Performance of A Star with Dynamic Programming Algorithms in Determining the Shortest Route

Henny Yusnita[1], Muhammad Zarlis[1*] and Sawaluddin[2]
[1]Faculty of Computer Science and Information Technology, Universitas Sumatera Utara, Indonesia
[2]Faculty of Math and Natural Science, Universitas Sumatera Utara

Keywords: Shortest path, dynamic programming, shortest route.

Abstract: A Star's algorithms tend to perform better when combined with heuristics. By implementing the heuristic function-based Best First Search (BFS) search mechanism, the system will find the shortest path from the origin to the destination nodes. In the Dynamic Programming algorithm, the search for the shortest route was based on interconnected stages by loading the route to be searched. In this study, a combination of the A Star algorithm and Dynamic Programming methods was used. Testing of A Star Algorithm with Dynamic Programming in the search for the shortest route from Pasar Petisah to Pasar Lau Cih was chosen. The results of the study found that the A Star - Dynamic Programming algorithm produced the shortest distance of 14.12 Km.

1 INTRODUCTION

The A Star algorithm is a development of the Dijkstra algorithm. To get the best results, a route search design by analyzing inputs and testing potential routes is carried out. In general, this algorithm uses traversal graphs and possible routes when searching for a route around a node. The basic characteristic of the algorithm used is the application of the Open List. The Open List is a node that has probably been skipped, has never been traversed, or is not a dead-end route. The Close List is an impassable node, such as a dead end or has been passed. Thus, the Close List function aims to re-analyze nodes that have been passed previously. Therefore, the route search algorithm becomes faster and does not repeat itself indefinitely. The closest route search algorithm will stop when the destination node has been found or the node in the Open List no longer exists. Implementation of Close List and Open List in finding the shortest route with a small number of nodes becomes less productive, which leads to the use of more complex algorithms (Taufiq, 2015).

Sharma & Pal (2015) compared the closest route search using the A Star algorithm and Dijkstra's algorithm. This study shows that the A Star algorithm performs better than the Dijkstra algorithm in all cases, where weights and distances are not limited. The A Star algorithm is proven to provide the best search results compared to others. Further research entitled Guided Hybrid A Star Path Planning Algorithm for Valet Parking Applications was piloted by Sedighi (2019). This research presents an innovative and computationally efficient approach to combining the search engine and the famous Hybrid A Star with visibility diagrams in the search for the shortest possible non-holonomic path in a hybrid (continuous discrete) atmosphere for valet parking. In the study, the Visibility Diagram was applied to introduce a cost saving function as well as for the Hybrid A Star algorithm. Furthermore, the shortest path found is used to provide the correct waypoint so that Hybrid A Star can plan the optimal route by taking into account non-holonomic constraints. This method has been researched extensively with results 40% faster than the Hybrid A-star algorithm or an average of 20%.

Karova & Penev's (2016) conducted a comparative study between A Star Algorithm and backtracking algorithms to find the shortest route of a maze. The results reveal that the backtracking algorithm can be used effectively in small mazes with few constraints. Likewise, it can also trace the shortest route for a shorter time. For larger size and number of obstacles in the maze, the algorithm produces the best results. In several trials, this

Yusnita, H., Zarlis, M. and Sawaluddin, .
Performance of A Star with Dynamic Programming Algorithms in Determining the Shortest Route.
DOI: 10.5220/0010335700003051
In *Proceedings of the International Conference on Culture Heritage, Education, Sustainable Tourism, and Innovation Technologies (CESIT 2020)*, pages 536-542
ISBN: 978-989-758-501-2

algorithm produced a shorter route in a shorter time compared to the other two algorithms. The genetic algorithm can be applied in some cases, where the computation time required is shorter compared to the shorter route. Obstacle removal will be simulated during the real-time search process starting from 10% to 30%. With such an algorithm, the robot will be able to move in a dynamically changing environment.

