Journal of Innovative Technology Convergence
Vol. 4, No. 1 June 2022, pp. 21-30

Comparative Analysis of Shortest Route Problem Algorithms for Emergency Response Services

Jeacar D. Zamora [1], Ace Marco Floro E. Angeles [2], Marnei A. Manalo [3]

Abstract: Emergency response services require a fast and shortest possible route when responding to emergency situations that include road accidents, crimes, the occurrence of fires, etc. In this regard, an efficient and robust shortest route locator is essentially important. This study primarily aims to analyze the performance of the various shortest route problem algorithms, including the regular Dijkstra's algorithm, the Bidirectional search algorithm, and the Multidirectional Dijkstra's algorithm (MDA). The implementations of these algorithms are evaluated and compared in order to determine which algorithm can be an excellent choice to be adopted by such emergency response services. The results showed that the MDA has efficient performance in both light and heavy traffic situations in terms of speed and response times.

Keywords: Dijkstra's algorithm, Bidirectional search algorithm, Multidirectional Dijkstra's algorithm, Shortest route problem, Emergency response services

1. Introduction

The shortest path algorithms [1] are popular in finding the optimal route for graph theory, networking or telecommunications, and road networks. In addition, these algorithms are applied in applications that automatically find directions between physical locations, such as navigation or driving apps, and websites like Google Maps, OpenStreetMap, or MapQuest. These applications require the fastest possible algorithm for finding directions, as they are performed in real-time.

In graph theory, the shortest path problem refers to finding the optimal path between two vertices (or nodes) wherein the sum of the weights of their edges is optimized [2][3]. In a road map, the shortest path problem refers to finding the shortest path between intersections (*i.e.*, vertices in a graph theory), which are weighted by the distance or length of their road segments. The intersections in a road map correspond to the vertices of the graphs, and the road segment between the two intersections corresponds to each edge of the graph. The weight of a graph edge between vertices may refer to the distance between

[1] College of Computer Studies, University of Antique, Sibalom, Antique, Philippines
Email: jeacarzamora@yahoo.com
[2] College of Computer Studies, University of Antique, Sibalom, Antique, Philippines
Email: eysangeles@gmail.com
[3] College of Computer Studies, University of Antique, Sibalom, Antique, Philippines
Email: marnei.manalo@gmail.com

Received [December 28, 2021]; Revised [February 25, 2022]; Accepted [April 8, 2022]

Published by InnoCon Publishing.
ISSN 2704-4440

the two intersections in a road map or the length associated with the road segment (*i.e.*, this length can also correspond to the time to traverse the segment or the cost of traversing the segment).

In networking or telecommunications, this theory can be analogous to route optimization, which deals with the min-delay path problem in transmitting data packets [4]. The optimized route means that it is the shortest path with the minimum number of nodes to traverse and minimum latency (*i.e.*, time consumed in transmitting data packets). The vertices of the graph may correspond to the routers that data packets may traverse, and the weight on the edge of the graph may correspond to the distance between the routers or the time required to transmit the data packets.

In this study, the performance of the various shortest route algorithms, including the regular Dijkstra's algorithm, Bidirectional search algorithm, and the Multidirectional Dijkstra's algorithm (MDA), was analyzed to find the shortest possible route for an emergency vehicle to travel from a starting place to the nearest destination. Specifically, this comparative analysis aims to: (1) solve the mean running time between regular Dijkstra's algorithm, bidirectional search and MDA in solving the shortest route in a graph when grouped by number of nodes, number of edges, and number of weights; (2) prove the reliability of MDA in solving the shortest route in terms of searching for multiple destinations simultaneously; and (3) determine the significant difference between natural Dijkstra's algorithm, bidirectional search, and MDA.

The remainder of this paper is organized as follows: Section 2 outlines the different shortest route algorithms; Section 3 details the methodology used in the analysis; the results and discussion are detailed in Section 4; and Section 5 concludes the study.

