United International Journal for Research & Technology

Volume 06, Issue 02, 2024 | Open Access | ISSN: 2582-6832

An Enhancement of A* Algorithm Applied in Automated Vehicle Parking

Mary Janelly S. Borbon[1], Rovia Zhen M. Indol[2], Raymund M. Dioses[3], and Dr. Khatalyn E. Mata[4]

[1,2]Student, [3]Faculty Adviser, [4]Subject Coordinator

[1,2,3,4]College of Information Systems and Technology Management, Pamantasan ng Lungsod ng Maynila (University of the City of Manila), Manila

***Abstract*—** The A* Algorithm is a path-finding algorithm that primarily uses weighted graphs and focuses on the heuristic values of nodes. However, while effective in generating a near-optimal path in a static environment, the traditional algorithm faces limitations in navigating dynamic environments, often resulting in collisions due to its inability to recognize dynamic and moving obstacles. This limitation makes it inefficient especially in complex environments with real-world scenarios. To address these limitations, an Enhanced A* Algorithm is proposed. This algorithm utilizes Navigation Mesh data structure to generate a more optimal route with local path planning and to dynamically adjust the parameters in two-dimensional non-grid environments. The performance of the algorithms was evaluated using 12 benchmarks, each corresponding to a distinct test case and levels of complexity. Then, in terms of dynamic obstacle avoidance, a comparison between the Enhanced A* Algorithm and the traditional algorithm was conducted. Statistical analyses were also performed to assess the consistency and validity of the findings. The results demonstrated that the Enhanced A* Algorithm successfully avoided all dynamic obstacles and moving objects encountered along the path in all distinct test cases. In contrast to the traditional algorithm, which achieved an average obstacle avoidance rate of 8.33%, the enhanced algorithm consistently demonstrated a 100% average obstacle avoidance rate. The enhanced algorithm outperformed the traditional A* algorithm in generating a path in a complex environment by exhibiting optimal dynamic obstacle recognition and avoidance. The Enhanced A* Algorithm is subsequently applied to autonomous vehicle parking, following standard parking restriction laws.

***Keywords*—** A* Algorithm, Automated Vehicle Parking, Dynamic Environment, Navigation Mesh, Path-finding Algorithm.

I. INTRODUCTION

The A* algorithm is a graph traversal path planning and heuristic search algorithm that can be used in finding optimal or near-optimal paths to take a system from a starting point to a final goal point within a specified environment (Karur et al., 2021). This algorithm uses weighted graphs in its implementation making it very efficient in taking a path with the least cost and in finding the best route in terms of distance and time traveled. It is primarily used in static environments where the conditions or obstacles do not change over time.

The A* algorithm works by evaluating the shortest path based on the information regarding the obstacles present in the environment (Karur et al., 2021). The algorithm primarily considers the combination of two factors when searching for optimal paths: first is the g(n), which is the cost of the path from the start node to node n, and the second is the heuristic estimate of the cost from node n to the destination node, or h(n). In each iteration of the main loop, the algorithm determines the node with the smallest evaluation function or f(n) to select the node that will be explored next (Baroud et al., 2019).

With the development of artificial intelligence, the A* algorithm has been used and tailored to different applications, such as in the gaming industry where it is used to optimize the pathfinding of a game map (Zhang et al., 2020). It is also widely used in urban intelligent transportation, and robot path planning (Dharmatti et al., 2021).

Multiple studies related in enhancing the A* algorithm relates to finding the shortest path in only a short time. In return, this disregards the generated paths of the algorithm which is unsmooth and jagged (R. Song, et. al, 2019). Another problem by the algorithm is when it is applied in land vehicles, it cannot perceive road turns. A study by S. Erke, et al, (2020), has a figure that shows the use of A* in land vehicles where it almost hits or it is too close to the obstacle which are unsatisfactory in real life applications. Furthermore, a problem with the A* algorithm is that it needs several key parameters

UIJRT ISSN: 2582-6832
United International Journal for Research & Technology
Volume 06, Issue 02, 2024 | Open Access | ISSN: 2582-6832

selected first before the algorithm is applied. These parameters are what affects the performance of the algorithm in terms of quality and time-consumed (S. Erke, et. al, 2020). There are still some defects on the A* algorithm even though it has been studied by multiple scholars proving its performance. Some of these defects are the small distances between the path and the obstacles, and the slow speed due to right-angle turns. These defects decrease the robustness of the planned path output by the algorithm (H. Wang, et. al, 2022).

