Literature Review - Shortest Path Finding on the UMN Campus

Junyuan Wang (wang9747), Yicheng Zhai (zhai0041)

March 31, 2023

1 Introduction

Since this project aims to help users find the shortest path within the UMN Twin Cities campus by utilizing OpenStreetMap (OSM), NetworkX, and Python, along with algorithms including A* and Dijkstra's algorithms, this literature review is separated into three parts. First, it will introduce these tools and algorithms by providing an overview of their key features and capabilities and explore how these methods have been employed in similar studies when solving real-world path-finding problems. Second, it will examine other studies using similar algorithms to ours. Finally, it will summarize the experimental approaches and conclusions derived from these studies.

2 Tools

2.1 OSM

Given that this project relies on map data from OpenStreetMap (OSM), we need to assess the trustworthiness of the data. Grinberger's study [4] offers an academic perspective on evaluating the reliability of OSM data, focusing on three perspectives of OSM data: Application of OSM data, OSM data quality, and Dynamics in OSM. The study highlights the impressive scale of the project, with almost 7.5 billion data nodes contributed by 1.8 million users as of March 2022. It is also common to see scholars incorporate OSM data by contributing mapping resources into more advanced models, especially machine learning. The analysis of OSM data quality provides evidence of a fair accuracy of OSM data on tourism. However, since the OSM data is ”crowd-sourced”, i.e, the communities might have different ways of communication and evaluations of science, the article points out that it requires more understanding to utilize OSM data. In light of these findings, we are confident in the reliability of OSM data, though we might spend more time on data processing.

2.2 NetworkX, Python (PyCharm)

To improve the efficiency of data manipulation and the quality of data visualization, the project will utilize NetworkX as an important tool while implementing our algorithms in Python programming language which is easy to code and document. In Hagberg's article [5], it introduces how flexible NetworkX is when representing different types of graphs, edges, and nodes that are fundamental for networks like ours. Beyond this, NetworkX also provides many functions to calculate the statistics of the network like connected nodes, coefficient, etc. By utilizing the ”dictionary of dictionaries” as its basic data structure for graphs, NetworkX facilitates finding the shortest path in a weighted graph and simplifies the algorithms required for this project. Additionally, NetworkX supports the integration of external tools, allowing us to incorporate more tools like Graphviz and Matplotlib into NetworkX to generate more detailed visualizations for the project.

3 Algorithms

3.1 A*

As A* and Dijkstra are two common algorithms to discuss and be implemented by scholars, Aziz [3] conducted an experiment using a 25x25 grid-based environment to compare the performances of A*, Ant Colony Optimization (ACO), and Dijkstra's Algorithms. Despite the limitations of the experiment, such as the distancing problem and different characteristics of the three algorithms, A* is shown to win the game.

Zeng [10] performed an empirical study on road network data in California, demonstrating that “on real road networks, A* outperforms the best implementations of the Dijkstra algorithm by a significant margin.” The superior performance of A* was achieved through the use of spatial coordinates to refine the search for the shortest path. The authors note that the potential of the A* algorithm can be further enhanced by improving its heuristic function. By generating more accurate estimated completion costs, the number of visited nodes and algorithm run-time can be reduced. This observation highlights the flexibility of the A* algorithm and motivates us to improve the A* algorithm for achieving better performance in this project.

3.2 Dijkstra

Analogous to our project, the self-driving automobile proposed in He's application [6] of Dijkstra's algorithm in finding the shortest path shares the same objective, but with the added dimension of determining the shortest distance between two regions. The authors have directed their focus towards the utilization of Dijkstra's algorithm and posited that ”Dijkstra algorithm is faster than other algorithms for it can calculate the shortest length to every point”. The algorithm employs a pyramid tree structure to pinpoint all vertex nodes that fall within the range of interest, as well as a heap to preserve distances and extract nodes with the least distance. These salient features impel us to prioritize testing and deploying Dijkstra's algorithm in our research project.

Fitro's experiment [2] which incorporates Geographic Information System (GIS), Google Map API, and Dijkstra's algorithm to find the shortest path at Taman Subdistrict, Indonesia, mentions

that Dijkstra's algorithm spends a large memory space. In the experiment, the author introduces a node combination technique to reduce memory usage, i.e., ”merging two nodes that have the closest distance”, and the optimal paths are shown in the result.
Wayahdi [8] presents a further comparative analysis of the efficiency of Greedy, A*, and Dijkstra, for determining the shortest path in a given graph. The Greedy algorithm, while speedy, may not necessarily guarantee a solution. On the other hand, the A* algorithms are relatively more efficient, but their performance is contingent upon complex data. In contrast, the Dijkstra algorithm invariably produces the optimal outcome, making it the ideal choice for shortest path determination. However, it may take longer to solve the problem than the other two algorithms. Despite this drawback, the author unequivocally recommends employing the Dijkstra algorithm for solving problems involving complex searches for determining the shortest path.

