Implementation of the A Star Heuristic Search Algorithm in Determining the Shortest Path

Leni Marlina, Aswandi

1,3Department of Computer System, Universitas Pembangunan Panca Budi, Indonesia
2Department of Network Computer Engineering Technology, Politeknik Negeri Lhokseumawe, Indonesia

Article Info

ABSTRACT

Finding the shortest path in a graph can be applied to various fields of shortest distance costs in routes, computer games, robotics or navigation. This study implements the A star heuristic search algorithm in determining the shortest path using the Visual Basic programming language and MySQL database. A star heuristic algorithm is implemented to find the shortest path between two vertices in a two-way weighted graph. We use a heuristic method to estimate the remaining cost from the start node to the destination point. This condition gives the A star algorithm the opportunity to choose the next closest node and optimize the search for the shortest route. Various experiments have been carried out with various graph conditions and with different complicated graphs to test the A star algorithm. The experimental results show that this algorithm succeeds in finding the shortest path efficiently in various situations. Our findings confirm that the A star algorithm has great potential in solving the shortest path search problem with an advantage in combining various information and remaining cost estimates, thereby minimizing the number of explored nodes and producing the shortest path efficiently.

Keywords: Shortest path, A star heuristic search algorithm, Two-way weighted graph, Heuristic method, Search efficiency

This is an open access article under the CC BY-NC license.

Corresponding Author:
Aswandi,
Department of Network Computer Engineering Technology,
Politeknik Negeri Lhokseumawe,
280 Lhokseumawe Road Blang Mangat, Banda Aceh, Indonesia.
Email: aswandi@pnl.ac.id

1. INTRODUCTION

The solution to finding the shortest route or the shortest path is very interesting to study, many problems can be solved in determining the shortest path. Determination of the shortest path can also be implemented in various trending applications such as vehicle navigation, route planning, robotics, and computer games [1][2]. Implementing the shortest path can save time, energy, and resources. A Star Algorithm is one method of approach in finding solutions to the shortest path finding problem. This algorithm is a combination of Dijkstra’s algorithm and Greedy Best-First Search. This algorithm can provide cost information and estimated remaining costs in determining the shortest path more...
efficiently. Although the A star algorithm has been carried out significantly in the literature, the implementation and performance analysis of this algorithm in the context of determining the shortest path is still a relevant research focus [3][4][5][6][7][8].

Several previous studies have identified that the performance of the A star algorithm depends on various factors, including choosing the right heuristic, graph structure, and the complexity of the area explored [9][10][11][12][13]. Therefore, there is a need to better understand the practical implementation of the A star heuristic algorithm in determining the shortest path in real-world scenarios. By combining information from existing literature and empirical experimental results, this study aims to provide deeper insight into the ability and potential of the A star heuristic algorithm in solving the shortest path problem [14][15][16][17][18]. This research will implement the A star heuristic algorithm into applications built with the Visual Basic programming language. This study also describes in detail the practicalities of this algorithm in determining the shortest path, this research is expected to provide a better guide to understanding how the A star algorithm can be applied effectively in various real-world application contexts.

2. RESEARCH METHOD

In many cases it would be much better if a function is defined to be a combination or sum of two or more components, namely $g(n)$ and $h(n)$. The $g(n)$ function is a measure of the costs incurred from an initial state to node $(n)$. The result of $g(n)$ is not the result of estimation but the sum of the costs of implementing each rule along the best path determined by the heuristic function to a node. For the function of $h(n)$ is a measure of the additional costs that must be incurred from node $(n)$ to reach the destination. Note that the function of $g(n)$ cannot be negative because if it is negative then the cycle-reversing path on the graph will look better the longer the number of paths. The mathematical function can be written as follows:

$$f(n) = g(n) + h'(n)$$

With $f(n)$ is function of evaluation, $g(n)$ is costs that have been issued from the initial state to node $(n)$, $h'(n)$ is estimation of costs incurred from state $n$ or node $(n)$ to the destination.

If $h = h'$, then the search process has reached its goal. If $g = h' = 0$ then $f'$ is random, which means the system cannot be controlled. If $g = k$, $k$ is a constant and usually has a value of 1, $h' = 0$, which means that the system uses the best first search technique.

![Figure 1. Weighted Graph](image)

Based on Figure 1, given a case, if a driver will go around from Pontianak to Melawi to return to Pontianak, in search of the shortest route. calculate the value of $h'(n)$ using the two-point formula:

$$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$
Calculation:
A ke B = (2,14), (14,19) = 13
A ke C = (2,14), (38,15) = 36,01
A ke D = (2,14), (32,4) = 31,62
A ke E = (2,14), (13,5) = 14,21
B ke A = (14,19), (2,14) = 13
B ke C = (14,19), (38,15) = 24,33
B ke D = (14,19), (32,4) = 23,43
B ke E = (14,19), (13,5) = 14,04
C ke A = (38,15), (1,14) = 36,01
C ke B = (38,15), (14,19) = 24,33
C ke D = (38,15), (32,4) = 12,43
C ke E = (38,15), (13,5) = 26,93
D ke A = (32,4), (1,14) = 31,62
D ke B = (32,4), (14,19) = 23,43
D ke C = (32,4), (38,15) = 12,53
D ke E = (32,4), (13,5) = 19,03
E ke A = (13,5), (1,14) = 14,21
E ke B = (13,5), (14,19) = 14,04
E ke C = (13,5), (38,15) = 26,93
E ke D = (13,5), (32,4) = 19,03