Presently, some studies modified and improved the A* algorithm, but all are for particular problems and applications. A study by C. Ju, et. al (2020), improved the algorithm by the use of the shortest line segment between two points for the path planning. H. Wang, et. al (2022) introduced an improved A* algorithm called the EBS-A* algorithm where they used expansion distance, bidirectional search, and smoothing in the algorithm's path planning.

Nowadays, finding a suitable parking space has become a significant concern for many individuals, particularly in urban areas where the number of car users is constantly increasing. This also poses challenges in acquiring efficient parking slots as it will cause congestion, wasted time, and aggravation for drivers. That is why integrating an efficient algorithm into automated vehicle parking is very essential.

A. Statement of the Problem

The algorithm can be inefficient to use in dynamic environments as previously feasible node pairs might not always be valid, making it challenging to compute the shortest path.

In order to generate the path, the start, goal, and obstacles must be predetermined. Meaning, it must be placed already in the map or grid. Once computation starts, the user is unable to add or remove obstacles in the grid. This does not take into consideration the parking space closure and other moving obstacles that might hinder the travel to the end destination. According to Karur et. al, A* algorithm is not efficient in dynamic environments as it only generates a path based on the information regarding the obstacles that are already present in the environment. This means that once there are abrupt changes in the environment, the algorithm might generate an impractical or suboptimal path.

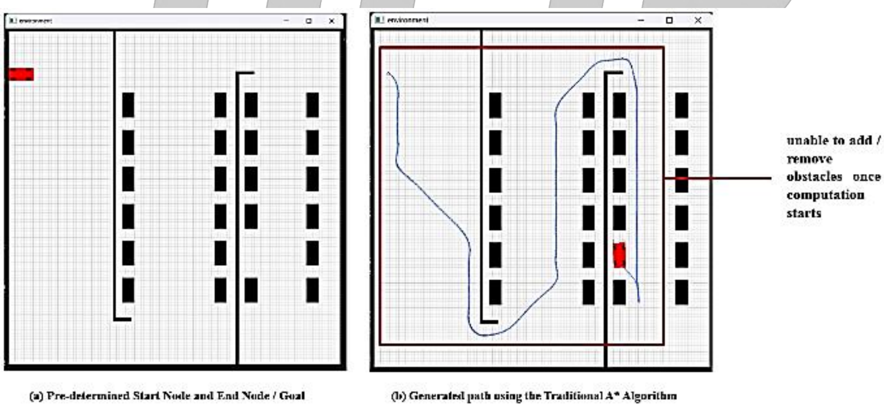

Figure 1: *Generated path being inefficient in dynamic environments*

B. Objective of the Study

The general objective of this study is to analyze and enhance the performance of A* algorithm to be able to maximize the limited capacity of a parking facility. Specifically, it aims to:

To be able to utilize the A* Algorithm efficiently in both static and dynamic environments by implementing artificial intelligence using Navigation Meshes to the A* Algorithm.

