine the shortest path between them. Furthermore, Bogdan (2015) said in practice, the Dijkstra's algorithm determines the shortest routes from the starting point to the other places on a map in addition to the shortest path between the start and finish points. The SPP is often used to minimize travel time and help individuals choose the best route when traveling from one point to another. Dijkstra's algorithm is one method for assisting and resolving the issue of helping individuals choose the shortest route.

One of the common issues arise is road congestion in crowded areas, especially in cities. It is largely driven by the rapid growth of the automobile industry and people's increased purchasing power. The rising number of vehicles, combined with the heavy reliance on private cars, is particularly noticeable during peak hours when people go out for lunch or running daily chores. This surge in vehicle usage often leads to traffic jams, and causing delays for many travelers. One of the solutions to address this issue is by using public transport services, which are highly encouraged by the government, such as buses. However, it is not easy to encourage people to use public transportation. Thus, the use of Dijkstra's algorithm is hoped to give idea to people on how to get to their destination in minimum travel time by using public transportation. This process is represented as a road network, where the edges are connected with nodes.

Based on the process, it is hoped that Dijkstra's algorithm resolves navigational issues. This may also save time and improve decision-making. This study offers a novel framework that enables drivers to efficiently manage their travel based on traffic situation. The findings demonstrate that, given the traffic count, drivers may find both the shortest and alternate paths for a given route. According to experimental data, Makariye (2017) said the algorithm performs better over a range of parameters and responds to the shortest path problem very quickly.

Referring to Bogdan (2015), with applications in timetable information systems, online routes, and navigation systems, algorithm engineering's study of route planning in transportation networks is important. In a weighted graph, the shortest path between nodes is found using Dijkstra's algorithm. The objective is to design a highly efficient parallel application while taking the density and structure of the graph into account. Sequential techniques might be more effective, although the effectiveness of the suggested approach depends on the density and structure of the graph, considering both long and short roads.

According to (Khaing et al., 2018), due to pollution, time wastage, and the usage of hydrocarbon fuels, the traffic issue in the Yangon area lowers the quality of life for its residents. The issue still exists despite an increase in the number of roads. An algorithm known as the shortest path was created to determine the optimal route while taking into account the current traffic situations and possible detours. The goal of Dijkstra's Algorithm is to identify the shortest path between a source and a destination while solving the SPP in graphs.

According to Ray (2022), vehicle monitoring is crucial for fleet management, mass transit management, public transportation management, and personal safety. Users frequently utilize Dijkstra's algorithm to find the shortest path. Real-time location tracking in a GIS application was created to enhance public transport. Drivers no longer need to drive across town because this system assigns passenger groups to taxis.

As highlighted in Chen (2022), public transport, such as buses and subways, is increasingly important as cities get larger. Using time or distance as a baseline, Dijkstra's algorithm can assist passengers in choosing which station to go between. Assuming that passengers can switch or transfer buses at each station, the algorithm imports roads and stations as nodes and edges. Dijkstra's algorithm is applied to the database processing.

However, the algorithm may be tedious and complicated to be calculated manually if large data is involved. Thus, a software needs to be used to manage the algorithm and to ease in calculations. Microsoft Excel is a versatile and widely accessible tool used for handling complex computations, data analysis, and optimization tasks. Its user-friendly interface and built-in functionalities, such as Visual Basic for Applications (VBA), make it an excellent platform for solving real-world problems. For instance, tools like the VRP Spreadsheet Solver utilize. Erdogan (2017) said, Excel to address routing and scheduling issues by integrating VBA with GIS data, enabling users to calculate travel distances, optimize routes, and visualize solutions dynamically.

On the other hand, Open Solver by Microsoft Excel extends the computational capability of spreadsheet modelling by allowing the use of sophisticated optimization algorithms. Unlike Excel's built-in Solver, Open Solver can handle larger datasets and more complex models, using powerful external solvers like Gurobi, CBC, and CPLEX to compute optimal solutions efficiently.

This paper only focuses on the application of Dijkstra’s algorithm in finding the shortest path of the public bus transportation. The objectives of this paper are firstly to find the minimum travel time from SOGO Kuala Lumpur to Bahtera Sri Kandi Sdn. Bhd. Secondly, to recommend the shortest route for bus transportation services. In the end, minimizing total travel time will contribute towards minimizing the waiting time of the bus passengers.

Methodology

Data Collection

A study was conducted to understand the application of Dijkstra's Algorithm in finding the shortest travel time. This study focused on a simplified network of public transport routes in Kuala Lumpur, using buses as the primary mode of transportation. The analysis was conducted manually at first to allow a deeper understanding of the how the algorithm’s work. This study serves as a foundation for applying the algorithm to a larger dataset, ensuring the methodology is accurate and efficient before expanding to cover more extensive transport networks.