In his research entitled Improving A Star Hierarchy Algorithm for Optimal Parking Path Plan from Large Parking Spaces, Cheng & Yan (2014) revealed that optimal parking lane planning in a large parking area. The optimal evaluation index is determined in a short period of time and then increases. Given the situation, the corresponding evaluation function is well developed by the A Star plus algorithm, which results in better, more efficient and more accurate searches. The evaluation index in the parking lane planning from the parking lot is the optimal time rather than the optimal distance. The algorithm must be able to avoid congested roads at the same time. The experimental results show that the built algorithm can effectively plot different optimal path information with additional attributive data parameters.

Dynamic Programming is a method for finding the shortest route by dividing the search area and stages. In this case, the route obtained from each region is considered a series of fastest paths that support each other. Dynamic Programming can identify and solve multiple problems, including work scheduling, optimal operation division, shortest route, and workforce size. When getting optimal results, the Dynamic Programming algorithm applies the principle of optimality by dividing the problem into several parts called stages. The set is then resolved with optimization until all cases are resolved. The best decision is a decision body made of all stages, also called optimal decisions (Fawwaz, 2019).

This study aims to determine, analyze, and study how the Dynamic Programming algorithm works integrated with the A Star algorithm to identify the shortest route on a city graph with certain limitations. Solve the shortest route problem using advanced recursive dynamic programming. The route discussed in this study is the route from Pasar Petisah to Pasar Lau Cih. However, it does not discuss road congestion, road sections, road conditions and road priorities. The routes covered in the analysis are land routes with good conditions that can be traversed by motorized vehicles.

The AI applied in this research is the Dynamic Programming algorithm to find a path when attacking opposing participants in a game. Police figures are tested in a game. From 10 experiments, he obtained 90% success in finding the shortest and optimal path with the Dynamic Programming algorithm (Fawwaz, 2019).

Dynamic Programming (DP) is an optimization algorithm that can be applied in everyday life. The DP algorithm is carried out in several stages so that problems can be perceived through a series of interdependent decisions. In solving the problem, the DP algorithm carries out two approaches, namely Forward and Backward so that the problem can be resolved as a whole by using these two procedures. Even though the final optimal value is the same, the achievement of the optimal value at each stage between the Forward and Backward approaches is different. Differences in absolute optimal values are obtained under the same conditions if resolved using Forward and Backward strategies (Kamal, 2017).

In this study, the proposed strategy provides a popular file-to-cache combination that maximizes the likelihood of optimal cache being found with pseudo-polynomial time complexity. For these purposes, the resource algorithm is known as the 0/1-Knapsack problem assuming that files are cached without the partition being used (Ayenew, 2018).

The A Star algorithm is a classic, efficient algorithm for searching the shortest path on a graph. The productivity of the algorithm depends on an evaluation function that estimates the heuristic value of the shortest path from the starting point to the destination. If you know the coordinates of the vertices (x, y), the heuristic value of the shortest path becomes the distance. This study presents the Index-Based A Star algorithm which aims to solve the shortest path problem on directed and weighted graphs with unknown vertices of coordinates. The development is done by using three indexes for each node, namely the earliest, second, and newest arrival index. Thus, the lower limit and upper limit of the shortest distance from the starting point to the target are based on these three indices. This algorithm makes use of the earliest arrival index to define the evaluation function of the A Star algorithm and applies three indexes to unused nodes to improve algorithm performance. Compared with the A Star algorithm, the additional time and space complexity of the experimental results proves the effectiveness of the proposed design (Lie, 2018).

The investigation aims to find out how the A Star algorithm determines the shortest route during traffic jams. The results of the experiment provide a more realistic picture of the A Star algorithm process in determining the closest path (Syukriyah & Solihin, 2016).