II. METHODOLOGY

A. Requirement Analysis

In enhancing the capabilities of A* algorithm for maximizing the limited capacity of a parking facility, three key requirements are addressed. The first requirement is to ensure the capability of the algorithm to have a continuous and smooth pathway by reducing the redundant traversal nodes. The aim is to apply a technique that can adjust the trajectory of the generated path and remove the unnecessary jagged lines to create a more optimal route, especially in complex environments. The second requirement is to add an expansion zone in the path planning to aid the problem of risking collision to the obstacles. This can be achieved by incorporating the rules of LTO regarding how far a car should be positioned from certain structures, such as walls or curb lines, and then adjusting the car's radius from the obstacles accordingly. The third requirement is to implement local re-planning, which allows the algorithm to operate efficiently in both static and dynamic environments. This involves the integration of the algorithm with Artificial Intelligence using Navigation Mesh (NavMesh), which provides a structured representation of walkable surfaces in an environment. This combined approach would help the traditional A* algorithm not only handles static environments, but also in dynamically changing environments to get the shortest path to the end node. To assess the enhanced algorithm's effectiveness, the researchers will limit their analysis to two domains which would include a two-dimensional parking lot layout with dynamic and moving obstacles.

B. Research Design

AI using Navigation Mesh (NavMesh)

The application of artificial intelligence employing Navigation Mesh (NavMesh) will be used to make the algorithm effective when operating in a dynamic environment. NavMesh is used to represent the walkable geometry within a virtual environment. By specifying the traversable regions, this technique helps the algorithm to efficiently determine the optimal path in a complex environment. Once the optimal path was determined, NavMesh also allows for smooth local movement by adjusting to obstacles or moving agents without recalculating the entire path.

C. Methods and Performance Metric

An input-process-output structure served as the framework in completing the objectives of this thesis. It has three inputs, which are the dynamic parking environment data, which includes real-time data about the obstacles, available parking spaces, and parked cars; the start node, which is the car's initial position; and the goal node, which is the predetermined destination of the car. To achieve the objectives of the study, three key steps will be employed such as implementing Navmesh, expanding obstacle radius to car, and implementing post-process path smoothing. The enhanced algorithm's output is the optical path generation and dynamic re-calculation of path anytime it comes across a dynamic or moving obstacle.

Figure 2: *Conceptual Framework for the Enhancement of A* Algorithm in Automated Vehicle Parking*

Obstacle Avoidance Rate

It evaluates the algorithm's efficiency, considering the number of dynamic obstacles and moving objects it avoided. The researchers will analyze and assess the differences between the obstacle avoidance rate of the traditional and enhanced algorithm by using the confusion matrix, which consists of four main components — true positive, true negative, false positive, and false negative. The value that would determine the obstacle avoidance rate would be the total number of true positives and true negatives divided by the total number of true positives, false negatives, true negatives, and false positives.

III. RESULTS AND DISCUSSION

This chapter presents the findings from the proposed enhanced algorithm. One of the objectives of this study is to make the algorithm generate a path efficiently in a dynamic environment. It will provide comparisons of the traditional A* algorithm and the enhanced A* algorithm applied in a two-dimensional parking lot to assess the enhanced algorithm's efficiency in path planning. For the experiment, both the traditional and enhanced algorithms underwent a series of simulations in a two-dimensional parking lot environment with a varying number of dynamic obstacles.