2. The Shortest Route Problem Algorithms

This section includes discussions on the overview of the various shortest route algorithms, which include Dijkstra's algorithm, the Bidirectional Search algorithm, and the MDA.

2.1 Dijkstra's Algorithm

Dijkstra's algorithm, conceived by computer scientist Edsger Dijkstra in 1956 and published in 1959, is a graph search algorithm that solves the single-source shortest path problem for a graph with non-negative edge path costs, producing a shortest path tree [5][6]. This algorithm is often used in routing and as a subroutine in other graph algorithms [7][8]. The Dijkstra's algorithm is shown in the pseudocode presented in Figure 1.

```
dist[s] ←0                          (distance to source vertex is zero)
for all v ∈ V–{s}
    do dist[v] ←∞                   (set all other distances to infinity)
S←∅                                 (S, the set of visited vertices is initially empty)
Q←V                                 (Q, the queue initially contains all vertices)
while Q ≠∅                          (while the queue is not empty)
do  u ← mindistance(Q,dist)         (select the element of Q with the min. distance)
   S←S∪{u}                          (add u to list of visited vertices)
   for all v ∈ neighbors[u]
      do if  dist[v] > dist[u] + w(u, v)        (if new shortest path found)
            then    d[v] ←d[u] + w(u, v)       (set new value of shortest path)
                                               (if desired, add traceback code)
return dist
```

Figure 1. Dijkstra's shortest path Algorithm

Stepwise procedure of Dijkstra's algorithm:

- Step 1: The source node is initialized and can be indicated as a filled circle.
- Step 2: The initial path cost to neighboring nodes (adjacent nodes) or link cost is computed, and these nodes are relabeled considering the source node.
- Step 3: Examine all adjacent nodes and find the smallest label; make it permanent.
- Step 4: The smallest label is now the working node, then Steps 2 and 3 are repeated till the destination node is reached.

2.2 Bidirectional Search Algorithm

The Bidirectional search algorithm refers to a graph search algorithm that is used to find the shortest path from an initial vertex to a destination vertex in a directed graph [9][10]. The algorithm runs two simultaneous searches, as depicted in Figure 2: one forward search from the initial vertex (origin) and one backward search from the destination vertex (destination), and stops the process when the two searches meet in the middle [11].

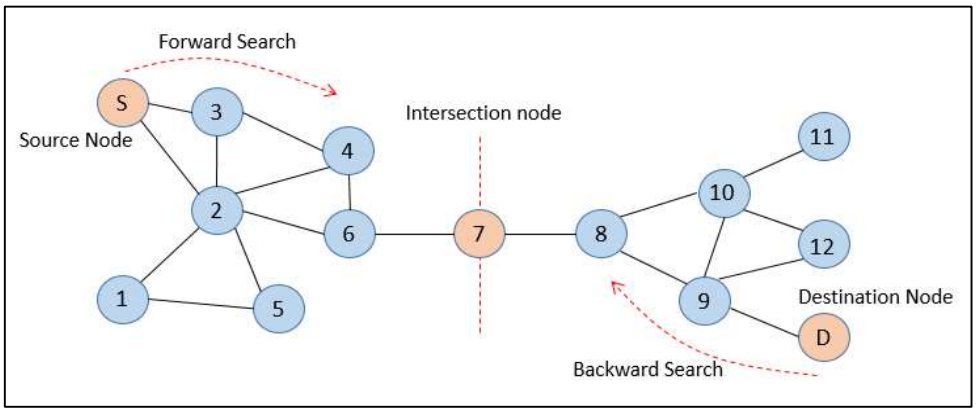

Figure 2. Search Paths from Start Address and Goal Address Connect to Finish the Search [12]

