CLSU International Journal of Science and Technology
www.clsu-ijst.org
ISSN 2507 -9638 (online)
The CLSU International Journal of Science and Technology was previously published as CLSU Scientific Journal since 1965.

Determining the Shortest Paths from NATO Stations to Different Colleges in CLSU Using Dijkstra's Algorithm

John Rafael M. Antalan[1], Jan Martin S. Gonzales[1,2] and Michael Angelo P. Valete[3]

[1]Department of Mathematics and Physics, College of Science, Central Luzon State University (3120), Science City of Muñoz, Nueva Ecija, Philippines
[2]Institute for Climate Change and Environmental Management, College of Science, Central Luzon State University (3120), Science City of Muñoz, Nueva Ecija, Philippines
[3]Philippine Space Agency, 29th Floor, Cyber One Building, 11 Eastwood Avenue, Bagumbayan, Quezon City, Metro Manila, Philippines

Email for correspondence: jrantalan@clsu.edu.ph

Submitted: January 29, 2023. Accepted: June 15, 2023. Published Online: August 31, 2023.

Abstract

In Mathematics and Computer Science, a graph is a structure made up of *vertices* (also called *nodes*) and lines called *edges* such that each edge connects two nodes. Graphs are used to depict various systems to facilitate further studies on them. In this simple study, we represent the three (3) New Association of Tricycle Operators (NATO) stations, seven (7) colleges (excluding the College of Agriculture and the College of Fisheries) of the Central Luzon State University (CLSU), and some major road intersections in CLSU as vertices; while the roads that connect them are represented by edges to obtain a graph representation for CLSU. The shortest paths going to the seven colleges in CLSU from the three NATO stations were then determined using the Dijkstra's Algorithm, limited only on the condition that a NATO driver has only one passenger or has more than one passenger but all of them are going to the same destination. With this study, a NATO driver can save fuel, students can save time, and Carbon Dioxide (CO2) emission from vehicles may be lessened by taking the shortest paths determined. The study also serves as a motivation in appreciating the applications of Mathematics in real life. For further study, it is good to determine the shortest paths from a specific college going to every other college.

Keywords: Graph, Shortest Path, Dijkstra's Algorithm, Gasoline Consumption, CO2 Emission

Introduction

The Central Luzon State University (CLSU) is one of the renowned and prestigious institutions of higher learning in the Philippines. It is located at Science City of Munoz, Nueva Ecija. With its agricultural state, the campus is a sprawling 658-hectare area (https://clsu.edu.ph/about-us/). Due to its vastness, the means of transportation is very important. The New Association of Tricycle Operators (NATO) made it possible for the students, faculty, and staff to conveniently go from one place to another inside the CLSU campus and arrive at their destinations on time.

Since gasoline prices in the world market have increased, and oil price hike in the country is unending, one major problem of motorists is the minimization of their gasoline consumption. Also, time is another human resource which must not be wasted. For a student, time is very important, and every second of it must not be wasted. Recognizing these problems, the researchers want to help the members of NATO CLSU in minimizing their gasoline consumption and the CLSU students in saving their time from commuting.

Knowing that gasoline consumption and travel time is directly proportional to the distance travelled, all other things equal (Sivak, 2013), like road conditions, vehicle conditions, vehicle load, etc., minimizing the distance to be travelled will minimize gasoline consumption and travel time. Thus, by taking the shortest path or route from a NATO station to a certain destination in CLSU, the gasoline consumption of a NATO driver will be lessened, and the travel time of students will be lessened as well. Also, according to WLTP Facts. EU, "the amount of Carbon Dioxide (CO2) a vehicle emit is directly related to the amount of fuel it consumes" for this reason, reduced distance travelled may reduce CO2 emission, which is beneficial to our environment.

This study aims to determine the shortest paths from a NATO station to different places in CLSU by applying Dijkstra's Algorithm. Specifically, it sought to obtain a graph representation of CLSU, and use the graph to determine the shortest paths from a NATO station to every college in CLSU (except the College of Agriculture and the College of Fisheries).

This research study will be beneficial to NATO operators, drivers, and their commuters, especially the CLSU students. The study, in addition, is beneficial to our environment. Finally, this research can be a motivation to others to learn how to apply graph theory or even other branches of Mathematics in our daily lives.

For the scope and limitation of this study, only the main roads were considered as edges of the graph. The road intersections, NATO stations, and colleges were considered as the nodes. Furthermore, the roads that obviously give a long path from the starting point or origin to the end point or destination will not be considered. These ideas were illustrated in Figure 1. The distinct nodes representing the different colleges were chosen to be located at the lobby or at the middle front of each of the colleges. This study is only limited on the situation that a NATO driver has only one passenger (special trip) or has more than one passenger going to the same destination (regular homogenous trip). Lastly, the study was conducted in 2014, and thus, does not include newly constructed buildings in CLSU such as the Department of Hospitality and Tourism Management Department Building (Part of CHSI).

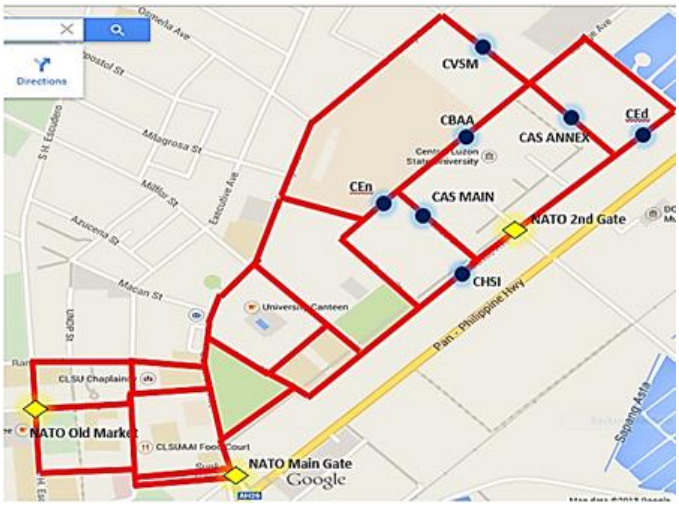

Figure 1. CLSU map with the considered roads marked with red lines taken via Google Maps (Google, n.d.)