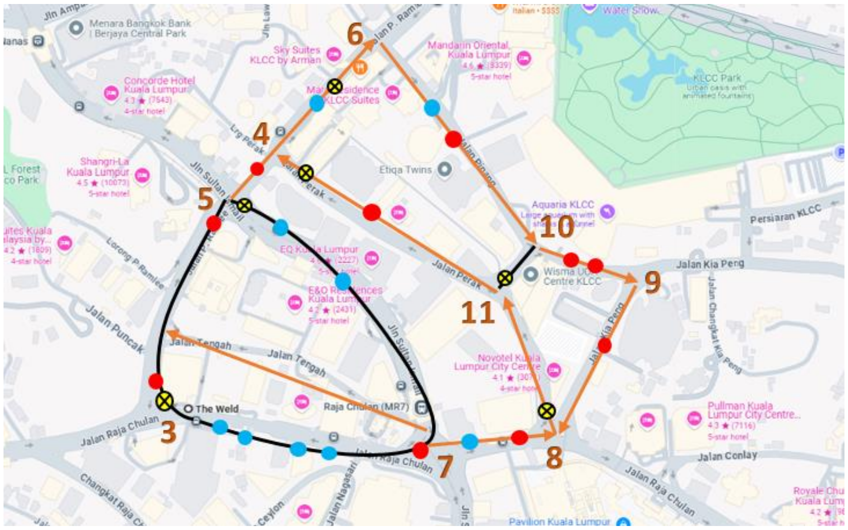

Figure 1: Route network for the area under study

Figure 1 shows the route of respective study. There are nine nodes with different penalties for each route. The starting node is denoted as node 3 and the destination node is node 10. As can be seen in Figure 1, the orange line indicates the one-way direction route. For instance, node 6 to node 10 are indicated by the directional arrows on the edges. Other than that, the routes can be travelled in two-ways direction. Based on this, the algorithm only focusing on feasible routes as set in the objective which is to achieve the minimum travel time. Next, Table 1 listed the description of the nodes. The road names are converted to numbers and on the whole, nine nodes are involved in this study.

Table 1: Description of the nodes involves in area of study

Legend	Description
3	The Weld The Weld
5-4	Jalan P.Ramlee
7	Jalan Raja Chulan
6	Jalan Pinang
8-11	Jalan Perak
9	Jalan Kia Peng
10	Bahtera Sri Kandi Sdn.Bhd

Table 2 describes the legend used in Figure 4 until Figure 9 to show how the Dijkstra's algorithm work to find the minimum travel time by manual calculation. Penalties are presented the same colour as in Figure 1.

Table 2: Description of legends involves in area of study

Legend	Description
3,4,5,6,7,8,9,10,11	Nodes
→ (black)	Edges two way route
→ (orange)	Edges one way route
● (blue)	Nodes
● (green)	Possible Nodes
● (yellow)	Confirmed Nodes

This study also comes with the assumption that this data:

- Captured on a weekday with typical and normal working hours.
- There are no accidents or construction sites along the route.
- The weather is good.
- The transport is in a good condition.
- The driver is knowledgeable about the roads and will keep to the posted speed limit.
- The data includes the following penalties such as traffic lights, traffic jams, junctions and road bumps.
- The roads taken are main roads only.
- Every visiting sequence has a set destination
- The calculation will begin from The Weld and stop at Bahtera Sri Kandi Sdn. Bhd.

Develop Travel Network

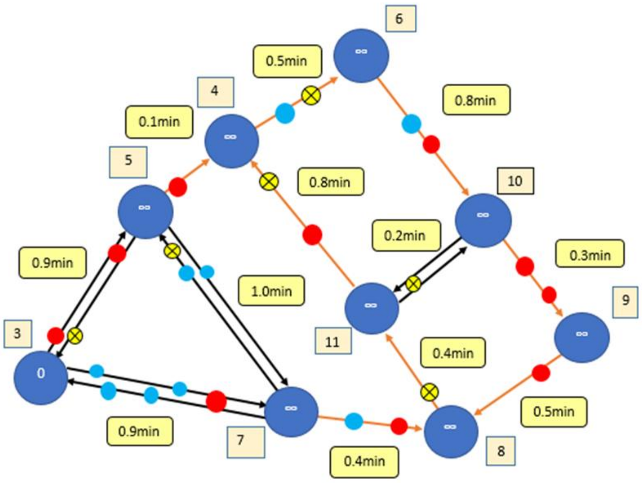

Figure 2: Network for area under study

Based on Figure 2, the network shows a directed graph for one-way journey, with nodes connecting the edges. The weights between the nodes is in minutes. The starting point is node 3 and the destination node is node 10. The nodes and edges have been carefully constructed to represent actual travel limitation, allowing it to compute the shortest journey durations with accuracy and efficiency.