2.3 Multi-Directional Dijkstra's Algorithm

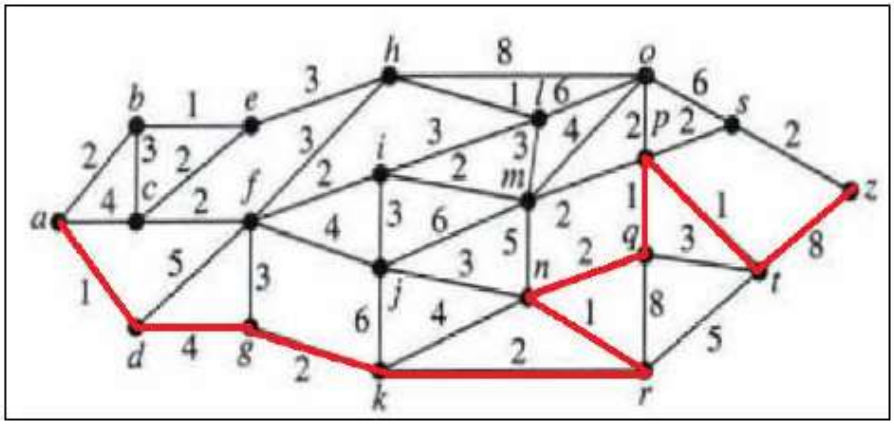

Figure 3. Shortest path in applied Multidirectional Dijkstra's Algorithm [12]

Multidirectional Dijkstra's Algorithm (MDA), as shown in Figure 3, is a hybrid algorithm based on the well-known shortest path algorithm, Dijkstra's algorithm, with the combination of a Bidirectional search algorithm [7][13][14]. The MDA is a greedy algorithm that searches all possible routes; this characteristic comes from the natural Dijkstra's algorithm. This will find the shortest route by starting at both the origin and multiple capable destinations to minimize the time of searching, as with the Bidirectional search algorithm [11]. This search will stop after the path of the starting point (*i.e.*, forward search) meets the path of the destination point (*i.e.*, backward search), and it will result in the shortest possible weight after the search meets both parties.

3. Methodology

In order to analyze the performance of different algorithms or implementations, including the Dijkstra, Bidirectional search, and MDA algorithms, instances of road networks (graphs) of different sizes and complexity are needed, as shown in Table 1. Road networks (graphs) were generated, where each station (node) (u; v) with u, v = 1, ..., n is located at coordinates (x; y) = ($u \cdot g + rnd(g)$; $v \cdot d + rnd(g)$). This basically describes a squared grid with grid size g, where the stations are translated out of their regular position by a random distance $rnd(g)$ between 0 and g. Each station that is not located on the boundary of the grid is connected by edges with its four direct neighbors (u−1; v), (u+1; v), (u, v−1), and (u; v+1). A list of 10.000 stations as an order list was then generated randomly to obtain three tables to be used as input data for all simulations that will be under consideration:

- Stations: List of all stations (nodes) in the graph with coordinates.
- Tracks: List of all tracks (edges) in the graph with the two connected stations and the length of the track.
- Transport orders: List of 10,000 stations to be visited subsequently by one single vehicle.

Table 1. Problem Instances for the Empirical Study

number of nodes	number of edges	avg. length of shortest path (num. of nodes)	number of sub-graphs	avg. number of edges between adjacent sub-graphs
$10 \times 10 = 100$	180	6.67	1	-
$25 \times 25 = 625$	1,200	16.67	1	-
$34 \times 34 = 1,156$	2,245	22.67	2	34
$41 \times 41 = 1,681$	3,282	27.33	4	20.5
$47 \times 47 = 2,209$	4,327	31.33	4	23.5
$52 \times 52 = 2,704$	5,308	34.67	4	26
$57 \times 57 = 3,249$	6,389	38	6	23.75
$81 \times 81 = 6,561$	12,966	54	12	23.63
$98 \times 98 = 9,604$	19,019	65.33	16	24.5
$114 \times 114 = 12,996$	25,772	76	20	25.65
$127 \times 127 = 16,129$	32,013	84.67	25	25.4
$139 \times 139 = 19,321$	38,374	92.67	30	25.48
$150 \times 150 = 22,500$	44,711	100	36	25