Essential Graph Theory Concepts

The field of Mathematics that studies graphs is called Graph Theory. In this section, we state some graph theory concepts which are essential in the understanding of the methods used in obtaining the results of this paper. For further reading in Graph Theory, we

refer the reader to Wilson (2010) and Bondy & Murty (2008).

Definition 1. A ***graph*** G is a structure consisting of two sets $V(G)$ and $E(G)$ where $V(G)$ is called the vertex set of G and $E(G)$ (possibly empty) is the edge set of G. The elements of $V(G)$ are called ***vertices*** or ***nodes*** and the elements of $E(G)$ are unordered pairs of distinct vertices in $V(G)$ called ***edges***.

Definition 2. A ***weighted graph*** is a graph whose every edge has an associated numerical value which is called ***weight***.

An example of a weighted graph (and respectively, a graph) is given in Figure 2. Let us agree to call that graph as our toy graph. Our toy graph G has vertex set $V(G) = \{A, B, C, D, E, F\}$ and edge set $E(G) = \{\{A, B\}, \{A, C\}, \{B, D\}, \{C, D\}, \{C, E\}, \{D, E\}, \{D, F\}, \{E, F\}\}$. The weight of the edge $\{A, B\}$ is 5 while the weight of the edge $\{E, F\}$ is 7.

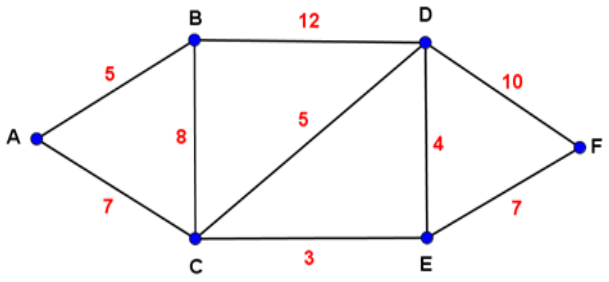

Figure 2. Our toy graph G.

Definition 3. A ***walk*** W in a graph is a finite non-null sequence $v_0, e_1, v_1, e_2, v_2, \ldots, e_k, v_k$ whose terms are alternately nodes and edges, such that, for $1 \leq i \leq k$, the ends of e_i are $v_{i-1}$and v_i.

Since the ends of an edge e_i are the vertices v_i and v_{i-1}, we simply denote the walk $W = v_0, e_1, v_1, e_2, v_2, \ldots, e_k, v_k$ by $W = v_0, v_1, v_2, \ldots, v_k$ for simplicity. The sequence of vertices and edges $\{A, \{A, B\}, B, \{B, D\}, D, \{D, E\}, E, \{E, F\}, F\}$ is an example of a walk in our toy graph. The pictorial representation of the walk is provided in Figure 3 with green-colored edges as guide. For simplicity, the said walk can be rewritten as $\{A, B, D, E, F\}$.

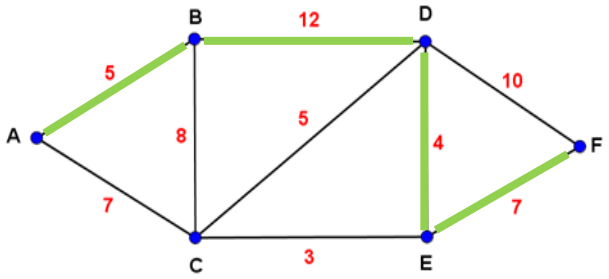

Figure 3. The walk $\{A, B, D, E, F\}$ in our toy graph.

Definition 4. A ***trail*** is a walk with distinct edges. On the other hand, a ***path*** is a trail with distinct vertices. Finally, a ***cycle*** is a trail whose origin (or starting vertex) and terminus (or end vertex) are the same.

The walk $\{A, B, D, E, F\}$ in Figure 3 is an example of a trail and a path. The walk $\{A, B, D, E, F, D, C, A\}$, pictorially depicted by yellow-colored edges in Figure 4 is not a path, but it is a trail and at the same time, a cycle.

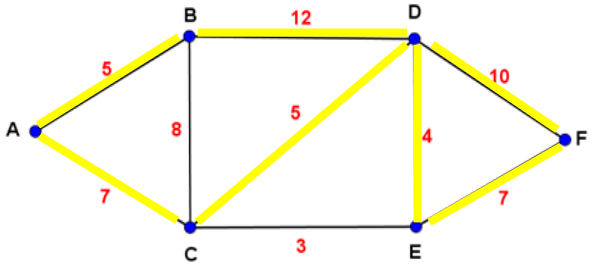

Figure 4. The cycle $\{A, B, D, E, F, D, C, A\}$ in our toy graph.

Definition 5. A ***tree*** is a connected graph that does not contain any cycle.

Our toy graph G is not a tree, however, it contains tree components. For instance, the walk $\{A, B, D, E, F\}$ in our toy graph is a tree.

