Literature Review on Advanced Search and Sorting Algorithms and General Systems and Technologies to Assist Navigation of Motor Vehicles and Optimize Routing

Table of Contents

1: Introduction

Throughout the evolution of computer science, advanced algorithms have always been a sector that is constantly advancing. These advancements can be directly related to the demand for optimization of various computer-based problems involving routing of roads and computer networks. One such problem revolves around the routes taken by solid waste collection vehicles around suburban or urban towns. Waste collection is complex in that it can be optimized based on a large pool of factors, including distance travelled, time taken, cost, and traffic impact. This literature review explores advanced algorithms that can be applied to optimize waste collection based on these factors, thus developing the base for a project to create an algorithm to optimize garbage truck routes. The project more specifically revolves around route sorting and planning algorithms, with the forefront algorithm being the A-Star Search Algorithm, a type of 'heuristic search' algorithm (Edelkamp, Jabbar, & Lluch-Lafuente, 2005). The algorithm is designed to find the most efficient path between two points, or nodes. This can be expanded to many fields of work, including routing, wireless routing, road networks, mazes, and neural networks. All that is needed for A-Star to work is a collection of points, nodes, or junctions. This project chooses A-Star for its ability to be easily adapted and implemented to virtually any programming language or environment.

As garbage truck routing has not been a common application of routing algorithms, it is important to research the various options that are possibly applicable to this project. Thus, two algorithms will be examined in this paper: Dijkstra's and A-Star. Along with its flexibility, A-Star was chosen because of its ability to factor in more variables than the other algorithm, which in this project happened to be traffic. While it has been commonly agreed upon that Dijkstra's is more than sufficient for ‘one-to-one’ routing problems (one start point and one endpoint) across road networks, it struggled when dealing with more stops (Zeng & Church, 2009). Various applications of the algorithm will be discussed in this paper, with a special focus on projects involving road navigation, traffic, and routing. In addition, this literature review will briefly touch on applications of the algorithm that extend beyond the field of navigation, such as energy conservation, so as to explore the utility of the algorithm.

2.1: Dijkstra Description and Current Applications:

Figure 1 shows how a standard environment for Dijkstra’s or A-Star is set up. Referring to node “a” as the start node and node “z” as the end node, the algorithm can now execute. Between each pair of nodes is what is called a cost function, typically referred to as “g(n).” This is a flexible value as it can be calculated in many different ways. The simplest way to calculate this value is based on distance, classified as either Euclidean or Manhattan (Gunawan *et al*, 2018). Simply measuring straight line distance, also known as Euclidean Distance, is used most commonly. If the example shown was using that model, then the Euclidean distance between nodes “a” and “b” would be 9 units. Another method of distance calculation is known as Manhattan Distance. This method is more applicable to road systems that are curvy and windy, as it splits straight line distance into horizontal and vertical components, parallel and perpendicular to the horizon respectively. Additionally, this method works well when latitude and longitude of real-life points can be retrieved from a GPS system.

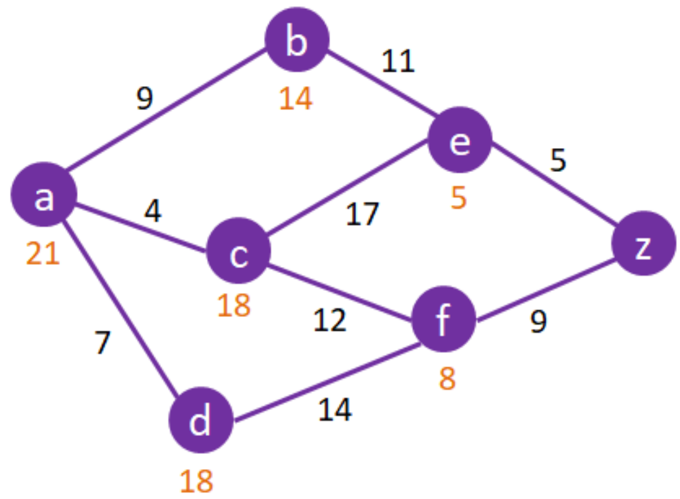

Figure 1: A Typical Graphical Environment Representation for Searching Algorithms

101Computing. (2018). *A-Star* [Diagram]. Website. https://www.101computing.net/star-search-algorithm/

Though distance is a good starting value for “g(n),” another feasible option (especially for roads), is known as a difficulty cost. While a freeway is physically larger than a backroad, it could be implied that the freeway is much easier and more efficient to travel. Thus, a longer freeway could be given a lower cost value, “g(n),” than a shorter backroad. In the example the path between nodes “c” and “f” seems to be much shorter than the path between nodes “a” and “d,” which could be explained with the former being a backroad and the latter being a freeway.

