Visualization of A-Star Pathfinding Algorithm

Mian Inshaullah Zia, 18PWCSE1721
Department of Computer Systems Engineering
University of Engineering and Technology
Peshawar, KPK, Pakistan.
18pwcse1721@uetpeshawar.edu.pk

Abstract

In 1959, Dijkstra proposed an algorithm that determines the shortest path between two nodes in a graph. The algorithm has redefined people's lives through Maps, road networks, and currently neural networks. While the manipulation of the shortest path between various locations appears to be a rising issue in road networks, an extensive range of applications and difficulties were overcome by developing different pathfinding algorithms such as Dijkstra's Algorithm. The classic Dijkstra's Algorithm was designed to solve the single-source shortest path problem for a static graph. It works starting from the origin node and calculates the shortest path on the whole network.[1]

In this paper, we will optimize the Dijkstra Algorithm by adding a heuristic function and turning the algorithm into an informed search algorithm through weighted graphs, called the A-star Algorithm.

By adding a heuristic function, the worst-case performance of the Dijkstra's algorithm evolves from $\boldsymbol{\Theta}(|\boldsymbol{V}|^2)$ ***to*** $\boldsymbol{\Theta}(|\boldsymbol{E}|)$**.** [1][2]

Introduction

The A-star algorithm is an informed search algorithm based on weighted graphs. The A* algorithm utilizes features of uniform-cost search, and pure heuristic search to compute an optimal solution efficiently. A* algorithm is also known as a 'best-first' algorithm in which the cost associated with a node is $\boldsymbol{f(n)} = \boldsymbol{g(n)} + \boldsymbol{h(n)}$, where g(n) is the cost of the path from origin to goal. Thus, f(n) estimates the ***lowest total cost*** of any solution path going through node n. [3]

At each point, a node with the lowest f(n) value is chosen for expansion. Connections among nodes of equal f(n) values are further dissected in favour of lower h(n) values. Our algorithm terminates when a goal is chosen for expansion.

Technique Used

- We have proposed a system that creates a console with a grid of nodes, each interconnected with one another vertically, horizontally, and diagonally.
- With the help of a game engine, we were able to draw nodes, the connection of the nodes, the origin, and the goal.
- The nodes highlight if they have been searched and show the shortest possible path from the origin node to the final node.

Program Function

- We have created a structure called ***sNode*** that contains the following elements.
    - Nodes in the X and Y-axis
    - Parent Node
    - Check if we visited the node.
    - Check if the node is an obstacle.
    - Find the absolute shortest path.
    - Find a comparatively short path.
- We further placed 16x16 nodes.
- We connected each node vertically, diagonally, and horizontally by creating vertices in a graph.
- With the help of the structure as mentioned above, we were able to utilize the shortest path, using ***Euclidean Distance Formula*** to utilize an admissible heuristic function in order to find the shortest path.
- With the help of the Euclidean Distance Formula, we were able to create newer vectors further and their functions to create the optimal path to the end-point.
- To visualize the algorithm, we filled each node with pixels. We further drew connections of the pixels, as mentioned earlier with the help of the Game Engine header file.
- With the help of the Game engine, we were able to instruct each node with unique colours to denote the following.
    - The optimal route (Magenta)
    - The searched nodes (Cyan)
    - The obstacles. (Red)
    - The Origin (Blue)
    - The Destination (Green)

Complexity Analysis

The time complexity of the A-star algorithm depends hugely on the heuristic function. The A-star algorithm is an extension of the Dijkstra Algorithm with an admissible heuristic function that utilizes the Euclidean Distance Formula and Triangle inequalities to implement an 'informed search.' [4]

Interestingly, the best case for Dijkstra's Algorithm is $\boldsymbol{\Theta(|E| + |V|log|V|)}$ in contrast to A-star's average case $\boldsymbol{\Theta(|E|)}$ and best case of $\boldsymbol{\Theta(1)}$. However, the A-star algorithm has a polynomial worst time complexity of $\boldsymbol{\Theta(b^d)}$. [1][4][5]

Conclusion

The A* algorithm is an innovative algorithm that is widely used in maps, road networks, games, and artificial intelligence. It may be one of the best and popular pathfinding and graph traversal algorithms.

Today, a plethora of games and web-based maps use this algorithm to optimize pathfinding by finding the shortest path efficiently.

In terms of finding the shortest path in the least amount of time, the A* algorithm has been deemed more efficient in contrast to Dijkstra's algorithm. Consequently, the A* algorithm may lack performance in larger grids with space complexity of $\boldsymbol{\Theta(b^d)}$.

However, the A* algorithm is undoubtedly the most suitable and fastest algorithm to find the shortest path in a limited grid. The A* algorithm takes the edge from various other algorithms because of its 'informed search technique.'

References

[1] Wikipedia (2020), Dijkstra's Algorithm, available at https://en.wikipedia.org/wiki/Dijkstra%27s_algorithm

[2] Wikipedia (2020), A-Star Algorithm available at https://en.wikipedia.org/wiki/A*_search_algorithm

[3] IJSR(2012), Memory Based A-Star Algorithm for Path Planning of a Mobile Robot available at https://www.ijsr.net/archive/v3i6/MDIwMTQ0NTE=.pdf

[4] Stack overflow (2017), "What is the time complexity of A-star algorithm," available at https://stackoverflow.com/questions/44849517/what-is-the-time-complexity-of-a-search

[5] Stack exchange (2017), "A* graph search time-complexity." Available at https://cs.stackexchange.com/questions/56176/a-graph-search-time-complexity

[6]Mcgill (2013), "Comp-424, Lecture 3," available at https://www.cs.mcgill.ca/~dprecup/courses/AI/Lectures/ai-lecture03.pdf

[7] Stack overflow (2010), "What is the complexity of A* exponential in memory?" available at https://stackoverflow.com/questions/1715401/why-is-the-complexity-of-a-exponential-in-memory

[8] GeekforGeeks (2018), "A* Search Algorithm," available at https://www.geeksforgeeks.org/a-search-algorithm/

[9] Levitin, Anany. (2000). Design and analysis of algorithms reconsidered. ACM Sigcse Bulletin. 32. 16-20. 10.1145/330908.331802

[10] Duchoň, František & Babinec, Andrej & Kajan, Martin & Beňo, Peter & Florek, Martin & Fico, Tomáš & Jurišica, Ladislav. (2014). Path Planning with Modified a Star Algorithm for a Mobile Robot. Procedia Engineering. 96. 10.1016/j.proeng.2014.12.098.

[11] Sneha Sawlani (2017), "Explaining the performance of Bidirectional Dijkstra and A* on Road Networks available at https://digitalcommons.du.edu/cgi/viewcontent.cgi?article=2303&context=etd

[12] One Lone Coder (2019), OlcPixelGameEngine available at https://github.com/OneLoneCoder/olcPixelGameEngine/wiki

[13] Duchoň, František & Babinec, Andrej & Kajan, Martin & Beňo, Peter & Florek, Martin & Fico, Tomáš & Jurišica, Ladislav. (2014). Path Planning with Modified a Star Algorithm for a Mobile Robot. Procedia Engineering. 96. 10.1016/j.proeng.2014.12.098.