We now end the section by introducing the Dijkstra's algorithm. The Dijkstra's algorithm is an algorithm for finding the shortest paths between nodes in a graph. It was developed by computer scientist Edsger W. Dijkstra in 1956 and published in (Dijkstra, 1959). Originally, the algorithm finds the shortest path between two nodes, however after 1959, various variants of the algorithm were developed. A more common variant fixes a single node as the "source node" and finds shortest paths from the source to all the other nodes; for instance, see Venkat (2014).

The details of the algorithm will be presented in the next section. As a notation, we let the node at which we are starting to be called the ***initial node***, the ***distance of node Y*** to be the distance from the initial node to Y, and the ***unvisited set*** be the set of all unvisited nodes.

Some research works that utilize Dijkstra's algorithm in determining a shortest path or route from one place to another were done by Shehzad and Shah (2009), Amaliah et al. (2016), Ojekudo and Akpan (2017), Gunawan et al. (2019), Fitriansyah et al. (2019). Shehzad and Shah used Dijkstra's algorithm to determine and evaluate the shortest paths in road network of Sindh-Pakistan. On the other hand, the group of Amaliah used the Dijkstra's algorithm in finding the shortest paths among cities in Java Island Indonesia. In a similar manner, Ojekundo and Akpan used Dijkstra's algorithm in helping the Dominion Paints Nig. Ltd in transporting their products from their production plant to store of sales. The work of the group of Gunawan focused on searching the nearest specialist doctor in Bandar Lampung Indonesia using Dijkstra's algorithm. Finally, the team of Fitriansyah used Dijkstra's algorithm to find a shortest path of tourist destination in Bali.

Other research studies that utilize Dijkstra's algorithm are the study of Xiao-Yan & Yan-Li (2010), Tirastittam & Waiyawuththanapoom (2014), and Parsakhoo & Jajouzadeh (2016). Xiao-Yan and Yan-Li considered the application of Dijkstra's algorithm in logistics distribution line. Tirastittam and Waiyawuththanapoom on the other hand created a transport planning system in Bangkok Metropolitan area using Dijkstra's algorithm. Lastly, Parsakhoo and Jajouzadeh used Dijkstra's algorithm in determining an optimal path for the road construction in Nahar Khvoran forest.

Materials and Methods

The map of CLSU with its actual road lengths used in this research was taken from the Google maps (Google, n.d.). The actual road lengths were measured in meters and were rounded up to two decimal places. After that, a graph representation of the CLSU map was made. To form the graph representation *(see Figure 5)*, the roads were represented by lines called *edges* and the road intersections and destinations were represented by dots called *nodes*. The nodes were labelled by the letters of the English alphabet and some letters of the Greek alphabet. Some of the node representations were:

α_1- NATO Main Gate Station

α_2- NATO 2nd Gate Station
α_3- NATO Old Market Station
Ω_1- College of Engineering
Ω_2- College of Arts and Sciences-Main (Now, College of Arts and Social Sciences)
Ω_3- College of Home Science and Industry
Ω_4- College of Education
Ω_5- College of Arts and Sciences-Annex (Now, College of Science)
Ω_6- College of Veterinary Science and Medicine
Ω_7- College of Business Administration and Accountancy

The remaining nodes were named from $A - Z$ and $A_1 - H_1$. For the names of the other nodes, we refer the reader to Antalan et al. (2015).

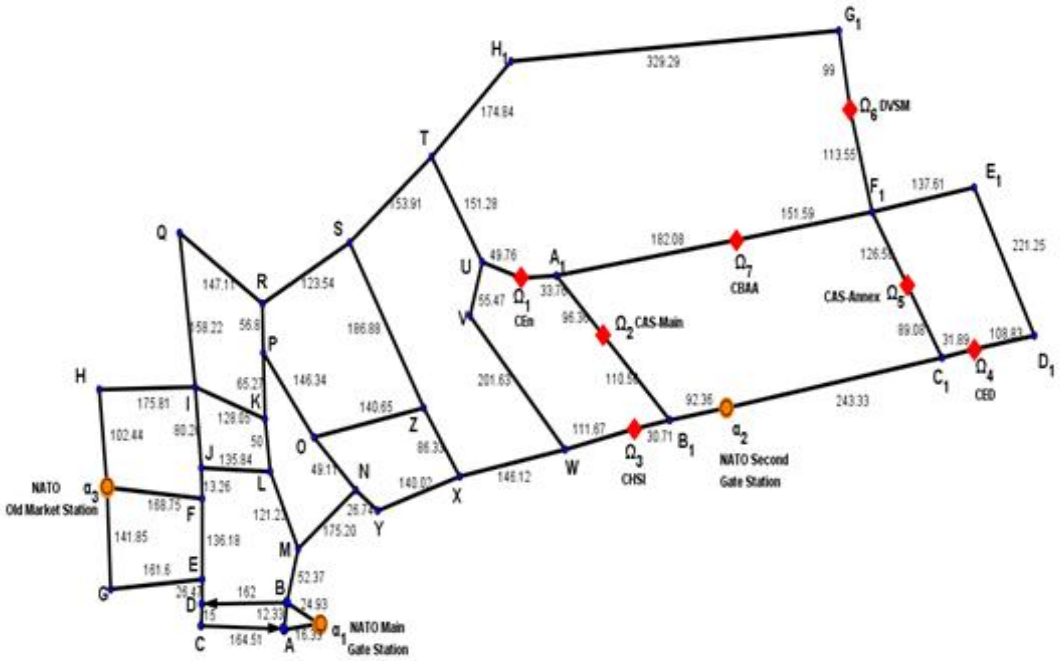

Figure 5. The weighted graph representation of the CLSU map.

In this research, the method used in obtaining the desired shortest paths is a variation of Dijkstra's Algorithm. The following are the steps of the algorithm.