Table 3 presents the travel times between one nodes to another nodes in a network, with each row representing a route from a starting point to an ending point. The table provides five main information which is the starting node, the ending node, travel time (excluding penalties), penalties (additional time caused by factors such as traffic lights, junctions, or bus stops), and the total travel time (a combination of travel time and penalties). For example, traveling from node 3 to node 5 takes 0.9 minutes as the base travel time, but with an additional penalty of 2 minutes, the total travel time becomes 2.9 minutes. This table was calculated manually by inputting values based on collected data, which are calculated manually using the Dijkstra's algorithm to determine the shortest travel time. The calculation of the total travel time with penalties in Table 3 is explained in detail in the next section.

Table 3: Travel Time between nodes in the study

Starting Point	Ending Point	Travel time (minutes)	Penalties (minutes)	Total travel times (minutes)
3	5	0.9	2.0	2.9
3	7	0.9	3.0	3.9
5	7	1.0	3.0	4.0
7	5	1.0	2.5	3.5
5	4	0.1	1.0	1.1
4	6	0.5	1.5	2.0
6	10	0.8	2.5	3.3
7	8	0.4	3.0	3.4
8	11	0.4	2.0	2.4
11	4	0.8	1.5	2.3
11	10	0.2	0.5	0.7

Finding the Shortest Path using Dijkstra's Algorithm

R is a collection of travel nodes initially. R is equal to $\{A\}$. Let node A be the source node and the starting point. $P(v)$ is the value of travel time from source node A to node v for all nodes v that are not in R. Each edge in the graph connection from nodes s to nodes v, has the same value of travel time, $P(v)$.

$P(v) = Q(A, v)$, if the nodes v and the source are connected directly.
$P(v) = \infty$, otherwise.

Therefore, the equation is:

$$D(s,v) = Min\left[P(v), P(w) + Q(w,v)\right] \tag{1}$$

where,

- $s =$ source node, where $s = A$
- $D(s, v)$ is the travel distance from node s (source node) to node v.
- $P(v)$ is the value of travel distance from source node 3 to node v for all nodes v that are not in R.
- $P(w)$ is the minimum value of travel distance.
- $Q(w, v)$ is the link cost from node w to node v if two nodes are connected.

In Dijkstra's Algorithm, the shortest path between the starting node and every other node is be determined. Longer travel time are skipped during the modification in order to maintain the shortest path from the starting point. The following steps are provided:

1. All nodes start with "infinite" distance; initialization of the initial node by 0.
2. Label the beginning node's distance as permanent, while all other distances as temporarily.
3. The initial node is active.
4. Compute the temporary distances of all neighboring nodes of the active nodes by adding its distance with the weights of the edges.
5. Update distance and set the node as an antecessor if the measured distance of the node is less than that of the current node. This step is also regarded as an update and is the principal concept of Dijkstra's Algorithm.
6. The minimum temporary distance setting of node as active. Mark the distance as permanent.
7. Redo step four to seven until there are no permanent distance nodes exist, which still have temporary distances for neighbours.

Steps in Dijktra's Algorithm for Finding Minimum Travel Time

In this section, the details of calculating the minimum travel time using the Dijkstra's algorithm is explained deliberately. The manual calculation is hoped to give better understandings in how the algorithm actually works. The procedure will be carried out throughout all the nodes and the algorithm will only stop until all the nodes have been processed. The calculation starts with step 1 as referring to Figure 3.

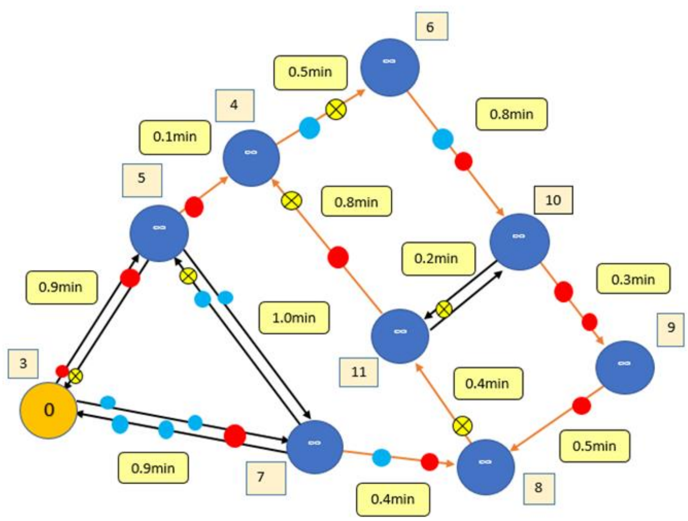

Figure 3: Node 3 is updated in orange

Step 1: Based on Figure 3, orange colour indicates the selected starting node. In this preliminary study, node 3 is set as a starting node. Starting node value is 0. From here, ∞ is assigned to all other nodes. The source node in this study is $R = \{3\}$.