3.3 Bellman-Ford

The Bellman-Ford algorithm is also widely utilized for solving the problem of identifying the shortest path between two points. In AbuSalim's comparative analysis [1], the article compares the Dijkstra and Bellman-Ford algorithms for optimizing the shortest path. The author of this analysis notes that both algorithms are highly effective for determining the single-source shortest path. Bellman-Ford, as a dynamic algorithm, can efficiently compute the shortest path even in the presence of negative edge weights. Additionally, it performs better on smaller graphs. On the other hand, Dijkstra's algorithm is better suited for larger graphs and positive edge weights, with a time complexity of $O(|E|+|V|log|V|)$, while Bellman-Ford's algorithm has a complexity of $O(|V||E|)$. In general, Dijkstra's algorithm is more suitable for real-time applications than Bellman-Ford's one. We will consider Dijkstra's algorithm the preferred option in our project. Nonetheless, we will also conduct experiments to evaluate the efficacy of the Bellman-Ford algorithm, which is still widely recognized as a highly effective algorithm for solving shortest-path problems.

3.4 Conclusion on Algorithms

Academically, we contend that the Greedy algorithm is an inadequate algorithmic choice due to its incapacity to yield a solution. Based on the comparative analysis presented in the extant literature, we contend that A* and Dijkstra algorithms are the two most highly regarded algorithms for implementation in this project. Our team is eagerly anticipating the practical application of both algorithms in our project to evaluate their effectiveness. Other algorithms like Bidirectional Search, Bellman-Ford, and Breadth-First Search might be implemented based on Russell's AI book [7] but will not be discussed here.

4 Conclusion

Upon reviewing studies on the reliability of OSM data, examining the capabilities of NetworkX and its powerful visualization tools, and analyzing different combinations of algorithms for solving the path-finding problem, we have ensured that our project is feasible and ”playable”. Moreover, we have found an exemplary application by Yan and Wong from The Hong Kong University of Science and Technology (HKUST), who developed the Path Advisor [9]. This tool provides 2D, 3D, and VR views of the campus map tool for the shortest path. If this project can be further developed, we hope that it can become the ”Path Advisor” for the University of Minnesota Twin Cities.

References

[1] Samah W.G. AbuSalim et al. “Comparative Analysis between Dijkstra and Bellman-Ford Algorithms in Shortest Path Optimization”. In: *IOP Conference Series: Materials Science and Engineering* 917.1 (Sept. 2020), p. 012077. ISSN: 1757-8981, 1757-899X. DOI: 10.1088/1757-899X/917/1/012077. URL: https://iopscience.iop.org/article/10.1088/1757-899X/917/1/012077.

[2] Arif Ilham Sulistio Purnomo Achmad Fitro Otong Saeful Bachri and Indra Frendianata. “Shortest Path Finding in Geographical Information Systems using Node Combination and Dijkstra Algorithm”. In: *International Journal of Mechanical Engineering and Technology* 9 (2 Feb. 2018), pp. 755–750. URL: https://iaeme.com/Home/article_id/IJMET_09_02_078.

[3] Anusha Aziz, Sheikh Tasfia, and M. Akhtaruzzaman. “A Comparative Analysis among Three Different Shortest Path-finding Algorithms”. In: *2022 3rd International Conference for Emerging Technology (INCET)*. 2022, pp. 1–4. DOI: 10.1109/INCET54531.2022.9824074.

[4] A. Yair Grinberger et al. “OSM Science—The Academic Study of the OpenStreetMap Project, Data, Contributors, Community, and Applications”. en. In: *ISPRS International Journal of Geo-Information* 11.4 (Mar. 2022), p. 230. ISSN: 2220-9964. DOI: 10.3390/ijgi11040230. URL: https://www.mdpi.com/2220-9964/11/4/230.

[5] Aric Hagberg, Pieter Swart, and Daniel S Chult. “Exploring network structure, dynamics, and function using networkx”. In: (Jan. 2008). URL: https://www.osti.gov/biblio/960616.

[6] Baoyi He. “Application of Dijkstra algorithm in finding the shortest path”. In: *Journal of Physics: Conference Series* 2181.1 (Jan. 2022), p. 012005. ISSN: 1742-6588, 1742-6596. DOI: 10.1088/1742-6596/2181/1/012005. URL: https://iopscience.iop.org/article/10.1088/1742-6596/2181/1/012005.

[7] Stuart J. Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Fourth edition. Pearson series in artificial intelligence. Pearson, 2021. ISBN: 9780134610993.

[8] M. Rhifky Wayahdi, Subhan Hafiz Nanda Ginting, and Dinur Syahputra. “Greedy, A-Star, and Dijkstra’s Algorithms in Finding Shortest Path”. In: *International Journal of Advances in Data and Information Systems* (2021).

[9] Yinzhao Yan and Raymond Chi-Wing Wong. “Path Advisor: A Multi-Functional Campus Map Tool for Shortest Path”. In: *Proc. VLDB Endow.* 14.12 (July 2021), pp. 2683–2686. ISSN: 2150-8097. DOI: 10.14778/3476311.3476319. URL: https://doi.org/10.14778/3476311.3476319.

[10] W. Zeng and R. L. Church. “Finding shortest paths on real road networks: the case for A*”. en. In: *International Journal of Geographical Information Science* 23.4 (Apr. 2009), pp. 531–543. ISSN: 1365-8816, 1362-3087. DOI: 10.1080/13658810801949850. URL: http://www.tandfonline.com/doi/abs/10.1080/13658810801949850.