Step 1: Construct an *n* column *by (n+1)* row table where *n* is the total number of nodes in the graph. In the first row of the table, label the first column with "*Current node*" and label the remaining columns with the nodes except the initial node.

Step 2: Set the initial node as current node.

Step 3: For the current node, consider all its unvisited neighbors and calculate their tentative distances. ***Tentative distances*** are distances obtained based on the current node (or row of the current node). Compare the newly calculated tentative distances to their corresponding current assigned values (or current distances) and assign the smaller one as the "new" current distance. ***Current distances*** are distances obtained from the previous current node (or row of the previous current node). Assign infinity for all other nodes. For simplicity, we use blank cell to denote infinity. In the case of the initial node being the first current node, just assign the newly calculated tentative distances as "new" current distance.

Step 4: When we are done considering all the neighbors of the current node, mark the current node as visited and remove it from the unvisited set. A visited node will never be checked again.

Step 5: If the smallest tentative distance among the nodes in the unvisited set is infinity or if all the nodes are already visited then stop. The algorithm has finished. If not,

select the unvisited node that is marked with the smallest "new" current distance, and set it as the new "current node" then go back to Step 3.

Illustration. Suppose in our toy graph, we are asked to find the shortest paths from the initial node **A** to all the other nodes. We denote by $d(X,Y)$ be the distance from node X to node Y.

Step 1: *Construct an n column by (n+1) row table where n is the total number of nodes in the graph. In the first row of the table, label the first column with 'Current node" and label the remaining columns with the nodes except the initial node.*

First, we construct the 6-column by 7-row table below and label the first column-row entry by "Current node" while the other entries in the first row are labeled B, C, D, E, and F, respectively (the non-initial nodes in our toy-graph).

Current node	B	C	D	E	F

Step 2: *Set the initial node as current node.*

Next, we set node **A** to be the current node since it is our initial node.

Current node	B	C	D	E	F
A					

Step 3: *For the current node, consider all its unvisited neighbors and calculate their tentative distances. Tentative distances are distances obtained based on the current node (or row of the current node). Compare the newly calculated tentative distances to their corresponding current assigned values (or current distances) and assign the smaller one as the 'new" current distance. Current distances are distances obtained from the previous current node (or row of the previous current node). Assign infinity for all other nodes. For simplicity, we use blank cell to denote infinity. In the case of the initial node being the first current node, just assign the newly calculated tentative distances as 'new" current distance.*

The unvisited neighbors of the current node **A** are the nodes **B** and **C** whose distances from **A** are 5 and 7 respectively. We then enter the respective tentative distances of **B** and **C** in their corresponding columns in the table. Since there is no distance of comparison at this stage, the respective distances automatically become the "new" current distance of node **B** and node **C** from node **A**.

Current node	B	C	D	E	F
A	5 (**A**)	7 (**A**)			

We note that the notation 5(**A**) in the B-column above, means that the shortest path to **B** from the initial node **A** is 5, and that path passed through node **A** right before reaching node **B**. This notation will be used throughout the implementation of the algorithm.

Since node **B** has the shortest current distance, we set it as the "new" current node. We now check the neighbors of **B** which are **C** and **D**.

To obtain the tentative distance of node **C**, we add the current distance of node **B** from node **A** to $d(B,C)$; that is, 5 + 8 = 13. Similarly, the tentative distance of node **D** is given by 5 + 12 = 17. Comparing the tentative distance of node **C** from node **A** which is 13 to the current distance of node **C** from node **A**, which is 7, the current distance is shorter, so we set it as the "new" current distance of node **C** from node **A**. Also, comparing the tentative distance of node **D** from node **A** which is 17 to the current distance of node **D** from node **A,** which is infinity, the tentative distance is shorter, so we set it as the "new" current distance of node **D** from node **A**.

Current node	B	C	D	E	F
A	5 (**A**)	7 (**A**)			
B	5 (**A**)	7 (**A**)	17(**B**)		

Step 4: *When we are done considering all the neighbors of the current node, mark the current node as visited and remove it from the unvisited set. A visited node will never be checked again.*

Observe that at this point, all the neighbors of the current node **B** have been considered. This means that **B** has been already visited so we color the B-column to denote that **B** is already an element of the visited set.

Current node	B	C	D	E	F
A	5 (**A**)	7 (**A**)			
B	5 (**A**)	7 (**A**)	17(**B**)		

Step 5: *If the smallest tentative distance among the nodes in the unvisited set is infinity or if all the nodes are already visited then stop. The algorithm has finished. If not, select the unvisited node that is marked with the smallest 'new" current distance, and set it as the new "current node" then go back to Step 3.*

Among the unvisited nodes, node **C** has the shortest "new" current distance, so we set it as the "new" current node. Since $d(C,D) = 5$, and the current distance of node **C** from node **A** is 7, we add them to get 12 as the tentative distance of node **D** from node **A**. This newly calculated distance, which is the tentative distance, is shorter than the current distance of node **D** from node **A** which is 17. Hence, we choose 12 as the "new" current distance of node **D** from node **A**. On the other hand, $d(C,E) = 3$ and adding it to the current distance of node **C** from node **A** which is 7, gives 10 as the tentative distance of node **E** from node **A**. Comparing the tentative distance of node **E** from node **A** which is 10 to the current distance of node **E** from node **A** which is infinity, the tentative distance is shorter, so we set it as the "new" current distance of node **E** from node **A**. Then we color the C-column for **C** has been visited.

Current node	B	C	D	E	F
A	5 (**A**)	7 (**A**)			
B	5 (**A**)	7 (**A**)	17(**B**)		
C	5 (**A**)	7 (**A**)	12(**C**)	10(**C**)	

Continuing the process, we will arrive with the table below.