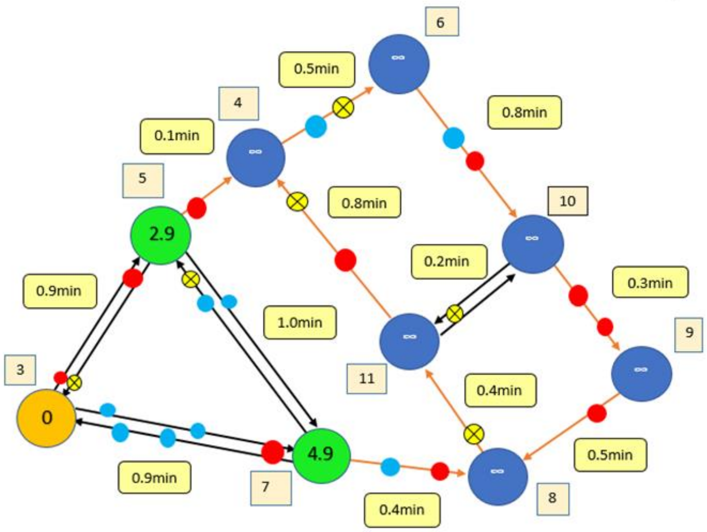

Figure 4: Selecting next possible shortest path from node 3

Step 2: The orange node has been recorded as being visited and will not be rechecked. Based on Figure 4, the next possible shortest path from node 3 in this stage can be either node 5 or node 7, with $P(3) = 2.9\ minutes$ and $P(7) = 4.9\ minutes$, respectively. By comparing the two values, node 5 is likely to be chosen as the current node. The time may be longer if node 3 to node 7 is chosen as the next current node. Thus, $R = \{3, 5\}$ is the current node with a total time of 2.9 minutes with two road penalties. The calculation between node 3 to 5 and node 3 to 7 is explained as follows:

Manual calculation from node 3 to node 5:

As in Figure 2, there are two penalties from node 3 to node 5. Each penalty is equal to additional one-minute travel time. Without the penalties, total travel time is only 0.9 minutes. However, in real life situations, vehicles need to follow strictly the rules abide by the authorities to ensure smooth driving to the destinations.

$\boldsymbol{If, R = \{3, 5\}}$
$Road\ penalties = 1\ minute + 1\ minute$
$Penalties = 2\ minutes.$
$P(w) = 0.9\ minute + 2\ minutes$

Thus, the current value from node 3 to node 5 is equal to 2.9 minutes. Next, the current travel distance will be updated to $D(3, 5)$. Denotes that the infinity symbol means that there are still nodes to be processed to reach the ending node.
$D(3, 5) = [2.9, (0.9 + 2) + \infty]$
$D(3,5) = 2.9\ minutes$

Hence, current shortest path from node 3 to node 5 is denoted by $R = \{3,5\}$ which equals to 2.9 minutes

Next, comparing the value with manual calculation from node 3 to node 7:

Now in Figure 2, driver may choose either go to node 5 or node 7. Previously, the calculation shows result from node 3 to node 5. Now there are 4 penalties involved in node 3 to node 7. Hence, additional four-minutes will be added up with the current travel time 0.9 minutes.

$\boldsymbol{If}, \boldsymbol{R} = \{\mathbf{3}, \mathbf{7}\}$
$Road\ penalties = 1\ minute + 1\ minute + 1\ minute + 1\ minute$
$Penalties = 4\ minutes.$
$P(w) = 0.9\ minute + 4\ minutes$

Thus, the total travel time is now 4.9 minutes. As can be seen, it takes two minutes longer compared to the other node selection. Therefore, the current travel distance is updated to $D(3,7)$ with the total travel time of 4.9 minutes.

$D(3,7) = [4.9, (0.9 + 4) + \infty]$
$D(3,7) = 4.9\ minutes$

Step 3: As a conclusion, the next selected shortest path is updated to node 5. Now node 5 is colored orange indicates that the node has been chosen as current node.

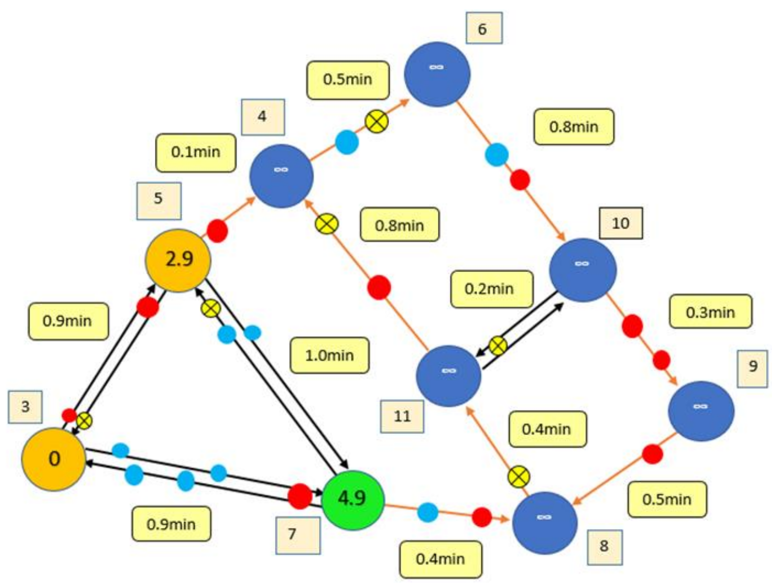

Figure 5: Node 5 updated as selected node (orange colour)