A second set of network instances is designed for analyzing transport networks containing sparsely connected clusters of densely connected stations. To obtain such instances, the original graphs were divided into sub-graphs and eliminated the connecting edges between adjacent sub-graphs. For graphs with more than 2,000 nodes, sub-graphs were defined with an average size of 610 nodes, where the number of sub-graphs and the average number of connecting edges between adjacent sub-graphs is given. Starting with these graph structures, the number of connecting edges between adjacent sub-graphs were reduced.

The algorithm will be tested through their running time on different file sizes on the same personal computer (PC), different file sizes on different PCs, and considering the Operating System (OS) of two computers used in testing. Five instance will be tested with different categories to get the average of the ten trials for speed efficiency: Fire Station, Search and Rescue, Police Station, Hospital, and Home.

The Chi-Test using Cramer's *V* Coefficient (*V*) is used to measure the strength of association between distance and time. It is used to determine the strength of the relationship regarding time in finding the shortest path and distance. Chi-squared values tend to increase with the number of cells; the lower the Cramer's *V* value, the better the performance of the algorithm.

$$V = \sqrt{\frac{X^2}{n(q-1)}} \tag{1}$$

The formula for Cramer's value is depicted in Equation 1, where X^2 is the Chi-squared value, n is the grand total of observations, and q is taken as the number of optional outcomes, and it functions as a measure of tendency towards a single outcome.

In this study, there are two sets of tests to be performed. First is the testing for speed, and second is the chi-square statistics using Cramer's *V* to measure the strength of the association between the distance and the time in finding the shortest path.

Some factors in the running of an application can be influenced by the hardware used, the programming language in which the program is encoded, and the operating system of the computer. The algorithms must be encoded in the same programming language and in the same manner, and they must run on the same PC. Specifically, the algorithms were encoded in Microsoft Visual Basic 6 and were tested on one computer, the Acer Intel N475 Atom with a processing speed of 1.83 GHz, 2 GB of RAM, and a Microsoft Windows Starter operating system (OS).

4. Results and Discussion

4.1 Test Results during Light Traffic Process

Three locations were tested ten times to determine the average speed during the light traffic process. The results of the speed tests are shown in Table 2. Based on the findings of the collected data, the mean speed of light traffic regarding finding the shortest path route, MDA, has the lowest time. The mean speed of Bidirectional search is 10.2795s, while the mean speed of Dijkstra is 9.7854s, and in MDA it is 8.06338s.

Table 2. Summary of the Mean Speed During Search with Regards to Distance (Light Traffic)

	Bidirectional	**Dijkstra**	**Multidirectional**
Fire	9.8351	9.3951	7.32121
Police	10.8678	10.1052	8.71537
Hospital	10.1355	9.8559	8.15358
Average	10.2795	9.7854	8.06338

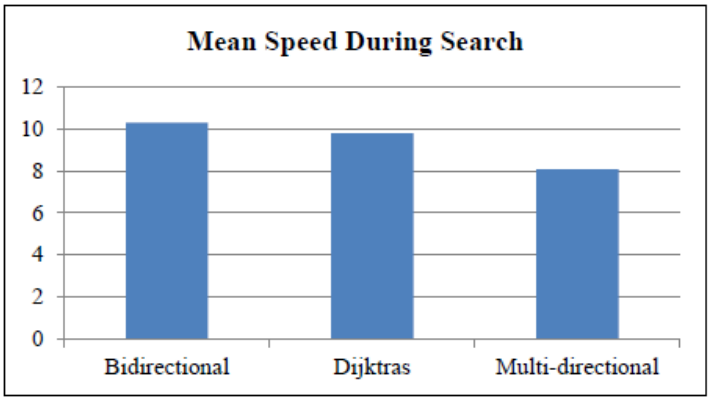

Figure 4. Summary of the Mean Speed During Search with Regards to Distance (Light Traffic)

The result shows that after undergoing ten trials of searching with three different algorithms, the Bidirectional search algorithm has the slowest mean speed, as shown in Figure 4, compared to the speed of the Dijkstra and MDA algorithms.