1.1 A Star Algorithm

Nils Nilsson developed the A star algorithm in 1964 based on Dijkstra's algorithm with the name A1 algorithm. In 1967, Bertram Raphael refined the first algorithm to become the A2 algorithm. However, in 1968, Peter E. managed to prove the supremacy of the A2 algorithm over the previous version. Thus, the A2 algorithm is claimed to be the most optimal algorithm in finding the closest route and is renamed to A Star. Based on time, the A Star algorithm is better than Dijkstra's algorithm which uses a heuristic search function. The A Star algorithm is the most frequently used algorithm in route finding and graph exploration, which plots the most efficient paths between points called nodes. This algorithm looks for a path or route very efficiently where this algorithm is also developed from the basic method of Best First Search (Syukriyah, 2016).

The main principle of the A Star algorithm is to find the shortest path from the starting point to the destination node which underlies the shortest distance and lowest cost. This algorithm process begins by placing A at the starting point and entering all neighboring nodes, without the obstacle attribute with A into the Open List. Then, the system identifies the smallest H node value in the Open List, and then moves A to the smallest H node value. The nodes before A are stored as the primary of A and are then classified into a Closed List. If another node has been relocated adjacent to A but is not a member of the Open List, that node will be categorized as Open List. After that, compare the existing G value with the previous one. If the previous G value was smaller, then A will return to the starting position. Node is designed to enter the Close List. These steps can be rerun until a solution is found or no other nodes are available in the Open List. Figure 1 illustrates how the concept of the A Star algorithm works.

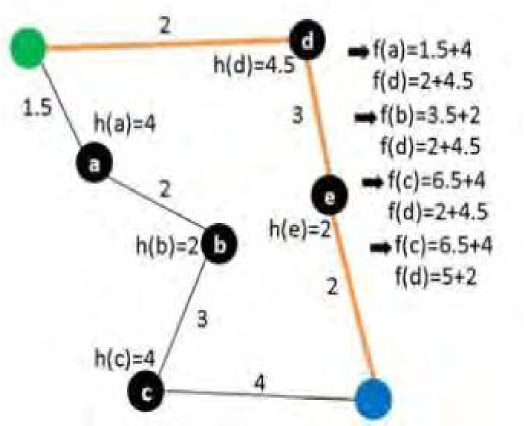

Figure 1: A Star Algorithm.

The Star algorithm has two main functions when determining the best solution. The first function is as g (x) which functions to calculate the total cost (distance or time) required from origin to destination nodes. The second function is as h (x) to estimate the total cost estimated from origin node to destination node.

$$G(n) = \sqrt{Xn^2 + Yn^2} \quad (1)$$

$$H(n) = |X(target) – X(n)| + |Y(target) – Y(n)| \quad (2)$$

$$F(n) = G(n)+F(n) \quad (3)$$

1.2 Dynamic Programming

Dynamic Programming was first introduced by Richard Bellman in the 1940's. This describes the problem-solving process, where the desired solution is the most optimal, often referred to as the global optimal (Ayenew, 2018). The optimal substructure means that the optimal solution for the previous sub-problem can be used to find the optimal solution for the problem at hand. In the graph, for example, the shortest path from origin to destination node can be found by first calculating the shortest subpath from all neighboring origin nodes to the destination node and then using the shortest subpath to obtain the shortest complete path. There are three stages in problem solving with an optimal substructure, namely:

1. Break the problem into smaller sub-problems,
2. Find the optimal solution to the sub-problem using these three steps recursively,
3. Use the optimal solution for the sub-problem to build the optimal solution for the problem at hand.

The problem-solving process is continued by dividing it into smaller parts until a simple case can be solved quickly. Dynamic Programming is a method of solving problems by breaking down solutions into groups (stages). Problem solving can be observed from a series of interrelated decisions.

Dynamic Programming emerged because of its ability to analyze and document the results at each stage using multiple tables for more detailed solutions. The characteristics of problem solving with Dynamic Programming can be described as follows:

1. Several possible options,
2. Solutions at each stage are built from the results of the completion of the previous stage,
3. The algorithm takes optimization requirements and limits and limits the number of options to be considered at each stage.