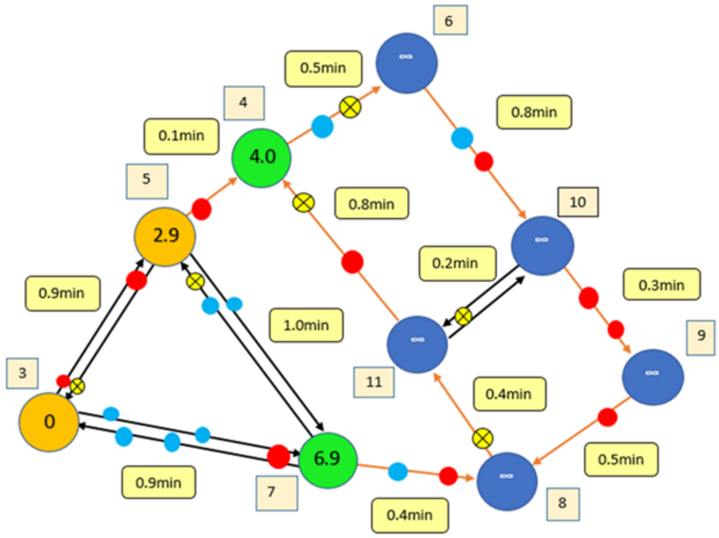

Figure 6: Node 5 to possible next node for travel time

Step 4: The same step is repeated again to find the next possible shortest path from node 5. Figure 6 shows that node 4 and node 7 is the next possible node.

In this situation, there is one road penalty from node 5 to node 4 and three penalties from node 5 to node 7. Now, the travel nodes R is updated to either $R = \{3,5,4\}$ and $R = \{3,5,7\}$. The calculation is explained as follows:

Manual calculation from node 5 to node 4.

$\boldsymbol{If, R = \{3,5,4\}}$
$Road\ penalties = 1\ minutes,$
$P(w) = 0.1\ minute + 2.9\ minutes + 1\ minute$

The calculation of $P(w)$ represent that 0.1 minute is the direct travel from node 5 to node 4, 2.9 minutes is the current travel time value and 1 minute is the penalty involved. Hence, the total travel time is now updated to $D\{3,5,4\}$.

$D\{3,5,4\} = Min[4.0, (0.1 + 1) + 2.9]$
$D\{3,5,4\} = 4.0\ minutes$

On the other hand, the manual calculation from node 5 to node 7 is as follows:
$\boldsymbol{If, R = \{3,5,7\}}$
$Road\ penalties = 1minute + 1minute + 1\ minute,$
$P(w) = 2.9\ minute + 1\ minute + 1minute + 1\ minute + 1\ minute$

The calculation of $P(w)$ now represent that 1 minute is the direct travel from node 5 to node 7, 2.9 minutes is the current travel time value and 3 minutes are the penalty involved. Hence, the total travel time is now updated to $D\{3,5,7\}$.

$$D\{3,5,7\} = Min[6.9, (1.0 + 3) + 2.9]$$
$$D\{3,5,7\} = 6.9\, minutes$$

ELSE,

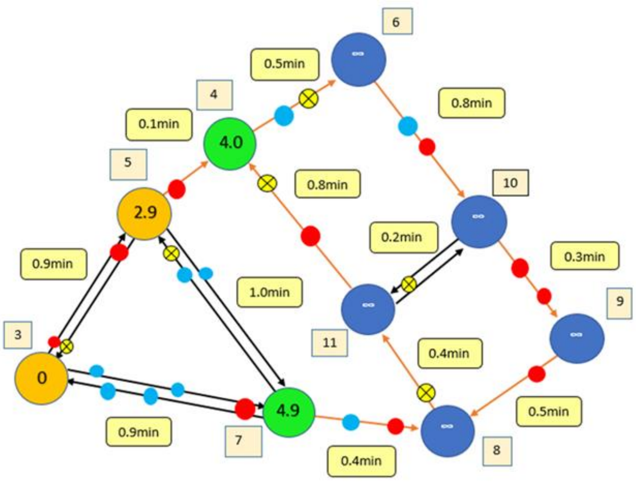

Figure 7: Node 3 to node 7 for another possible shortest node

Step 5: Since Figure 7 is still not selected as the shortest path, it can still be considered as the next possible routes. Earlier in step 3, the travel time from node 3 to node 7 is 4.9 minutes which is shorter than $D\{3,5,7\}$. Due to this, the current shortest path is updated to node 3 to node 7 which is 4.9 minutes. It is still possible since node 3 to node 7 is not selected yet and it can still be processed as possible minimum travel time. There are three choices of next shortest path that are $D\{3,5,4\} = 4\, minutes$, $D\{3,5,7\} = 6.9\, minutes$ or $D\{3,7\} = 4.9\, ,minutes$. Based on the choices, $D\{3,5,4\}$ has the shortest value and thus, $D\{3,5,4\}$ is updated to the latest shortest path and is updated as orange colour as in Figure 8.