Volume 06, Issue 02, 2024 | Open Access | ISSN: 2582-6832

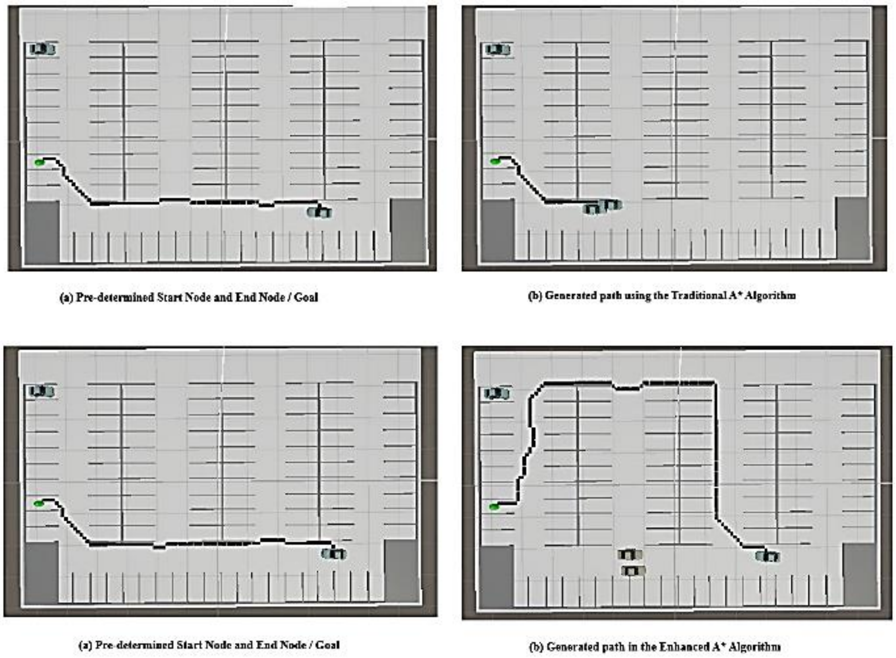

Figure 3: *Performance of the Traditional and Enhanced A* Algorithm Under Various Conditions (Dynamic Obstacles)*

Figure 3 shows the comparison of having dynamic obstacles in the traditional and enhanced algorithm. The proponents have pre-determined the start node and goal node, then position the obstacles in such a way as to block the initial path that is generated by the algorithm. The results have shown that the enhanced algorithm has performed better than the traditional algorithm in generating a path with abrupt dynamic obstacles. In the figure, it has shown that the enhanced algorithm has the capability to anticipate possible dynamic obstacles and immediately recalculate the path before the direct interference occurs between the car and obstacles. Unlike in the traditional algorithm, the generated path did not consider the obstacles' movements, leading to inefficient navigation. The enhanced algorithm also has shown that while avoiding the abrupt dynamic obstacles, it also considers the new optimal path that it generates to reach the goal node or destination.

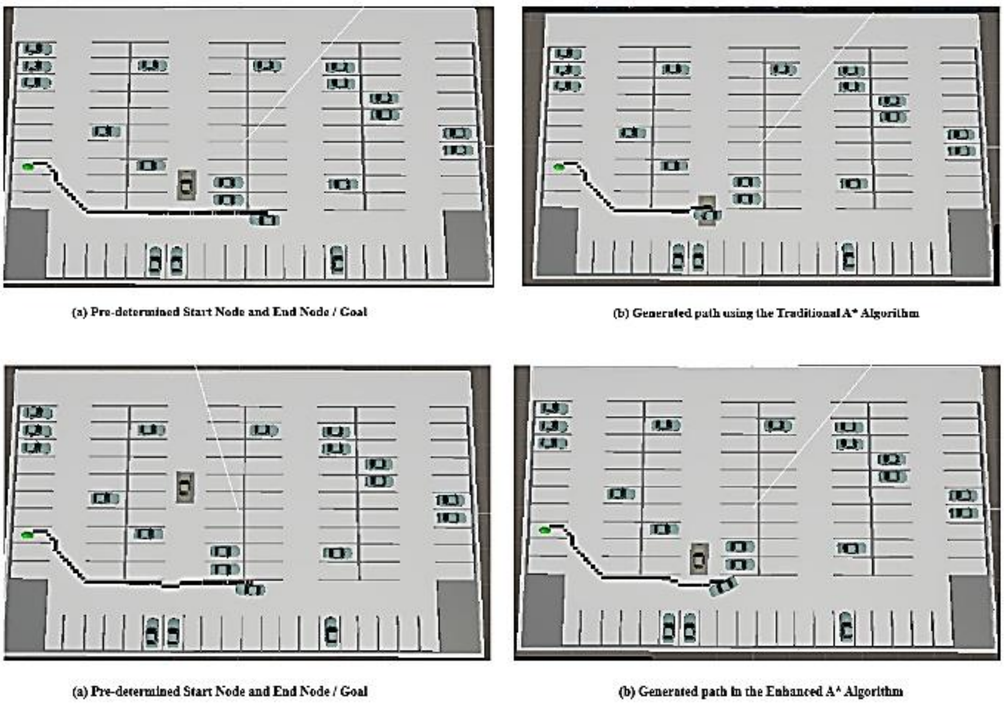

Figure 4: *Performance of the Traditional and Enhanced A* Algorithm Under Various Conditions (Moving Objects)*