Current node	B	C	D	E	F
A	5 (**A**)	7 (**A**)			
B	5 (**A**)	7 (**A**)	17(**B**)		
C	5 (**A**)	7 (**A**)	12(**C**)	10(**C**)	
E	5 (**A**)	7 (**A**)	12(**C**)	10 (**C**)	17**(E)**
D	5 (**A**)	7 (**A**)	12(**C**)	10 (**C**)	17**(E)**
F	5 (**A**)	7 (**A**)	12(**C**)	10 (**C**)	17**(E)**

The last row is highlighted because of its importance. It gives the shortest distance of every node from node **A**. Also, from it, we can determine the shortest distance of a node, and a shortest path to every node from **A**. To obtain the shortest path from the initial node **A** going to the node, say **F**, just look at the last row of the F-column.

From there, by the notation 17**(E)**, we see that the distance of node **F** from node **A** is 17 and that the shortest path going to **F** comes from node **E**. Then look at the last row of the E-column. We see that the shortest path to node **E** comes from node **C**. The shortest path to node **C**, as we can see at the last row of the C-column, comes from node **A**, which is our initial node. Therefore, a shortest path from node **A** to node **F** is given by the *path* **ACEF.** By applying the same method, we obtain the list of shortest paths from node **A** with their corresponding distances in Table 1.

Table 1. The shortest paths from node **A** with their corresponding distances in our toy graph.

DESTINATION	SHORTEST PATH	DISTANCE
B	**AB**	5
C	**AC**	7
D	**ACD**	12
E	**ACE**	10
F	**ACEF**	17

These paths are shown and are represented by the broken lines in the graph shown in Figure 6.

DOI: https://doi.org/10.22137/ijst.2023.v7n2.08

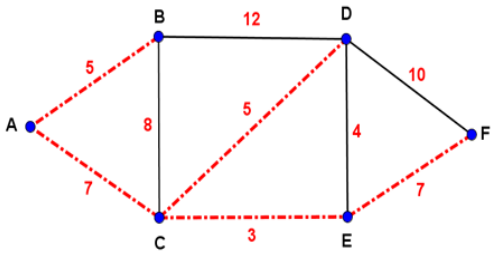

Figure 6. The shortest paths from node A to all the other vertices in our toy graph.

Results and Discussion

Applying the process discussed in the previous section to Figure 5, the researchers obtain the shortest paths from every NATO station to the different colleges in CLSU. The results are summarized in Tables 2 – 4. For the complete details of the calculations, we refer the reader to Antalan et al. (2015).

Table 2. The shortest paths from node α_1 with their corresponding distances.

Destination	Shortest path	Distance (m)
Ω_1(CEn)	$\alpha_1 BMLKRSTU\Omega_1$	849.09
Ω_2(CAS Main)	$\alpha_1 BMNYXW\Omega_3 B_1$	818.32
Ω_3(CHSI)	$\alpha_1 BMNYXW\Omega_3$	677.05
Ω_4(CEd)	$\alpha_1 BMNYXW\Omega_3 B_1 \alpha_2 C_1 \Omega_4$	1075.34
Ω_5(CAS Annex)	$\alpha_1 BMNYXW\Omega_3 B_1 \alpha_2 C_1 \Omega_5$	1132.53
Ω_6(CVSM)	$\alpha_1 BMLKRSTH_1 G_1 \Omega_6$	1251.18
Ω_7(CBAA)	$\alpha_1 BMLKRSTU\Omega_1 A_1 \Omega_7$	1064.93

Table 2 shows the shortest paths from the NATO Main Gate station to the seven colleges in CLSU with their corresponding distances. If a student wishes to go to the College of Arts and Sciences Annex (now College of Science) via NATO tricycle, the NATO driver must take the route shown in Figure 7 with a total length of 1132.53 meters. That is, from the NATO Main Gate station, the driver must pass through the College of Home Science and Industry via the road intersections B, M, N, Y, X and W. Then, pass through NATO Second Gate Station via the road intersection B_1. Finally, the driver must turn left via the road intersection C_1 to arrive at the College of Arts and Sciences Annex (now College of Science).

