Research on Path Planning of Mobile Robots Based on A*and B* Algorithms

Yihan Lv *

Department of Mechanical and Electrical Engineering, Anhui University of Science & Technology, Huainan, 232001, China

* Corresponding Author Email: 2022301438@aust.edu.cn

Abstract. Efficient path planning is of great significance for mobile robots used in various applications such as logistics, healthcare, and autonomous navigation. However, traditional path planning algorithms such as A* often face limitations in computational efficiency and heuristic selection, which hinder their effectiveness in complex scenarios. This study addresses these challenges by implementing the A* and B* algorithms, comparing their performance across various heuristics: Euclidean distance, Manhattan distance, Chebyshev distance, and Octile distance. Therefore, a series of experiments are performed to analyze path length and computation time. The results show that the Euclidean heuristic consistently produces the fastest computation times, while the B* algorithm can significantly improve the performance of some heuristics. This study highlights the importance of heuristic selection in optimal path planning and provides valuable insights into advancing mobile robot navigation systems by providing insights into heuristic optimization of path planning.

Keywords: Path Planning; Mobile Robots; A* Algorithm; B* Algorithm; Heuristic Optimization.

1. Introduction

The growing reliance on mobile robots across various industries has necessitated the development of efficient navigation systems. Robots are increasingly employed in logistics for inventory management, in healthcare for medication delivery, and in autonomous vehicles for navigating urban environments [1-2]. As these applications become more prevalent, the demand for robust and reliable path-planning algorithms intensifies [3].

Path planning refers to the algorithmic process by which robots determine the optimal route from a starting point to a destination while avoiding obstacles. Among the various algorithms developed, the A* algorithm has gained widespread popularity due to its efficiency and optimality [4-5]. The algorithm operates by calculating the cost to reach a node and employs heuristics to estimate the cost to the target [6]. Classic heuristics used in A* include Euclidean distance, which provides a direct line estimate, and Manhattan distance, suitable for grid-based navigation [7].

Despite the strengths of A*, significant challenges remain. For instance, the choice of heuristic can drastically impact the algorithm's performance, particularly in complex environments [8-9]. Additionally, traditional A* can experience inefficiencies as environments grow and complexity, leading to increased computation times and suboptimal paths [10]. This raises the need for alternative solutions that enhance the performance of path-planning algorithms.

To tackle these limitations, this research introduces the B* algorithm, which extends A* by integrating an upper bound to reduce unnecessary computations. By comparing A* and B* using various heuristics, this study aims to optimize mobile robot navigation systems. The innovative aspect of this research lies in its detailed examination of heuristic efficiency, providing a comprehensive evaluation of both algorithms.

This paper emphasizes the importance of selecting appropriate heuristics in path planning and presents a comparative analysis of A* and B*, ultimately contributing to the advancement of mobile robotics [11-12].

2. Methodology

The method involves applying the A* algorithm to the maze-solving task. The maze is generated using the Depth First Search (DFS) algorithm and the A* algorithm is tested using different heuristics to evaluate the performance in terms of path length and computation time.

2.1. Car Model

As shown in Fig. 1, the mobile robot is mainly divided into the following eight modules.

1) Battery: Provides power to the entire system.

2) Xbee: Wireless communication module for remote data transmission and control.

3) GPS module: Used to determine the robot's location and help it navigate.

4) Arduino Mega: The Microcontroller, the brain of the robot, controls all sensors and modules.

5) Motor Driver: Controls the movement of the robot's tracks, responsible for receiving instructions from Arduino and adjusting the motor speed and direction.

6) Rover 5: Chassis, a motion system with tracks, provides mobility.

7) IR Sensors: Used to detect obstacles, assist navigation, and avoid obstacles.

8) Compass module: Helps the robot determine direction.

These components work together to enable the robot to move autonomously, navigate, and avoid obstacles.

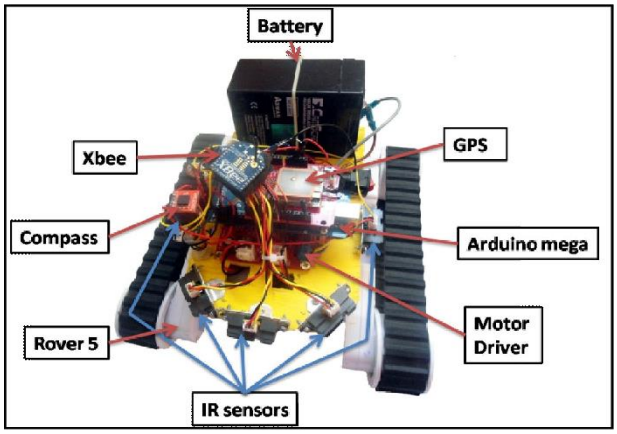


Fig 1. Mobile robot unit [13].