4.2 Test Results During the Heavy Traffic Process

Three locations were tested ten times to determine the average speed during the heavy traffic process. The results of the speed tests are shown in Table 3. The mean speed in the heavy traffic process of Bidirectional search is 10.3379s, Dijkstra has 9.8371s, and the MDA has a mean speed of 7.8020s.

Table 3. Summary of the Mean Speed During Search with Regards to Distance (Heavy Traffic)

	Bidirectional	**Dijkstra**	**Multidirectional**
Fire	9.9226	9.5652	7.5000
Police	10.1179	9.7922	7.9015
Hospital	10.9733	10.1534	8.0045
Average	10.3379	9.8371	7.8020

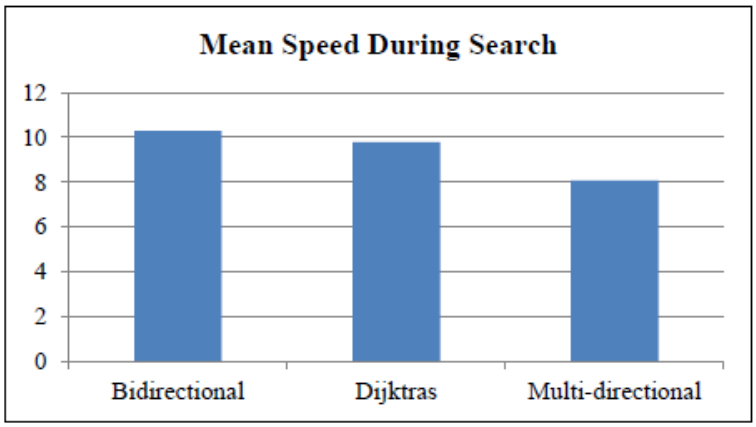

Figure 5. Summary of the Mean Speed During Search with Regards to Distance (Heavy Traffic)

Results show that after undergoing ten trials of searching with three different algorithms, the Bidirectional search algorithm has the slowest mean speed compared to the speed of the Dijkstra and MDA algorithms, as shown in Figure 5.

Table 4. F-TEST Results of the Mean Speed During the Light Traffic Process

Source of Variation	SS	df	MS	F	*p*-value	F-Critical
Between Groups	8.1204	2	4.0602	13.4748	0.006038	5.1433
Within Groups	1.8079	6	0.3013			
Total	9.9284	8				

To ascertain the significant difference in the mean speed of the three algorithms with regards to finding the shortest path process for light traffic, an F-Test or ANOVA was used, where the level of significance was set at 0.05. As reflected in Table 4, the computed p-value was 0.006038. Since the *p*-value is less than the level of significance, the null hypothesis was rejected. Therefore, there is a significant difference in the speed of the three algorithms with regards to finding the shortest path for light traffic.

Table 5. F-TEST of the Mean Speed During the Heavy Traffic Process

Source of Variation	SS	df	MS	F	*p*-value	F-Critical
Between Groups	10.8229	2	5.4114	34.4229	0.000515	5.1432
Within Groups	0.9430	6	0.1571			
Total	11.7661	8				

To ascertain the significant difference in the mean speed of the three algorithms with regards to finding the shortest path process for heavy traffic, an F-Test or ANOVA was used, where the level of significance was set at 0.05. As reflected in Table 5, the computed *p*-value was 0.000515, which is less than the level of significance, hence, the null hypothesis was rejected. Therefore, there is a significant difference in the speed of the three algorithms with regards to finding the shortest path for heavy traffic.