In Dynamic Programming, optimal decisions are made on the principle of optimality. If the total solution is optimal, then the solution part to the k-stage must also be optimal. (Plancher, 2017).

In the principle of optimality, if you work from stage k to stage k + 1 we can use the optimal results from stage k without having to return to the initial stage where the cost at stage k + 1 is the cost generated in stage k plus the cost from stage k to stage k + 1 as shown in Figure. 2.

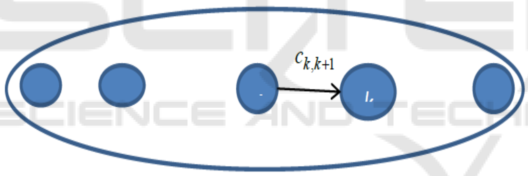

Figure 2: Principle of Optimality (Munir, 2013).

Dynamic Programming has the following characteristics:

1. The problem is divided into stages where only one decision is taken per set.
2. Each stage consists of a corresponding state and its state is a number of possible inputs at that stage.
3. The solution of each stage is declared with a status corresponding to the status of the next step in the next step.
4. The cost at each stage increases regularly as the number of stages increases.
5. The value at each stage depends on the current value.
6. The solution at each stage does not depend on the solution made in the previous stage.
7. The recursive relation marks the solution of each state at stage k, which provides the best solution for each state in the previous stage.

There are 2 (two) approaches in Dynamic Programming, namely:

1. The forward or up-down approach,
2. Backward or bottom-up approach.

For example x1, x2,…, xn represent the decision variables that must be made respectively for stages 1, 2, ..., n, then:

1. Dynamic Programming Forward moves from stage 1, continues to stage 2, 3, and so on until stage n where the sequence of decision variables is x1, x2, ..., xn.
2. Dynamic Programming backward moves from stage n, continues backward to stages n - 1, n - 2 and so on until stage 1. The set of decision variables is xn, xn-1, ..., x1. stage k + 1 = (costs generated in stage k) + (costs from stage k to stage k + 1), k = 1, 2, ..., n - 1.

While the principle of optimality in backward dynamic programming is the cost at stage k = (costs generated in stage k + 1) + (costs from stage k + 1 to stage k), k = n, n - 1, .., 1.

Following are the steps to develop the Dynamic Programming algorithm:

a) Perform the structural characteristics of the optimal solution,
b) Recursively determine the optimal solution value,
c) Calculate the optimal solution value for forward or backward,
d) Develop optimal solutions.

As for the principle, Dynamic Programming is based on multi-stage graph. Each node in the graph represents a state, while V1, V2, ... represent a stage. Figure 3 shows a multi-stage process.

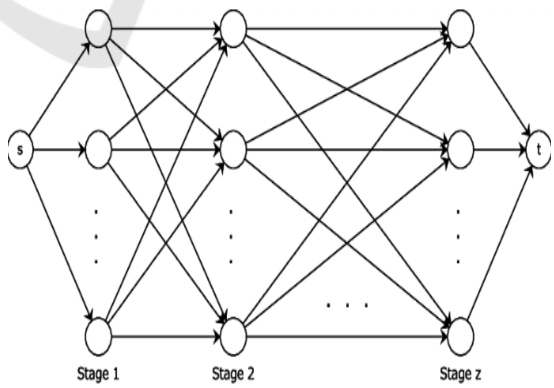

Figure 3: Multi-Stage Graph.

Several problems in the Multi-Layout Graph related to Dynamic Programming are categorized as follows:

1. Stage (k) is the process of selecting the destination node as shown in Figure 3; there are five stages.
2. The status associated with each stage is a node in the graph.

1.3 Research Design

The research design is to analyze the Dynamic Programming method - A Star to find the shortest route. The research design is presented in Figure 4.