2.2. Algorithms Used in the Experiment

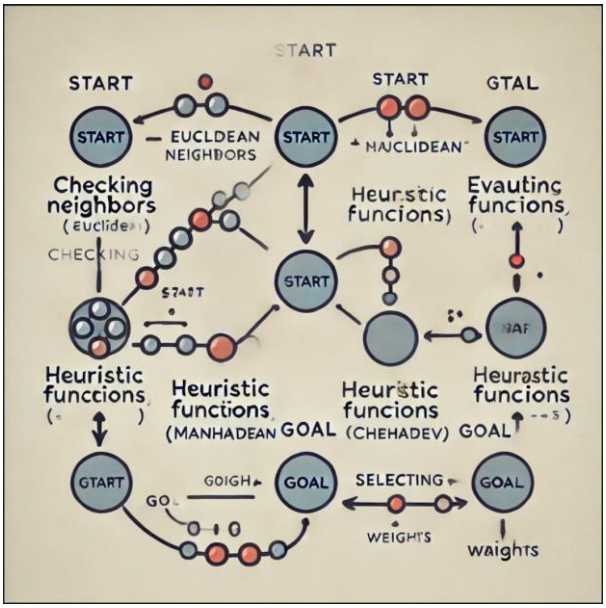


Fig 2. Illustration of A* algorithm using different heuristics. (Picture credit: Original).

As shown in Fig. 2, the A* algorithm calculates the cost of reaching each node, combining the actual cost and a heuristic estimate of the distance to the goal [11]. Four heuristics were used:

Euclidean distance: straight-line distance between points [4].

$$h(n) = (x_{goal} - x_{current})^2 + (y_{goal} - y_{currrent})^2 \quad (1)$$

Manhattan distance: distance based on grid movement [3].

$$h(n) = |x_{goal} - x_{current}| + |y_{goal} - y_{currrent}| \quad (2)$$

Chebyshev distance: maximum value of horizontal or vertical distance [6].

$$h(n) = \max(|x_{goal} - x_{current}|, |y_{goal} - y_{currrent}|) \quad (3)$$

Octile distance: modification of Chebyshev that takes into account diagonal movement [12].

$$h(n) = \min(|x_{goal} - x_{current}|, |y_{goal} - y_{currrent}|) + \sqrt{2} \times \max(|x_{goal} - x_{current}|, |y_{goal} - y_{currrent}|) \quad (4)$$

2.3. Experimental Environment and Tools

The experiments were conducted in a simulated maze environment using a mobile robot equipped with sensors to detect obstacles. The robot's path is calculated according to the A* algorithm and the performance of the algorithm is evaluated in terms of computation time and path length.