In order to further assess the efficiency of the enhanced algorithm, the researchers also did a series of simulations incorporating moving objects in the running program and compared the performance of the traditional algorithm and the enhanced algorithm under these conditions.

In Figure 4, it shows the comparison of the performance of traditional and enhanced A* algorithm when it encounters a moving object. When a moving object was encountered, the traditional algorithm was unable to recalculate the path and instead the path still tends to pass through it rather than creating a more optimal path. This performance is inefficient and may result in vehicle collisions. Using the enhanced algorithm, on the other hand, it automatically generates a more optimal path when it comes into range with a moving object. Based on this comparison, the improved A* algorithm outperforms the traditional one.

Table 1: *Number of Dynamic Obstacles Avoided by the Car Using the Existing and Enhanced A* Algorithm*

Number of Dynamic Obstacles	Obstacle Avoidance Rate	
	Existing A* Algorithm	Enhanced A* Algorithm
1 dynamic obstacle	0.0%	100%
2 dynamic obstacle	0.0%	100%
3 dynamic obstacle	0.0%	100%
4 dynamic obstacle	0.0%	100%
5 dynamic obstacle	0.0%	100%
6 dynamic obstacle	0.0%	100%
1 dynamic obstacle with 1 moving objects	0.0%	100%
2 dynamic obstacle with 2 moving objects	0.0%	100%
3 dynamic obstacle with 3 moving objects	33.33%	100%
4 dynamic obstacle with 4 moving objects	50.0%	100%
Average Avoidance Rate:	8.333%	100%

True Positive (TP) = Obstacle Avoided Successfully
False Positive (FP) = Obstacle Avoided but no obstacle given
False Negative (FN) = Obstacle Not Avoided
True Negative (TN) = No Obstacle Given, No Avoidance
TP+TN//TP+FN+TN+FP = Avoidance Rate

The researchers ran two sets of simulations to determine how many dynamic obstacles the generated path avoided by using the traditional algorithm and enhanced algorithm and to compare the performance of both algorithms. The first set of simulations is in an environment with an increasing number of dynamic obstacles without moving objects, whereas an increasing number of dynamic obstacles with moving objects are included in the second set. The results are measured using the obstacle avoidance rate and have found out that while the traditional algorithm struggled with local path planning with an average obstacle avoidance rate of 8.333%, the enhanced A* algorithm consistently avoided all dynamic obstacles and moving objects that the path encountered with an average obstacle avoidance rate of 100%, as shown in the record presented in Table 1.

IV. CONCLUSION

This study presented the enhanced A* Algorithm for path planning which aims to improve the traditional algorithm's efficiency by avoiding dynamic obstacles and moving objects in a complex environment. Since the traditional A* Algorithm was utilized for shortest path determination, particularly on a static environment, we have presented a strategy to enable the enhanced algorithm to function in a dynamic environment for improving efficiency. The researchers integrated the Navigation Mesh data structure on the traditional A* algorithm and simulated a series of tests to evaluate its effectiveness in solving the problem of the existing algorithm. In 12 out of 12 benchmarks with various test case scenarios, the experiments have shown that the Enhanced A* Algorithm outperforms the traditional algorithm by avoiding all the dynamic obstacles as well as the moving objects that it encounters along the path. This makes the enhanced algorithm efficient to navigate in both static and dynamic environments with moving objects.