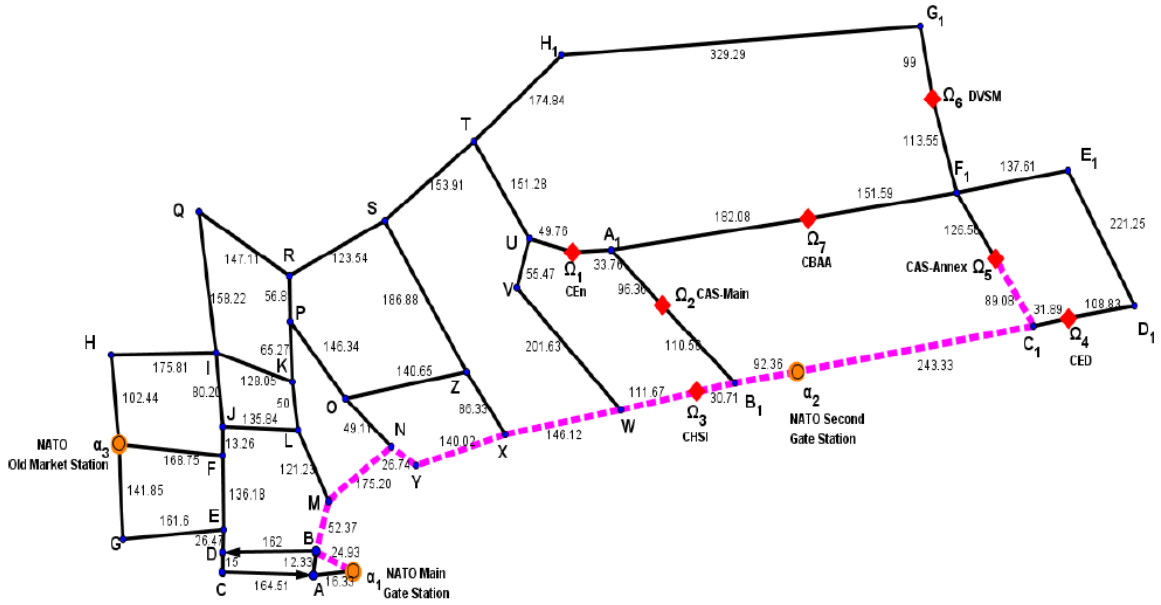

Figure 7. The shortest path from the NATO MAIN GATE station to CAS Annex (now College of Science).

For the visual representation of the paths stated in Table 2, we again refer the reader to Antalan et al. (2015).

Table 3. The shortest paths from node α_2 with their corresponding distances.

Destination	Shortest path	Distance (m)
Ω_1(CEn)	$\alpha_2 B_1 \Omega_2 A_1 \Omega_1$	333.04
Ω_2(CAS Main)	$\alpha_2 B_1 \Omega_2$	202.92
Ω_3(CHSI)	$\alpha_2 B_1 \Omega_3$	123.07
Ω_4(CEd)	$\alpha_2 C_1 \Omega_4$.	275.22
Ω_5(CAS Annex)	$\alpha_2 C_1 \Omega_5$	332.41
Ω_6(CVSM)	$\alpha_2 C_1 \Omega_5 F_1 \Omega_6$	572.54
Ω_7(CBAA)	$\alpha_2 B_1 \Omega_2 A_1 \Omega_7$	481.36

Table 3 shows the shortest paths from the NATO Second Gate station to the seven colleges in CLSU with their corresponding distances. If a student wishes to go to the College of Veterinary Science and Medicine via NATO tricycle, the NATO driver must take the route shown in Figure 8 with a total length of 572.54 meters. That is, from the NATO Second Gate Station, the driver must turn left via the road intersection C_1, then pass through the College of Arts and Sciences Annex (now College of Science) and road intersection F_1 before arriving at the College of Veterinary Science and Medicine.

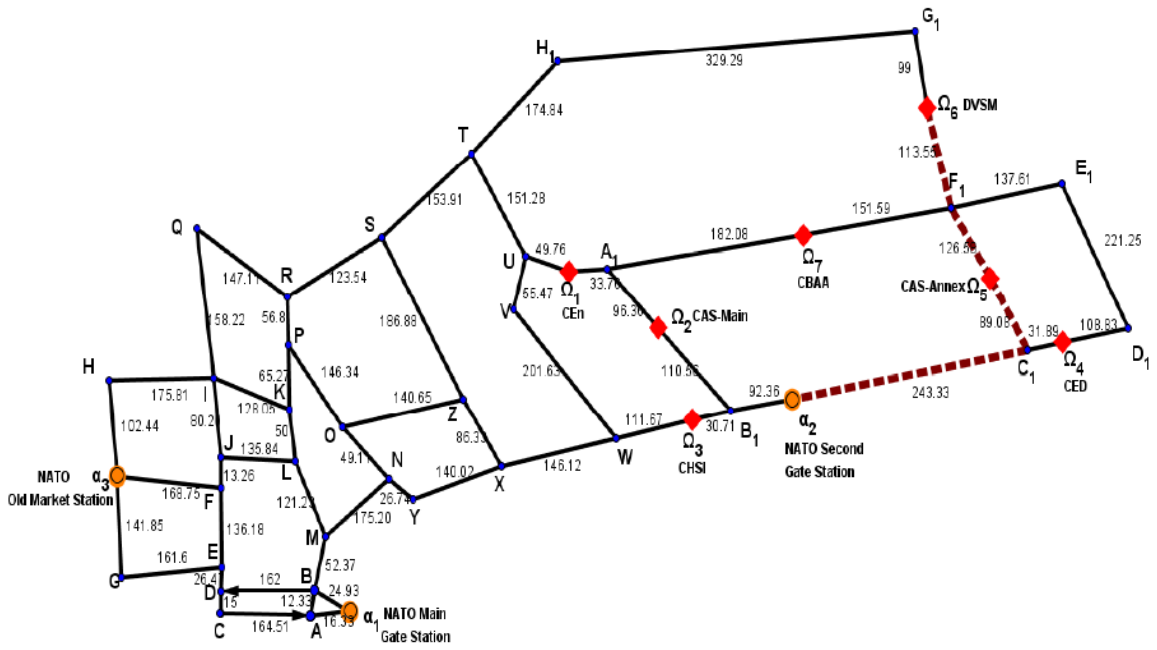

Figure 8. The shortest path from the NATO Second GATE station to CVSM.

For the visual representation of the paths stated in Table 3, we again refer the reader to Antalan et al. (2015).

Table 4. The shortest paths from node α_3 with their corresponding distances.