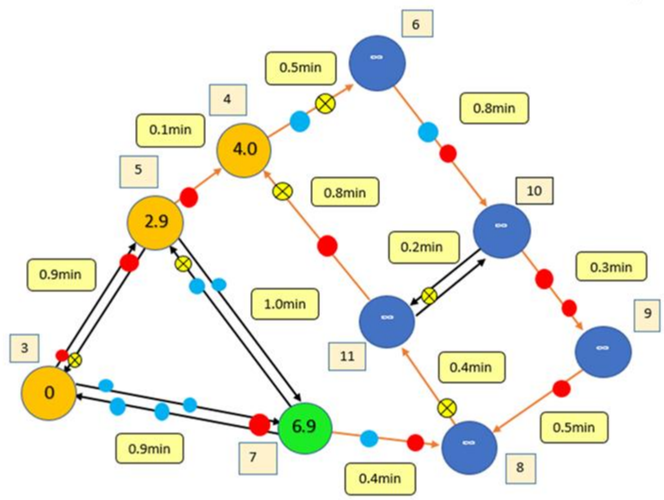

Figure 8: Node 4 is updated as the next shortest path

Step 6: Node 4 is currently selected as shortest path as in Figure 8. Therefore $R = \{3,5,4\}$ with the total of 4 minutes.

Same procedure will be carried out throughout findings shortest path problem until all the nodes has been processed. The calculation stops once the minimum travel time is obtained. After all the shortest path has been updated, the colours are changed to orange. Figure 9 shows the completed network with all the nodes has been processed.

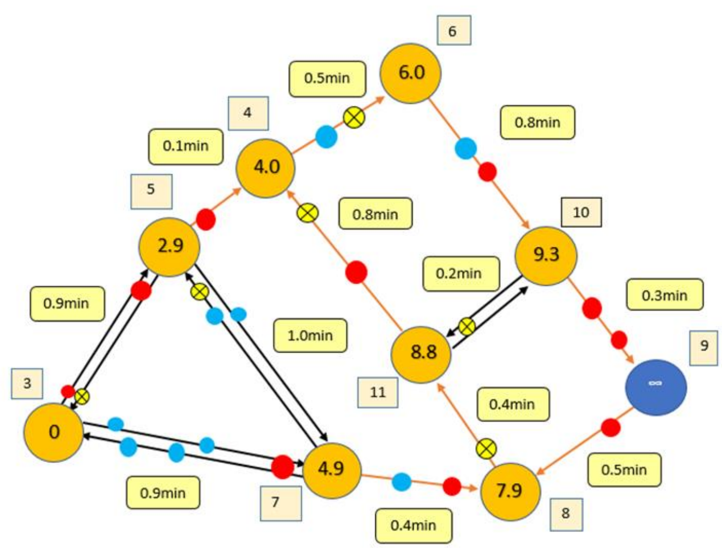

Figure 9: All nodes have been processed and completed

Finally, based on the manual calculation, it is found out that the shortest path from node 3 to node 10 is achieved via path 3 → 5 →4 → 6 → 10. It can also be represented as in the R terms where $R = \{3,5,4,6,10\}$ with 9.3 minutes including all the road penalties. However, doing manual calculations can be tedious and time consuming. It also may increase the error in calculations. Due to this, the help of software is useful since people needs to have fast result in just a matter of time. Thus, in this study, the Microsoft Excel solver is used to find the minimum travel time.

Microsoft Excel Solver for Finding Minimum Travel Time

In this study, Microsoft Excel Solver was utilized as the primary tool for solving the optimization model related to route planning. Solver was chosen due to its capability to handle complex optimization problems efficiently and its integration with Excel, which allowed seamless data management and computation. By using Solver, the optimal travel routes and their corresponding times were determined, forming the foundation for further analysis and decision-making in this research. This preliminary study aimed to validate the feasibility of the proposed approach and ensure accurate results for subsequent stages of the study.