V. RECOMMENDATIONS

For future research papers and further exploration of the enhancements made in this study, the researchers advised to further optimize the algorithm by supporting multi-level parking structures. This includes assessing the performance of the algorithm in an environment with elevation variations, including ramps or inclined planes to ensure the path finding efficiency of the algorithm. Additionally, the study should be extended to address various road conditions, such as road humps, in order to further ensure vehicle performance and safety.

REFERENCES

[1] Al-Ansarry, S., Al-Darraji, S., Honi, D. (2023). An Efficient Path Planning in Uncertainty Environments using Dynamic Grid-Based and Potential Field Methods. Retrieved from https://www.researchgate.net/publication/372165738_An_Efficient_Path_Planning_in_Uncertainty_Environments_using_Dynamic_Grid-Based_and_Potential_Field_Methods

[2] Berglund, T., Brodnik, A., Jonsson, H., Staffanson, M., & Soderkvist, I. (2010). Planning Smooth and Obstacle-Avoiding B-Spline Paths for Autonomous Mining Vehicles. IEEE Transactions on Automation Science and Engineering, 7(1), 167–172. doi:10.1109/tase.2009.2015886

[3] Dundar, Y. (2021). Dynamic path finding method and obstacle avoidance for automated guided vehicle navigation in Industry 4.0. Retrieved from https://www.sciencedirect.com/science/article/pii/S1877050921019086

[4] Erke, S., Bin, D., Yiming, N., Qi, Z. (2020). An improved A-Star based path planning algorithm for autonomous land vehicles. Retrieved from https://www.researchgate.net/publication/346176563_An_improved_A-Star_based_path_planning_algorithm_for_autonomous_land_vehicles

[5] Fahleraz, F. (2018). A Comparison of BFS, Dijkstra's and A* Algorithm for Grid-Based Path-Finding in Mobile Robots. Retrieved from https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Makalah/Makalah-IF2211-2018-016.pdf

[6] Fu, B., Chen, L., Zhou, Y., Zheng, D., Wei, Z., Dai, J., Pan, H. (2018). An improved A* algorithm for the industrial robot path planning with high success rate and short length. Retrieved from https://www.sciencedirect.com/science/article/abs/pii/S0921889017306590

[7] Hassan, S., Islam, N., Fahim, A., Turja, T., Chowdhury, S. (2020). Automated Parking System using Graph Algorithm. Retrieved from https://www.researchgate.net/publication/340081381_Automated_Parking_System_using_Graph_Algorithm

[8] Iskandar, U., Diah, N., Ismail, M., Abdullah, A. (2022). Comparing the efficiency of Pathfinding Algorithms for NPCs in platform games. Retrieved from https://www.journalppw.com/index.php/jpsp/article/view/5109/3325

[9] Karur, K., Sharma, N., Dharmatti, C., & Siegel, J. E. (2021). A Survey of Path Planning Algorithms for Mobile Robots. Vehicles, 3(3), 448–468. doi:10.3390/vehicles3030027

[10] Mac, T. T., Copot, C., Tran, D. T., & De Keyser, R. (2016). Heuristic approaches in robot path planning: A survey. Robotics and Autonomous Systems, 86, 13–28. doi:10.1016/j.robot.2016.08.001

[11] Min, H., Xiong, X., Wang, P., & Yu, Y. (2020). Autonomous driving path planning algorithm based on improved A* algorithm in unstructured environment. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering, 095440702095974. doi:10.1177/0954407020959741

[12] Nadira, S., Omar, R., Hailma, C. (2016). Potential field methods and their inherent approaches for path planning. Retrieved from https://www.researchgate.net/publication/313389747_Potential_field_methods_and_their_inherent_approaches_for_path_planning

[13] Omonkhodion, G. (2023). A Comparative Study of A* and Greedy Best-First Search Algorithms in Solving 8-Puzzle Game. International Journal of Social Sciences and Scientific Studies, 3(1), 2321 - 2329. Retrieved from https://ijssass.com/index.php/ijssass/article/view/157

[14] P. Tozour, "Search space representations." In AI Game Programming Wisdom 2, edited by Steve Rabin. Charles River Media, 2004, pp. 85–102.