After that look for the value of $f(n)$, $g(n)$ is obtained from measuring the distance between 1 point to another point and then finding the value of $f(n)$ with the formula $f(n)=h'(n)+g(n)$:
A ke B = 18 + 18 = 31
A ke C = 36,01 + 37 = 73,01
A ke D = 31,62 + 40 = 71,62
A ke E = 14,21 + 20 = 34,21
B ke A = 13 + 18 = 31
B ke C = 24,33 + 29 = 53,33
B ke D = 23,43 + 34 = 57,43
B ke E = 14,04 + 16 = 30,04
C ke A = 36,01 + 37 = 73,01
C ke B = 24,33 + 29 = 53,33
C ke D = 12,43 + 17 = 29,43
C ke E = 26,93 + 35 = 61,93
D ke A = 31,62 + 40 = 71,62  
D ke B = 23,43 + 34 = 57,43  
D ke C = 2,53 + 17 = 29,43  
D ke E = 19,03 + 20 = 39,03  
E ke A = 14,21 + 20 = 34,21  
E ke B = 14,04 + 16 = 30,04  
E ke C = 26,93 + 35 = 61,93  
E ke D = 19,03 + 20 = 39,03

Figure 2. Calculation Results of Algorithm A*

The vector data model is represented by symbols or what is hereinafter known in Geographic Information Systems as features, such as point features, line features, area features (surface). The purpose of designing a database server in this application is to record the distance from the starting point to the destination point. Besides that, the database is placed on notepad so that the system can be processed quickly.

Table 1. The performance of

<table>
<thead>
<tr>
<th>No.</th>
<th>Titik</th>
<th>X</th>
<th>Y</th>
<th>Cabang</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A</td>
<td>20</td>
<td>20</td>
<td>B H</td>
</tr>
<tr>
<td>2</td>
<td>B</td>
<td>40</td>
<td>20</td>
<td>A C D</td>
</tr>
<tr>
<td>3</td>
<td>C</td>
<td>50</td>
<td>58</td>
<td>B D E H X</td>
</tr>
<tr>
<td>4</td>
<td>D</td>
<td>63</td>
<td>20</td>
<td>B C F</td>
</tr>
<tr>
<td>5</td>
<td>E</td>
<td>77</td>
<td>53</td>
<td>C F G J X</td>
</tr>
<tr>
<td>6</td>
<td>F</td>
<td>83</td>
<td>20</td>
<td>D E G V</td>
</tr>
<tr>
<td>7</td>
<td>G</td>
<td>96</td>
<td>45</td>
<td>E F J V</td>
</tr>
<tr>
<td>8</td>
<td>H</td>
<td>20</td>
<td>86</td>
<td>A C L X</td>
</tr>
<tr>
<td>9</td>
<td>I</td>
<td>107</td>
<td>98</td>
<td>J K L N X Z</td>
</tr>
<tr>
<td>10</td>
<td>J</td>
<td>100</td>
<td>76</td>
<td>E G I X</td>
</tr>
</tbody>
</table>
3. RESULTS AND DISCUSSIONS

Below is a picture of an application that has been built with the Visual Basic programming language:

![Algorithm Main Menu Display](image)

The starting point can be determined by filling in letters in the Initial textbox. Figure 4 is the display result for determining the starting point.
The coordinates used in this study are limited to 26 coordinates according to the number of letters of the alphabet. This is to limit so that the calculation is not too much. Coordinates are made starting from the letters A to Z. Each coordinate functions as a 2-way weighted graph where A has access to B and vice versa B has access to A.

The end point can be determined by filling in letters in the End textbox. Figure 5 is the display result of determining the End Point.

In this section, the results of calculating the closest route to the starting point A and ending point S will be displayed. It can be seen that the weight achieved is 275 with the resulting route being A-B-C-X-I-N-S. Figure 6 is the display result of the closest route search.
4. CONCLUSION

Research related to the shortest route A* algorithm has several conclusions, namely the A* algorithm can determine the shortest route from the starting point to the ending point and vice versa from the ending point to the starting point. The application that was built was successful in determining the weight used from the distance of two points. The A* algorithm has a heuristic function in helping to determine the distance from neighboring nodes to endpoints.

ACKNOWLEDGEMENTS

Thanks are extended to the Universitas Pembangunan Panca Budi in particular the UNPAB Research Institute and Study Center and the UNPAB National and International Journal Affairs Institute for recording this research and publication.

REFERENCES