4.3 Summary of Cramer's V Result for the Shortest Path

The Chi-Square results are processed using the data analysis in MS Excel 2007 (Mega Stat). Table 6 summarizes the result of the Chi-Square test using Cramer's *V*. The mean Cramer's *V* for light traffic using Bidirectional search and Dijkstra was quite the same, but the MDA has the lowest value. The average Cramer's *V* of Bidirectional was 31%, the Dijkstra had 29%, and the MDA had 21%. This means that the MDA with a lower value is an efficient solution for finding the shortest route.

Table 6. Mean Result of the Cramer's *V* During Encryption Process

Distance	Bidirectional	Dijkstra	MDA
5	32%	32%	21%
10	29%	28%	21%
15	32%	28%	22%
20	32%	27%	21%
Average	31%	29%	21%

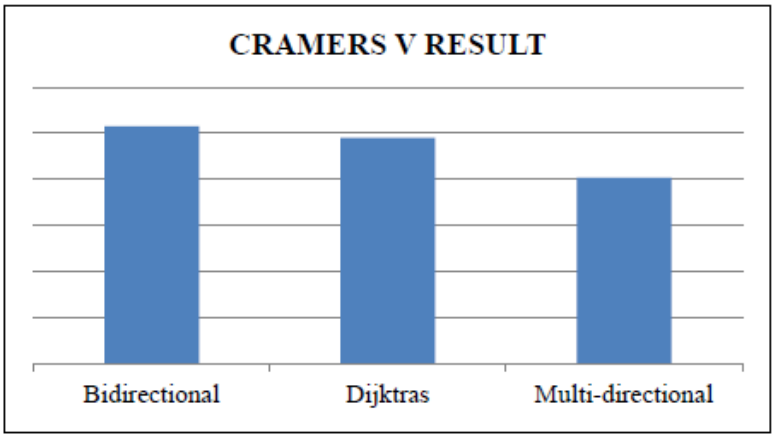

Figure 6. Average of Cramer's *V* Test

Results show that the MDA has the lesser value among the three Crammers *V* tests, as shown in Figure 6, which means that the MDA is more efficient as compared with both the Dijkstra's and Bidirectional search algorithms.

4.4 Statistical Result of the Three Algorithms with Regards to the Shortest Path

Table 7. F-test result of the mean Cramer's *V*

Source of Variation	SS	df	MS	F	*p*-value	F-Critical
Between Groups	246838	3	82279.33	8.073131	0.008371	4.066181
Within Groups	81534	8	10191.75			

To ascertain the significant difference in the mean Cramer's *V* of the three algorithms with regards to finding the shortest path for the light traffic process, an F-Test or ANOVA was used, where the level of significance was set at 0.05. As reflected in Table 7, the computed *p*-value was 0.008371. Since the *p*-value is less than the level of significance, the null hypothesis was rejected. Therefore, there was a significant difference in the mean Cramer's *V* of the three algorithms with regards to the light traffic process.

5. Conclusion

This study has analyzed the various shortest route algorithms that include the Dijkstra, Bidirectional Search, and MDA algorithms in determining the shortest route for emergency departments and services for faster transport from one place to their destinations. Based on the results, integrating the MDA algorithm using a map to determine the shortest route will have a good benefit for the road user when traveling on emergency transportation because it has a user friendly interface that is very understandable and easy to use. It can also give the user the shortest possible route with distance, as compared with the Bidirectional search and Dijkstra's algorithms, and can really be helpful for emergency vehicles for a faster response.

Based on the findings of the collected data, the mean speed of the three algorithms in finding the shortest path was not equal; that is, the *p*-value was smaller than the computed *p*-value. Hence, the first null hypothesis was rejected. In addition, during the heavy traffic process, the mean speed of the three algorithms regarding file size was not equal. The *p*-value was smaller than the computed *p*-value. Hence, the second null hypothesis was rejected. Moreover, for determining the average Cramer's *V*, the results were not equal, the *p*-value was less than the level of significance. Thus, there is a significant difference in the average Cramer's *V* value; therefore, the final null hypothesis is rejected.