[15] Permana, S., Bintoro, K., Arifitama, B., Syahputra, A. (2018). Comparative Analysis of Pathfinding Algorithms A *, Dijkstra, and BFS on Maze Runner Game. Retrieved from https://www.researchgate.net/publication/325368698_Comparative_Analysis_of_Pathfinding_Algorithms_A_Dijkstra_and_BFS_on_Maze_Runner_Game

[16] Rachmawati, D. and Gustin, L. (2019). Analysis of Dijkstra's Algorithm and A* Algorithm in Shortest Path Problem. Retrieved from https://iopscience.iop.org/article/10.1088/1742-6596/1566/1/012061/pdf

[17] Ravankar, A., Ravankar, A., Kobayashi, Y., Hoshino, Y., & Peng, C.-C. (2018). Path Smoothing Techniques in Robot Navigation: State-of-the-Art, Current and Future Challenges. Sensors, 18(9), 3170. doi:10.3390/s18093170

[18] Singh, Y., Sharma, S., Sutton, R., Hatton, D., & Khan, A. (2018). A constrained A* approach towards optimal path planning for an unmanned surface vehicle in a maritime environment containing dynamic obstacles and ocean currents. Ocean Engineering, 169, 187–201. doi:10.1016/j.oceaneng.2018.

[19] Song, R., Liu, Y., Bucknall, R. (2019). Smoothed A* algorithm for practical unmanned surface vehicle path planning. Retrieved from https://www.sciencedirect.com/science/article/abs/pii/S0141118718302621

[20] Tang, G., Tang, C., Claramunt, C., Hu, X., Zhou, P. (2021). Geometric A-Star Algorithm: An Improved A-Star Algorithm for AGV Path Planning in a Port Environment. Retrieved from https://ieeexplore.ieee.org/document/9391698

[21] Wang, H. Lou, S., Jing, J., Wang, Y., Liu, W., Liu, T. (2022). The EBS-A* algorithm: An improved A* algorithm for path planning. Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8853577/

[22] Wang, J., Wu, X., & Xu, Z. (2008). Potential-based obstacle avoidance in formation control. Journal of Control Theory and Applications, 6(3), 311–316. doi:10.1007/s11768-008-6222-z

[23] Wang, X., Shi, H., Zhang, C. (2020). Path Planning for Intelligent Parking System Based on Improved Ant Colony Optimization. Retrieved from https://ieeexplore.ieee.org/document/9052744

[24] You, Z., Shen, K., Huang, T., Liu, Y., Zhang, X. (2023). Application of A* Algorithm Based on Extended Neighborhood Priority Search in Multi-Scenario Maps. Retrieved from https://www.mdpi.com/2079-9292/12/4/1004

[25] Zhang, C., Ao, L., Yang, J., Xie, W. (2020). An Improved A* Algorithm Applying to Path Planning of Games. Retrieved from https://www.researchgate.net/publication/345400800_An_Improved_A_Algorithm_Applying_to_Path_Planning_of_Games

[26] Zhang, J., & Sun, Y. (2022). A Real-Time multiplayer FPS game using 3D modeling and AI machine learning. https://aircconline.com/csit/papers/vol12/csit121310.pdf

[27] Zhang, J., Wu, J., Li, Y. (2021). Autonomous land vehicle path planning algorithm based on improved heuristic function of A-Star. Retrieved from https://journals.sagepub.com/doi/pdf/10.1177/17298814211042730

[28] Zhang, B., Li, Z., Ni, Y., Li, Y. (2022). Research on Path Planning and Tracking Control of Automatic Parking System. Retrieved from https://doi.org/10.3390/wevj13010014

[29] Zhu, M., Xiao, C., Gu, S., Du, Z., & Wen, Y. (n.d.). A circle grid-based approach for obstacle avoidance motion planning of unmanned surface vehicles. https://arxiv.org/pdf/2202.04494