Once all the cost values are determined, Dijkstra’s algorithm creates what is known as a priority queue, which is able to determine the most effective method of travel. Priority queues appear in many algorithms in programming, and effectively take a list of elements and remove one at a time based on certain factors (Cormen, 1990). Starting from node “a,” the algorithm goes through every node that connects to “a” and takes into account each cost value. The node with the least cost, in this case “c,” is then moved to the top of the queue, and the process repeats with the nodes connecting to “c.” The algorithm keeps into account all visited and unvisited nodes, and a node is only considered visited if all of its connecting nodes have been considered as possibilities. Eventually, the shortest total cost can be determined with Dijkstra’s consistently.

2.2: Considering the Flaws of Dijkstra’s

Dijkstra’s algorithm suffers from one major flaw, which is its efficiency. It is very possible for the algorithm to check many inefficient paths and move further and further away from the goal node. The algorithm has no sense of direction, and only works to find the lowest costs, rather than the overall path. Thus, Dijkstra’s is effective for short routing problems, but falls significantly behind in larger projects, especially to A-Star (Thorup, 1999).

Currently, Dijkstra’s serves the purpose of being an easy pathfinding algorithm to be used by instructors to educate students. It demonstrates fundamental aspects of more complex algorithms, while also being a good building block to expand more algorithms off of.

3.1: A-Star Description

A-Star is a direct improvement on Dijkstra’s, as it addresses the aforementioned flaw that inhibits Dijkstra’s effectiveness. Rather than only basing its priority queue on one cost value, “g(n),” A-Star combines this cost function with what is known as a heuristic value, commonly referred to as “h(n).” A heuristic can best be referred to as an educated guess, since it is neither exact nor complexly calculated. Rather this heuristic serves the purpose of giving the algorithm a sense of direction and directing it towards the goal node.

Before the algorithm does any routing, each node is given this heuristic value, which can be calculated in many ways. If distance is being used for the cost function (Euclidean or Manhattan), the most effective method of estimating this heuristic is also by using distance. Other options become more obscure and complicated, but cost estimation is still possible if distance is not an option.

Once both functions are established, A-Star works exactly the same as Dijkstra's, with the latter's g(n) value being replaced by the former's composite g(n) + h(n) value. In Figure 1, the cost from "a" to "b" would be 9 + 14, or 23. This value more accurately represents the cost to move from node "a" to "b," as it applies to the network as a whole.

3.2: Wireless Networks

A practical application of A-Star is in wireless sensor networks. Currently, researchers are attempting to maximize the effectiveness of the algorithm in the multi-billion-dollar industry of wireless routing (Septiana, Soesanti, & Setiawan, 2016). Though these applications are complicated, they still revolve around the fundamental facets of A-Star, namely, g(n) and h(n), though each of these functions can be expanded to be more complex. Take, for instance, a formula derived for g(n) by a researcher optimizing a wireless network, as seen in Figure 2 (Septiana, Soesanti, & Setiawan, 2016):

$$g(n) = \frac{RSSI_{P \quad to \quad C}}{RSSI_{\min}}$$

$$RSSI = A - 10 n_p \log d \ , \ n_p = 2$$

$$RSSI = (P_t - PL) - 20 \log d$$

$$LQI = \frac{RSSI}{RSSI_{\min}}$$

Figure 2: A More Complex Calculation of g(n) for A-Star

Septiana, R., Soesanti, I., & Setiawan, N.A. (2016). "Evaluation function effectiveness in Wireless Sensor Network routing using A-star algorithm," *2016 4th International Conference on Cyber and IT Service Management*, Bandung, 2016, pp. 1-5.doi: 10.1109/CITSM.2016.7577519

So, while it is clear that A-Star is basic at its core, it can be expanded to fit many complex needs, such as optimizing wireless networks, or even optimizing delivery systems.