This study will be a great help for the Emergency Response Services for a faster response, but this study does not cover any technical issues such as damaged vehicles, ongoing road or bridge construction, or blocked roads.

References

[1] K. Mehlhorn and P. Sanders, "*Chapter 10: Shortest Paths*", in Algorithms and Data Structures: The Basic Toolbox, Berlin, Heidelberg, Germany, Springer, 2008, pp.191-215, doi: 10.1007/978-3-540-77978-0, ISBN 978-3-540-77977-3.

[2] V. Joshi, "*A Gentle Introduction to Graph Theory*", Medium, www.medium.com/basecs/a-gentle-introduction-to-graph-theory-77969829ead8 (Accessed December 5, 2022).

[3] R. J. Wilson, "*Introduction to Graph Theory*", 4th Edition, Addison Wesley Longman Limited, England, 1996, ISBN-13: 978-0582249936.

[4] M. Loem, "*Network Optimization (1): Shortest Path Problem*", Medium, www.medium.com/swlh/network-optimization-1-shortest-path-problem-3757a67a129c (Accessed December 5, 2022).

[5] E. W. Dijkstra, "*A Note on Two Problems in Connexion with Graphs*", Numerische Mathematik, vol. 1, December 1959, pp.269-271, doi: 10.1007/BF01386390.

[6] P. L. Frana and T. J. Misa, "*An Interview with Edsger W. Dijkstra*", Communications of the ACM, vol. 53, no. 8, August 2010, pp.41-47, doi: 10.1145/1787234.1787249.

[7] M. Sniedovich, "*Dijkstra's Algorithm Revisited: the OR/MS Connexion*", ifors.ms.unimelb.edu.au, www.ifors.ms.unimelb.edu.au/tutorial/dijkstra_new/ (Accessed November 12, 2019).

[8] Programmer All, "*Dijkstra Algorithm*", www.programmerall.com/article/224531575/ (Accessed November 12, 2019).

[9] A. V. Goldberg, "*Point-to-Point Shortest Path Algorithms with Preprocessing*", in SOFSEM 2007: *Theory and Practice of Computer Science*, J. van Leeuwen, G. F. Italiano, W. van der Hoek, C. Meinel, H. Sack, F. Plášil, (Eds.), Lecture Notes in Computer Science, vol. 4362, Berlin, Heidelberg, Germany, Springer, 2007, doi: 10.1007/978-3-540-69507-3_6.

[10] Wikipedia, "*Artificial Intelligence/Search/Heuristic search/Bidirectional Search*", www.en.wikibooks.org/wiki/Artificial_Intelligence/Search/Heuristic_search/Bidirectional_Search (Accessed November 12, 2019).

[11] H. Kaindl and G. Kainz, "*Bidirectional Heuristic Search Reconsidered*", Journal of Artificial Intelligence Research, vol. 7, December 1997, pp. 283-317, doi: 10.1613/jair.460.

[12] J. D. Zamora, A. M. F. E. Angeles, M. A. Manalo, "*A Study of the Multidirectional Dijkstra's Algorithm in Solving the Shortest Route Problem*", Journal of Innovative Technology Convergence, vol. 2, no. 1, June 2020, pp. 1-10, doi: 10.69478/JITC2020v2n1a01.

[13] D. Atzmon, J. Li, A. Felner, E. Nachmani, S. Shperberg, N. Sturtevant, S. Koenig, "*Multi-Directional Heuristic Search*", in Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, May 2020, pp. 4062-4068, doi: 10.24963/ijcai.2020/56210.24963/ijcai.2020/558.

[14] R. C. Prim, "*Shortest Connection Networks and Some Generalizations*", Bell System Technical Journal, vol. 36, no. 6, November 1957, pp.1389-1401, doi: 10.1002/j.1538-7305.1957.tb01515.x.