Destination	Shortest path	Distance (m)
Ω_1(CEn)	$\alpha_3 FJLKPRSTU\Omega_1$	968.41
Ω_2(CAS Main)	$\alpha_3 FJLKPRSTU\Omega_1 A_1\Omega_2$	1098.53
Ω_3(CHSI)	$\alpha_3 FJLMNYXW\Omega_3$	1038.83
Ω_4(CEd)	$\alpha_3 FJLMNYXW\Omega_3 B_1\alpha_2 C_1\Omega_4$	1437.12
Ω_5(CAS Annex)	$\alpha_3 FJLKPRSTU\Omega_1 A_1\Omega_7 F_1\Omega_5$	1462.42
Ω_6(CVSM)	$\alpha_3 FJLKPRSTH_1 G_1\Omega_6$	1370.50
Ω_7(CBAA)	$\alpha_3 FJLKPRSTU\Omega_1 A_1\Omega_7$	1184.25

Table 4 shows the shortest paths from the NATO Old Market station to the seven colleges in CLSU with their corresponding distances. For an illustrative example, if a student wishes to go to College of Education via NATO tricycle, the NATO driver must take the route shown in Figure 9 with a total distance of 1437.12 meters. Starting from the NATO Old Market Station, the driver must pass through the College of Home Science and Industry via the road intersections F, J, L, M, N, Y, X, W. Then pass through NATO Second Gate Station via the road intersection B_1. Finally, the driver must past through the road intersection C_1 to arrive at the College of Education.

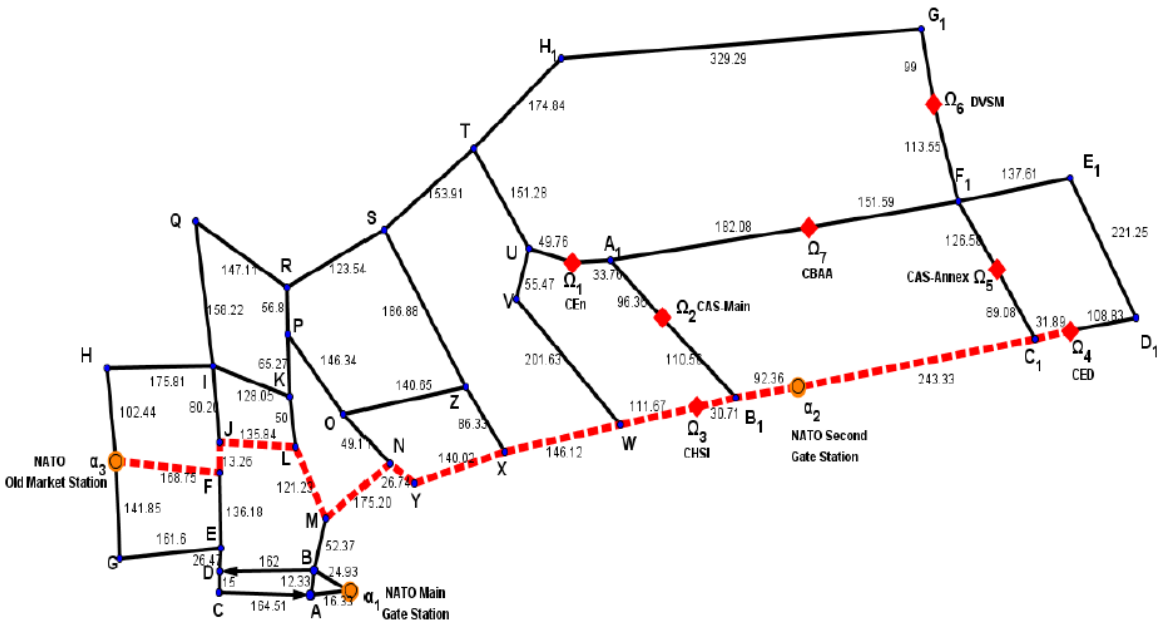

Figure 9. The shortest path from the NATO Old Market station to College of Education.

For the visual representation of the paths stated in Table 4, we again refer the reader to Antalan et al. (2015).

To end the results section, we present the shortest paths tree generated by the algorithm. The shortest paths tree is a tree containing all the shortest paths from a particular start vertex (NATO Main Gate, NATO 2nd Gate, or NATO Old Market) to all the desired end vertices (Colleges in CLSU). The shortest path trees are given in Figures 10 – 12.

Figure 10. The shortest paths three from the NATO Main Gate station to the colleges of CLSU taken via Google Maps (Google, n.d.).

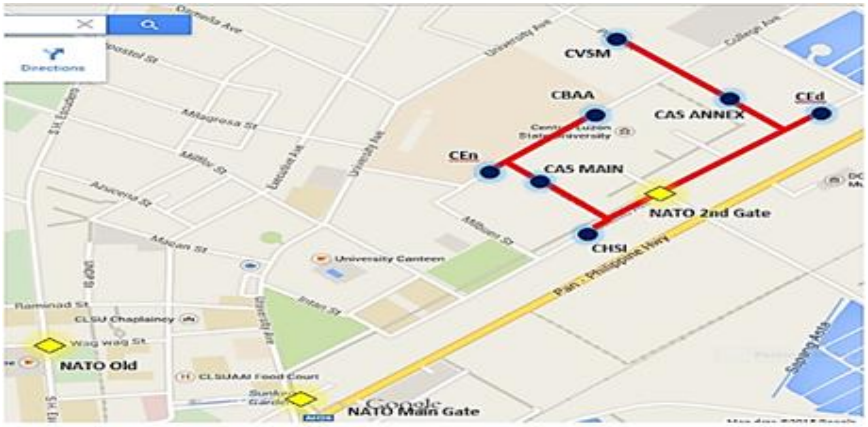

Figure 11. The shortest paths tree from the NATO Second Gate station to the colleges of CLSU taken via Google Maps (Google, n.d.).

Figure 12. The shortest paths tree from the NATO Old Market station to the colleges of CLSU taken via Google Maps (Google, n.d.)