3.3: Optimizing Delivery Systems

Probably the most similar application of A-Star to garbage truck routing is the optimization of delivery systems. Consider the development of a mobile application that could be used to help route a delivery person. Taking into account the possibility of packages needing to be delivered to remote and rural areas, this application of A-Star would need to run as a standalone app, meaning no help from the internet or the cloud (Gunawan *et al*, 2018). Such an app would have to have a preloaded database of roads and their cost values. This is most easily accomplished by using geographical data, which runs on a satellite connection and therefore needs no internet connection. Researchers choose to label nodes, in this case houses, with latitude and longitude measurements, shown in Table 1, using simple coordinate geometry to calculate g(n) and h(n), as seen in Figure 3.

Table 1: Latitude and Longitude values for nodes, used to calculate distance.

Gunawan, D., Marzuki, I., & Candra, A. (2018). *J. Phys.: Conf. Ser.* 978 012122

Table 1. Example coordinates

Node	Street Name	Coordinates
A	Jl. Ringroad/Jl. Sunggal	3.5861425, 98.6325452
B	Jl. Sunggal/Jl. Setiabudi	3.5852696, 98.6368904
C	Jl. Sunggal/Jl. Sei Batang Hari	3.5844988, 98.6406202
D	Jl. Gatot Subroto/Jl. Sunggal	3.5894582, 98.6418203
E	Jl. Ringroad/Jl. Gatot Subroto	3.5902065, 98.6270023
F	Jl. Asrama/Jl. Gaperta	3.6042378, 98.6293834
G	Jl. Kapten Muslim/Jl. Gaperta	3.6026019, 98.6415906

The above coordinates were found, and A-Star was implemented fully on a low-powered and outdated smartphone. It follows that A-Star can be implemented with minimal resources, which is promising for cost evaluation models for standalone devices using the algorithm.

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

$$h(n) = \sqrt{(q1 - p1)^2 + (q2 - p2)^2}$$

Figure 3: An easier-to-follow example of the calculation done by A-Star for its heuristic value, h(n). This example uses coordinate geometry for both subfunctions.

Gunawan, D., Marzuki, I., & Candra, A. (2018). *J. Phys.: Conf. Ser.* 978 012122

3.4: Minimizing Energy Waste

In the field of wireless sensor navigation, the speeds of systems have reached an impressive level, though energy consumption seems to be trending upwards, leading to shortened network lifetimes (Ghaffari, 2014). To combat this, A-Star can be used once again, but with energy consumption as the determining factors of g(n) and h(n), rather than distance. A "hop" can be measured as one jump from a node to the next, which researchers try to minimize with algorithms, as these hops cost significant energy. A-Star can be used to get from a start node to an end node in the least possible number of "hops," so the heuristic, or h(n) value is ever-important to give the network a sense of direction towards the goal.

4.1: Current Attempts to Predict and Measure Traffic

In terms of garbage delivery, an often-overlooked factor that contributes to the flaws of the current systems are traffic impact. While there are robust systems in place for measuring traffic over long periods of time, methods of calculating traffic density over short time intervals are still in development (Ma, D., Sheng, B., Jin, S., Ma, X. & Gao, P, 2018). One advanced approach takes historical traffic data and clusters it based on patterns a smart algorithm observes. These patterns could be dependent on a number of factors from car density, time and date, location, etc. Then short-term measurements of any number of these factors can be taken and a smart predictor algorithm can match these factors to a pattern stored in its database. This predicts the traffic, which can be quantified as length of congestion per time unit.

4.2: Taking Measurements for Real-Time Calculations

Another aspect of navigation that has been researched within the last decade are methods to accurately measure real-time location and generate navigation internally. Essentially this research looks to cut down on the dependencies of the internet, while remaining smart and adaptable. One such method of achieving this is known as Extended Kalman Filtering (EKF) (Mungia, 2014). This method seeks to calculate real-time kinematic states of a vehicle without human assistance. This is an extension of a previous method known as Inertial Navigation System (INS) (Ohlmeyer, 2006).