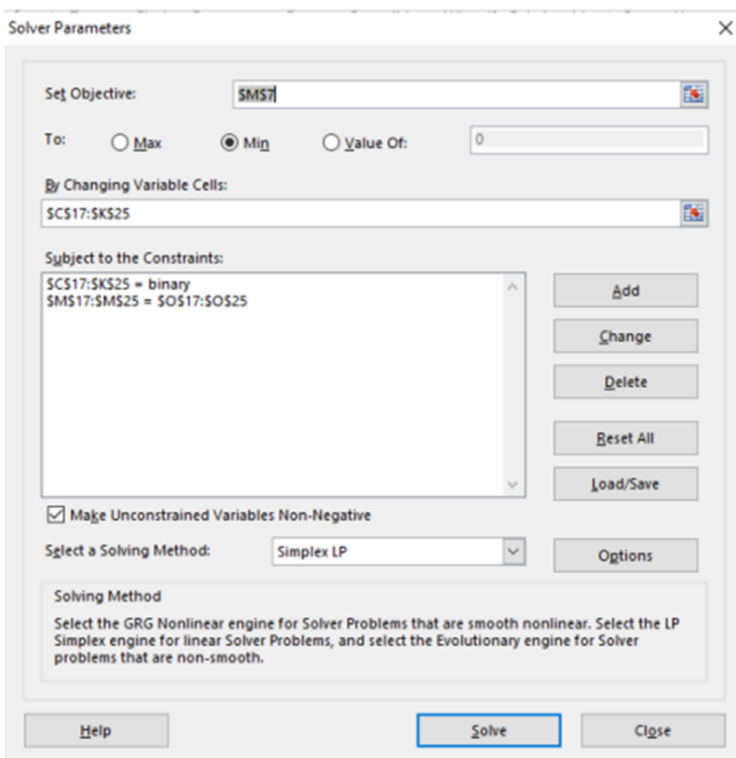

Figure 10: Microsoft Excel solver to find the shortest route with minimum travel time

After setting up the tables in Excel (Table 4), use the Excel Solver to find the shortest route with the least travel distance. Click the Solver button (blue line) to open the Solver dialog box, as shown in Figure 10. Set the objective to "Min" to minimize travel distance, and select the nodes column as the variable cells in the Excel which is the empty column and row before the result were filled as in Table 5.

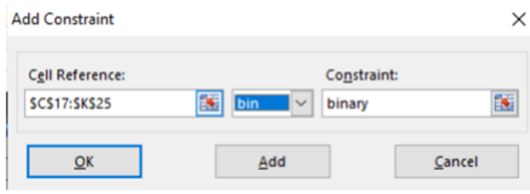

Figure 11: Solver box to add constraints

Next, add constraints in the solver box. Figure 11 shows the first constraint, where the nodes column is set as all different so that there are no repeated nodes. Constraint two was set the starting point and ending point must be equal to one and relate with the $SUMIF$ function. The solving methods are set to Simplex LP and can be solved by clicking the Solve button. The Total Travel Time will be automatically appeared follows the data that has been filled.

The data obtained from area of study has been applied in Microsoft Excel and based on the steps shown. The results obtained are the same as manual calculation in preliminary study which is from $3 - 5 - 4 - 6 - 10$ with a total travel time of 9.3 minutes. This can be seen in Table 4.

Table 4: Travel time from one to one connected node

		TO								
		3	5	4	6	7	8	9	10	11
FROM	3	100	2.9	100	100	3.9	100	100	100	100
	5	2.4	100	1.1	100	4	100	100	100	100
	4	100	100	100	2	100	100	100	100	100
	6	100	100	100	100	100	100	100	3.3	100
	7	4.9	3.5	100	100	100	3.4	100	100	100
	8	100	100	100	100	100	100	100	100	2.4
	9	100	100	100	100	100	2.5	100	100	100
	10	100	100	100	100	100	100	2.3	100	0.7
	11	100	100	2.3	100	100	100	100	0.7	100

Table 4 shows the setup of calculation to find minimum travel time for preliminary study. The column besides the column "FROM" represent as nodes, below the words "TO" also represent as nodes which it will read from left to right. The number of 100 means there is no route between the two nodes, such as from node 3 to node 4, 100 was filled like shows in the Table 4. If the route from one node to other nodes is exist the value of travel time will fill in the table such as from node 3 to node 5 there is route between the two nodes which is 2.9 minutes.

Once the settings are done, the calculation can be obtained by just choosing the starting and ending point. The result based on the Microsoft Excel is discussed and compared with the manual calculation under the Result and Discussion section.

Result and Discussion

All the data pertaining is focusing on the bus transportation in Kuala Lumpur. From there, a complete road network is developed. Hence, once the starting node and the destination node is identified, a detailed road network that shows complete travel times is developed as in Figure 2. In this case, SOGO Kuala Lumpur is the starting node (node 3) and Bahtera Sri Kandi Sdn. Bhd. (node 10) is the ending node whereby the other nodes is connected by edges.

Based on the route network, there are many possibilities to reach node 10. The main concern is whether the route chosen is the fastest. Since in this case, people or the passanger will choose to ride in a less crowded and less congested bus. Other than that, passengers aim at arriving earlier at the designated place. Therefore, shortest path is vital. Thus, the Dijkstra's algorithm is applied in determining the minimum travel time. The results of this algorithm is explained in details in this section.