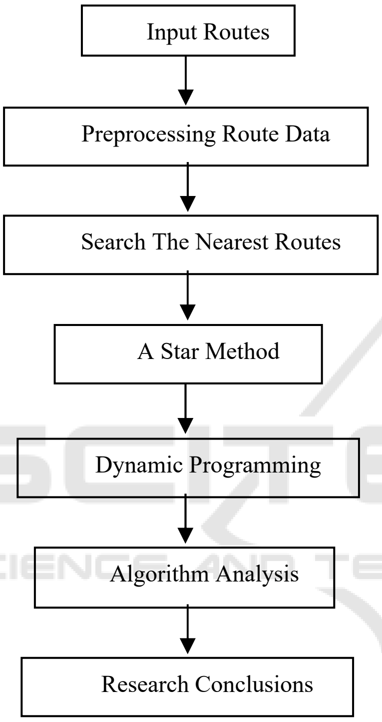

Figure 4: Research Design.

Initially the user enters the route as data input and the initial process is carried out by the route processing software. After the processing of the route is complete, the search results performed using the A Star algorithm produce the shortest route distance. Combination with Dynamic Programming algorithm is then performed.

1.4 Problem Identification

Based on the previous explanation, the problem lies with the A Star algorithm which finds the shortest possible route according to the standard rules regardless of the whole problem. This algorithm works thoroughly against all available alternatives. So as to produce solutions and processing time for relatively small problems, which are less than optimal. Like the A Star algorithm in finding a solution for each iteration, it forms two arrays: Open and Close List as temporary data stores that require memory resources.

2 RESEARCH METHOD

This study discusses the shortest route search method with the Dynamic Programming algorithm on the A Star algorithm on a dataset in the form of a route between the Petisah market to the Lau Cih market. There are several possible routes to the intended destination, namely through the Captain Patimura-Jamin Ginting road, through the Captain Patimura road - Dr Mansyur-Setia Budi Street and Captain Patimura Street - Jamin Ginting Street - Harmonika-Setia Budi Street. Following are the steps to analyze the shortest route search using Dynamic Programming on the A Star algorithm:

1. Perform a search for the shortest distance using the A Star algorithm.
2. Perform a search for the closest distance with the A Star-Dynamic Programming algorithm.
3. Perform an analysis to evaluate which of the two methods above gives the best results.

3 RESULTS AND DISCUSSIONS

3.1 Results

The optimal route search test results for each algorithm can be seen in table 1, table 2 and table 3.

1. Testing Results with A Star Algorithm

Table 1: Test Results of A Star Algorithm.

No	Route	Distance (Km)
1	Route -1	15.2
2	Route -2	14
3	Route -3	15
4	Route -4	14
Average		**14.57**

2. Testing Algorithm with Dynamic Programming.

Table 2: Results of Testing Dynamic Programming Algorithms.

No	Route	Distance (Km)
1	Route -1	14.7
2	Route -2	14

3	Route -3	14.2
4	Route -4	14
Average		**14.22**

3. Testing algorithm with Dynamic Programming - A Star.

Table 3: Test Results of Dynamic Programming - A Star Algorithm.

No	Route	Distance (Km)
1	Route -1	14.5
2	Route -2	13.3
3	Route -3	14.7
4	Route -4	14
Average		**14.12**

3.2 Discussions

In the Table 1, the search process for the shortest distance from the separating market to the Lau Cih market was carried out with four routes using the A star algorithm, the test process for each route is carried out 10 times so that the test results are obtained with an average of 14.57 km. In Table 2, the search process with Dynamic Programming algorithm was also carried out by testing each route 10 times with the four routes so that an average of 14.22 km was obtained. Table 3, In the Dynamic Programming - A Star algorithm, the testing process is carried out on each route 10 times on the four routes, so that an average of 14.12 km is obtained.