These systems use gyroscopes to measure inertial vector quantities from the combination of a 3-axis gyroscope with a 3-axis accelerometer and 3-axis magnetometer (Mungia, 2014). Recent research has looked into how the cost of these sensors can be reduced. With the combination of these sensors, the EKF system can update its location and navigation in multiple ways, including an update based on GPS measurements or an update based on the velocity vectors.

5: Conclusion

This literature review has highlighted two major algorithms, Dijkstra's and A-Star, as well as their applications across many fields. The pros and cons of each algorithm has been considered, and it has been concluded that while Dijkstra's algorithm is proficient in simple routing tasks, A-Star's inclusion of a heuristic is paramount when applied to road networks. The heuristic assists greatly in road-based problems since it gives the algorithm a sense of direction when routing and avoids obscure and unrealistic paths. Practical applications of A-Star have been examined, including standalone delivery truck navigation and wireless sensor routing. The benefits of the algorithm in these various fields has been noted, and thus a project plan has been determined for the routing of garbage trucks. To begin, an environment for reading will be designed in either JAVA or Python, using the JAVA Graphics package or Python Graphics modules. Beginning with hard-coded road networks, A-Star will be tested with arbitrary start and end nodes to ensure the algorithm can work in the environment. As the algorithm becomes more complex, the benefits of A-Star will be seen as it will be able to handle the various roadblocks, busy roads, and dead-ends that the environment may contain. As a proof of concept, the initial algorithm will be adapted to route between five randomly selected pairs of nodes, whilst avoiding as many overlaps as possible. The principles of traffic and how it can be predicted and measured have been examined, opening up possible methods of traffic data retrieval to be taken in as inputs for the algorithm. Also, the possibility of a mobile application has been considered, and it has been found that an algorithm of A-Star's caliber can run on low-powered machines, a promising prospect for the future of this project.

6: References

Cormen, T. H. (2001). **Introduction to algorithms**. Cambridge, Mass: MIT Press.

Edelkamp, S., Jabbar, S., & Lluch-Lafuente, A. (2005). Cost-Algebraic Heuristic Search. *Proceedings of the National Conference on Artificial Intelligence. 3.* 1362-1367.

Ghaffari, A. (2014). An Energy Efficient Routing Protocol for Wireless Sensor Networks using A-star Algorithm. *Journal of Applied Research and Technology*, *12*(4). https://doi.org/10.1016/S1665-6423(14)70097-5

Gunawan, D., Marzuki, I., & Candra, A. (2018). *J. Phys.: Conf. Ser.* 978 012122

Ma, D., Sheng, B., Jin, S., Ma, X. & Gao, P. (2018). "Short-Term Traffic Flow Forecasting by Selecting Appropriate Predictions Based on Pattern Matching," in IEEE Access, vol. 6, pp. 75629-75638, 2018, doi: 10.1109/ACCESS.2018.2879055.

Mikkel Thorup. (1999). “Undirected single-source shortest paths with positive integer weights in linear time.” ACM 46, 3 (May 1999), 362–394. DOI:https://doi.org/10.1145/316542.316548

Munguía, R. (2014). A GPS-aided Inertial Navigation System in Direct Configuration. *Journal of Applied Research and Technology*, *12*(4). https://doi.org/10.1016/S1665-6423(14)70096-3

Ohlmeyer, E. J. (2006). "Analysis of an Ultra-Tightly Coupled GPS/INS System in Jamming," *2006 IEEE/ION Position, Location, And Navigation Symposium*, Coronado, CA, 2006, pp. 44-53. doi: 10.1109/PLANS.2006.1650586.

Septiana, R., Soesanti, I., & Setiawan, N.A. (2016). "Evaluation function effectiveness in Wireless Sensor Network routing using A-star algorithm," *2016 4th International Conference on Cyber and IT Service Management*, Bandung, 2016, pp. 1-5.doi: 10.1109/CITSM.2016.7577519

Zeng, W. & Church, R. (2009). Finding shortest paths on real road networks: The case for A. *International Journal of Geographical Information Science. 23.* 531-543. 10.1080/13658810801949850.

101Computing. (2018). *A-Star* [Diagram]. Website. https://www.101computing.net/star-search-algorithm/