Conclusion

In this paper, following the 2014 setting of establishments in CLSU, the shortest paths going to the seven colleges of CLSU from all NATO stations were determined using the Dijkstra's Algorithm, with the condition that a NATO operator has only one passenger or has more than one passenger but all of them are going to the same destination. As a result, a NATO driver coming from NATO Main Gate station going to any of the seven colleges of CLSU must follow the paths presented in Table 2. On the other hand, a NATO driver coming from NATO Second Gate station going to any of the seven colleges of CLSU must follow the paths presented in Table 3. Finally, a NATO driver coming from NATO Old Market station going to any of the seven colleges of CLSU must follow the paths presented in Table 4.

To further improve this study, the researchers recommend the following: (1) consider other frequent destinations in CLSU such as food courts, the College of Agriculture, and College of Fisheries, (2) consider as additional edges of the graph the passages which are not roads but NATO operators use as their path, (3) consider the condition of every road in giving its weight, (4) consider the condition that a NATO operator has at least two passengers going to different destinations, and (5) reconduct the study covering the recent infrastructure developments in CLSU.

Acknowledgements

The authors would like to thank their families and friends for their motivation. They are thankful also to the Central Luzon State University and to the Department of Mathematics and Physics for their unending support and encouragement. Lastly, the authors are indebted to the anonymous reviewers for their comments and suggestions that contributed to the improvement of the manuscript.

References

Amaliah, B., Fatichah, C. & Riptianingdyah, O. (2016). Finding the Shortest Paths Among Cities in Java Island Using Node Combination Based on Dijkstra Algorithm. *International Journal on Smart Sensing and Intelligent Systems*, 9(4), 2219-2236. https://doi.org/10.21307/ijssis-2017-961

Antalan, J., Gonzales, J., & Valete, M. (2015). *Essential images and appendices of the undergraduate mathematics special problem - Determining the shortest paths from NATO stations to different colleges in CLSU: An application of Dijkstra's algorithm*. Unpublished manuscript. *https://www.researchgate.net/publication/358199072_DETERMINING_THE_SHORTEST_PATHS_FROM_NATO_STATIONS_TO_DIFFERENT_COLLEGES_IN_CLSU_AN_APPLICATION_OF_DIJKSTRA'S_ALGORITHM*

Bondy, J., & Murty, U. (2008). *Graduate texts in mathematics – Graph theory*. Springer

Central Luzon State University. (n.d.). About us. From *http://clsu.edu.ph/about-us*

Dijkstra, E. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik. 1, 269-271. https://doi.org/10.1007/BF01386390*

Fitriansyah, A., Parwati, N. W., Wardhani, D. R., & Kustian, N. (2019, October). Dijkstra's algorithm to find shortest path of tourist destination in Bali. In *Journal of Physics: Conference Series* (Vol. 1338, No. 1, p. 012044). IOP Publishing. *https://doi.org/10.1088/1742-6596/1338/1/012044*

Google. (n.d.). [Google Maps Central Luzon State University]. Retrieved January 2015, from *https://www.google.com/maps/search/central+luzon+state+university/@15.7318223,120.9280215,18z/data=!3m1!4b1x*

Gunawan, R. D., Napianto, R., Borman, R. I., & Hanifah, I. (2019). Implementation of Dijkstra's Algorithm in Determining the Shortest Path (Case Study: Specialist Doctor Search in Bandar Lampung). *International Journal of Information System and Computer Science*, 3(3), 98-106. https://core.ac.uk/reader/276535300

Ojekudo, N.A., & Akpan, N.P. (2017). An Application of Dijkstra's Algorithm to Shortest Route Problem. *IOSR Journal of Mathematics*, 13(3), 20-32. https://www.iosrjournals.org/iosr-jm/papers/Vol13-issue3/Version-1/C1303012032.pdf

Parsakhoo, A., & Jajouzadeh, M. (2016). Determining an optimal path for forest road construction using Dijkstra's algorithm. *Journal of Forest Science 62(6), 264-* 268. *https://doi.org/10.17221/9/2016-JFS*

Shehzad, F., & Shah, M. A. A. (2009). Evaluation of Shortest Paths in Road Network of Sindh-Pakistan. *Pakistan*

Journal of Commerce and Social Sciences, 3, 67-79. https://www.econstor.eu/bitstream/10419/187996/1/pjcss025.pdf

Sivak, M. (2013). Effects of Vehicle Fuel Economy, Distance Travelled, and Vehicle Load on the Amount of Fuel Used for Personal Transportation in the U.S.: 1970-2010. University of Michigan Transportation Research Institute (UMTRI). https://deepblue.lib.umich.edu/bitstream/handle/2027.42/96632/102926.pdf

Tirastittam, P., & Waiyawuththanapoom, P. (2014). Public transport planning system by Dijkstra's algorithm: Case study Bangkok metropolitan area. *International Journal of Computer and Information Engineering 8(1), 54-59. https://doi.org/10.5281/zenodo.1336370*

Venkat, R. (2014). *Pathfinding - Dijkstra's algorithm.* Unpublished manuscript. *http://cs.indstate.edu/~rjaliparthive/dijkstras.pdf*

Wilson, R. (2010). *Introduction to graph theory (5th ed.).* Pearson

What is the Link between the CO2 Emissions and Fuel Consumption of My Car? (2017). Retrieved from https://www.wltpfacts.eu/link-between-co2-emissions-fuel-consumption/?fbclid=IwAR3pido4muKMilEePZok4w3Y458x00d55okPbpGLhuFmohkSuPi6igj_1D4

Xiao-Yan, L., & Yan-Li, C. (2010, August). Application of Dijkstra algorithm in logistics distribution lines. In *Third International Symposium on Computer Science and Computational Technology (ISCSCT'10), Jiaozuo, PR China* (pp. 048-050). https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=205ddd97e5e4a925823ff723b4fbd3f9a5070fb3