From the test results of each algorithm above through the route through Pasar Petisah to Pasar Lau Cih Medan with the testing process carried out 10 times on each route, it is found that Dynamic Programming - A Star is the most optimal route with an average distance of 14.12 Km. as shown in Table 4.

Table 4: Route Optimum Distance of Each Algorithm.

No	Algorithm	Distance (Km)
1	A Star	14.57
2	Dynamic Programming	14.22
3	Dynamic Programming - A Star	14.12

From Table 4, the optimal distance for each algorithm can be plotted as in the graph in Figure 5.

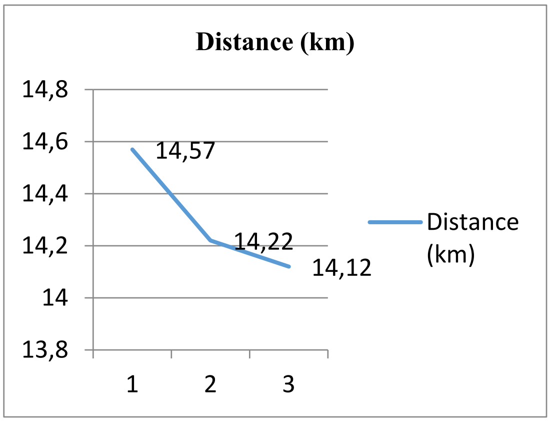

Figure 5: Optimum Distance of Each Algorithm.

From the graph above, it can be seen that Dynamic Programming - A Star algorithm is the most optimal route with an average distance of 14.12 Km

4 CONCLUSIONS

By using these three algorithms, it can be concluded that the shortest route from Petisah market to Lau Cih Medan market is as follows; the optimal route distance between the two zones is 14.12 km. The results of the three searches show that the closest distance is almost the same, but not significantly different. This is because the route you are looking for is a road with low complications or a limited number of nodes. The analysis of the three methods above requires further research that needs to be done by further researchers by adding more nodes and the weight of the congestion and difficulties when crossing the route.

REFERENCES

Abdussakir, Azizah, N. N & Nofandika, F. F., 2009. *Teori Graf*, UIN-Malang Press. Malang.

Ahmad, I. & Widodo, W., 2018. Penerapan Algoritma A Star (A*) pada Game Petualangan Labirin Berbasis Android, Jurnal Khazanah Informatika Vol. 3 No. 2 Desember 2017.

Andiany, F. E & Hadikurniawati, W., 2018. Implementasi Algoritma Dijkstra Untuk Mencari Rute Terpendek Antar Kantor Dan Estimasi Penggunaan Bahan Bakar Kendaraan (Studi Kasus PT. Telkom Indonesia Regional IV JATENG-DIY) Teknik Informatika, Fakultas Teknologi Informasi, Universitas Stikubank. Prosiding SENDI_U 2018 ISBN.

Ayenew, T. M., Xenakis, D., Passas, N. & Merakos, L., 2018. Dynamic Programming Based Content Placement Strategy for 5G and Beyond Cellular Networks. 2018 IEEE 23rd International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD).

Bazaraa, M. S. & Jarvis. J. J,. *Linear Programming and Network Flows*. John Wiley & Sons.

Bondy, J.A. & U.S.R Murty., 1976. *Graph Theory with Applications,* the Macmillan Press Ltd. London.

Chartrand & Gary Ortrud R. O., 1993. Applied and Algorithmic Graph Theory, McGraw. Hill. Inc.

Dreyfus, S.E & Averil, M. Law., 1977. *The Art and Theory Dynamic Programming,* Academic Press. London.

Evans. J.R., & Minieka. E., 1992. *Optimization Algorithms for Networks and Graphs,* Marcel Dekker, Inc.

Fawwaz, E., Winarta, A., Selvianna, Ramli, J. J & Waruwu, L. M., 2019. Penerapan Algoritma Dynamic *Programming* Pada Pergerakan Lawan Dalam Permainan *Police &Thief*, Journal of Informatics and Telecommunication Engineering, Edisi Januari 2019.