Table 5: Result of the minimum travel time from node 3 to node 10

FROM \ TO	3	5	4	6	7	8	9	10	11	TOTAL OUT	OUT - IN	=	
3	0	1	0	0	0	0	0	0	0	1	1	=	1
5	0	0	1	0	0	0	0	0	0	1	0	=	0
4	0	0	0	1	0	0	0	0	0	1	0	=	0
6	0	0	0	0	0	0	0	1	0	1	0	=	0
7	0	0	0	0	0	0	0	0	0	0	0	=	0
8	0	0	0	0	0	0	0	0	0	0	0	=	0
9	0	0	0	0	0	0	0	0	0	0	0	=	0
10	0	0	0	0	0	0	0	0	0	0	-1	=	-1
11	0	0	0	0	0	0	0	0	0	0	0	=	0
TOTAL IN	0	1	1	1	0	0	0	1	0				

Table 5 shows the result after run the open solver. The last column shows that there is $1, -1$ and 0.1 are represent as the first node and -1 represent as last node, other nodes was filled with 0. Shows that in the column "Total OUT", the selected node chosen after run the Excel Solver The "Total OUT" column in shortest route problems using Microsoft Excel Solver shows the number of outgoing connections from each node. In a directed graph, where each path can only be used once, a value of "1" indicates an active connection selected by Solver. Since paths are non-repetitive, Solver ensures each connection is unique and forming a valid route. Constraints like "OUT - IN = 0" for intermediate nodes and "OUT - IN = ±1" for start and end nodes maintain route integrity. For example, if Node 3 has a Total OUT of 1, it means Solver selected one outgoing path from Node 3 as part of the shortest route.Number 1 (yellow) shows the confirmed shortest route from node 3 to node 10 other than another route.

TOTAL TRAVEL TIME
9.3

Figure 12: Total Travel Time for preliminary study

Objective function in the excel solver in Figure 12 refers to the solution which is to find the minimum travel time. The result using Microsoft excel shows the total value 9.3 minutes as the from node 3 to node 10.

Based on the comparison between manual calculation and execution using Microsoft Excel software in the previous section, it shows that the minimum travel time from SOGO Kuala Lumpur to Bahtera Sri Kandi Sdn. Bhd is achieved via path $3 \rightarrow 5 \rightarrow 4 \rightarrow 6 \rightarrow 10$, with the total distance of 9.3 minutes. The result shows that the manual calculation is correct since both procedures obtained the same result. This proves that the Dijkstra's algorithm is stable and easy to compute due to the straightforward process.

Conclusion

The study has thus successfully applied Dijkstra's Algorithm in finding the minimum travel time from SOGO Kuala Lumpur to Bahtera Sri Kandi Sdn. Bhd. The algorithm estimated realistic travel times once real factors like traffic lights, junction penalties, and delays at bus stops were added. This approach arrived at the best routes with the minimum time taken, hence assisting in better planning for optimization of bus routes. The study also succeeded in recommending the ideal routes of buses with a minimum of overall travel time in Malaysia, taking into consideration essential factors affecting urban mobility. These recommended routes minimize the time spent in traveling at traffic lights, junctions, and stops.

References

Bogdan, P. (2015). Dijkstra Algorithm in Parallel- case study. Dijkstra Algorithm in Parallel. https://doi.org/10.1109/carpathiancc.2015.7145044

Chen, R. (2022). Dijkstra's Shortest Path algorithm and its application on bus routing. Advances in Economics, Business and Management Research/Advances in Economics, Business and Management Research. https://doi.org/10.2991/aebmr.k.220502.058

Erdogan, G. (2017). An open-source Spreadsheet Solver for Vehicle Routing Problems. Computers & Operations Research, 84, 62–72. https://doi.org/10.1016/j.cor.2017.02.022

Indrajaya, M. A., Affandi, A., & Pratomo, I. (2015). Design of Geographic Information System for tracking and routing using Dijkstra Algorithm for public transportation. 2015 1st International Conference on Wireless and Telematics (ICWT). https://doi.org/10.1109/icwt.2015.7449240

Khaing, O., Wai, H. H., & Myat, E. E. (2018). Using Dijkstra's algorithm for public transportation system in Yangon based on GIS. International Journal of Science and Engineering Applications, 7(11), 442–447. https://doi.org/10.7753/ijsea0711.1008

N. Makariye, "Towards shortest path computation using Dijkstra algorithm," 2017 International Conference on IoT and Application (ICIOT), Nagapattinam, India, 2017, pp. 1-3, https://doi.org/10.1109/ICIOTA.2017.8073641

Mason, A., & Dunning, I. (2010). OpenSolver: Open source optimisation for Excel. Proceedings of the Annual Conference of the Operations Research Society of New Zealand.

Ray, A. (2022). Analysis and design of public transport Route planner: Dijkstras Algorithm. International Journal for Research in Applied Science and Engineering Technology, 10(6), 4571–4575. https://doi.org/10.22214/ijraset.2022.45018