Ibrahim & Mussafi, N. S. M., 2013. *Pengantar Kombinatorika dan Teori Graf*, Graha Ilmu. Yogyakarta.

Jumadi., 2015. Penentuan Rute Terpendek Menuju Kampus Menggunakan Algoritma *Dynamic Programming*, Jurnal Tekno Edisi Juli 2014 Volume VIII No. 1 ISSN 1979-8911.al Ilmu Komputeran

Lipschutz, S. & Lipson, M., 2008. Matematika Diskrit. Edisi Ketiga, Penerbit Erlangga. Jakarta.

Plancher, B., Manchester, Z. & Kuindersma, S., 2017. Constrained Unscented Dynamic Programming. 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) September 24–28, 2017, Vancouver, BC, Canada.

Rahmadani, P. A., Saintika,Y. & Prabowo, W. A., 2018. Implementasi Metode Dynamic Programming Pada Sistem Pendukung Keputusan Untuk Menentukan Menu Diet Bagi Penderita Diabetes, Prosiding SENDI_U 2018 ISBN.

Setiawan, K., Supriyadin, Santoso, I. & Buana, R., 2018. Menghitung Rute Terpendek Menggunakan Algoritma A* Dengan Fungsi *Euclidean Distance,* Seminar Nasional Teknologi Informasi dan Komunikasi 2018 (SENTIKA 2018) ISSN: 2089-9815 Yogyakarta, 23-24 Maret 2018.

Syukriyah, Y., Falahah & Solihin., H. 2016. Penerapan Algoritma A* (Star) Untuk Mencari Rute Tercepat Dengan Hambatan. Seminar Nasional Telekomunikasi dan Informatika 2016.

Taufiq, P. J., Wibowo, A.T & Septiana, G., 2015. Implementasi Dan Analisis Algoritma A*(Star) Untuk Menentukan Jalur Dengan Multiple Goal Pada Pergerakan NPC (Non-Playable Character). E-Proceeding of Engineering: Vol.2, No.3 Desember 2015 Page 7799.

Duchon, A., Babineca, A., Kajana, M., Beno, P., Florek, M., Ficoa, T. & Jurišicaa, L., 2014. Path planning with modified A star algorithm for a mobile robot. Modelling of Mechanical and Mechatronic Systems MMaMS 2014. Elsevier journal Procedia Engineering 96 (2014) 59 – 69.

Gong, D. & Liu, X., 2016. A Comparative Study of A-star Algorithms for Search and rescue in Perfect Maze, IEEE International Journal School of Electronic Information and Control Engineering Beijing university of Technology.

Sharma, S. K. & Pal, B. L., 2015. Shortest Path Searching for Road Network using A* Algorithm. International Journal of Computer Science and Mobile Computing a Monthly Journal of Computer Science and Information Technology Vol. 4, Issue. 7, July 2015, pg.513 – 522.

Sedighi, S., Nguyen, D. V. & Kuhnert, K. D., 2019. Guided Hybrid A-star Path Planning Algorithm for Valet Parking Applications. Panasonic Automotive Systems Europe GmbH Langen, Germany. 2019 5th International Conference on Control, Automation and Robotics.

Firmansyah E. R., Masruroh, S. U. & Fahrianto, F., 2016. Comparative Analysis of A * and Basic Theta* Algorithm in Android Bases Pathfinding Games. 2016 6th International Conference on Information and Communication Technology for the Muslim World.

Karova, M. & Penev, I., 2016. Comparative analysis of algorithms to search for the shortest path in a maze, 2016 IEEE International Black Sea Conference on Communications and Networking (BlackSeaCom).

Cheng, L. & Yan, C. L. B., 2014. Improved Hierarchical A-star Algorithm for Optimal Parking Path Planning of the Large Parking Lot, Proceeding of the IEEE International Conference on Information and Automation Hailar, China, July 2014.