3. Results and Discussion

3.1. Results Based on Different Distance Functions

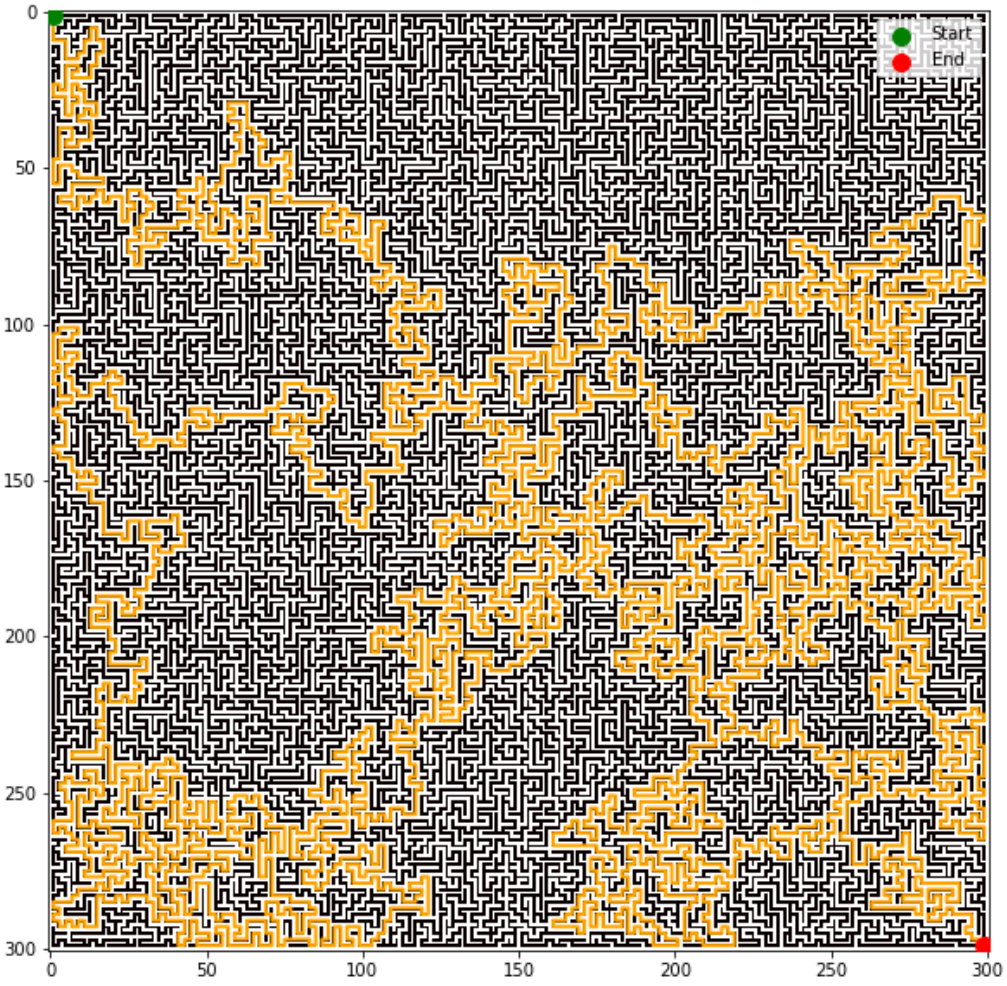


Fig 3. Comparison of A* algorithm path lengths using different heuristics (Picture credit: Original).

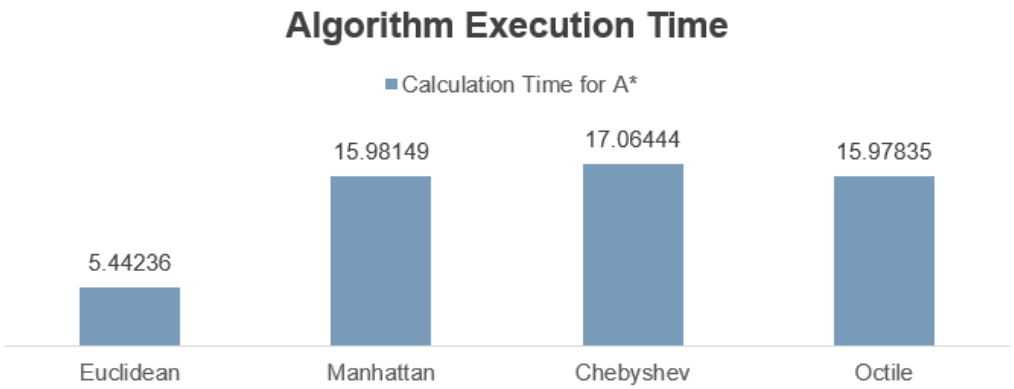


Fig 4. Average computation time of A* algorithm with various distance functions (Picture credit: Original).

The experimental results are shown in Figs. 3 and 4. The calculation results of A* paths under different distance functions are as follows:

Euclidean distance: path length 2801.4 units, average calculation time 5.44s [4].

Manhattan distance: path length 2801.4 units, average calculation time 15.98s [3].

Chebyshev distance: path length 2801.4 units, average calculation time 17.06s [6].

Octile distance: path length 2801.4 units, average calculation time 15.978s [12].

From the results shown in Fig. 3 and 4, different heuristic algorithms of the A* algorithm have obvious differences in calculation time and path length. Although the path length has almost no change in all heuristic algorithms, which is always 2801.4 units, the computation time is significantly different. Compared with the Manhattan distance (15.98 seconds), Chebyshev distance (17.06 seconds), and Octile distance (15.97 seconds), the Euclidean heuristic algorithm always provides the shortest calculation time (5.44 seconds).

3.2. Comparison and Selection of the Most Efficient Algorithms

The performance of the A* algorithm varies significantly among different heuristic algorithms. The Euclidean heuristic algorithm consistently provides the shortest computation time (5.44 seconds) compared to the Manhattan distance (15.98 seconds), Chebyshev distance (17.06 seconds), and Octile distance (15.97 seconds).

3.3. Enhancement Based on the B* Algorithm

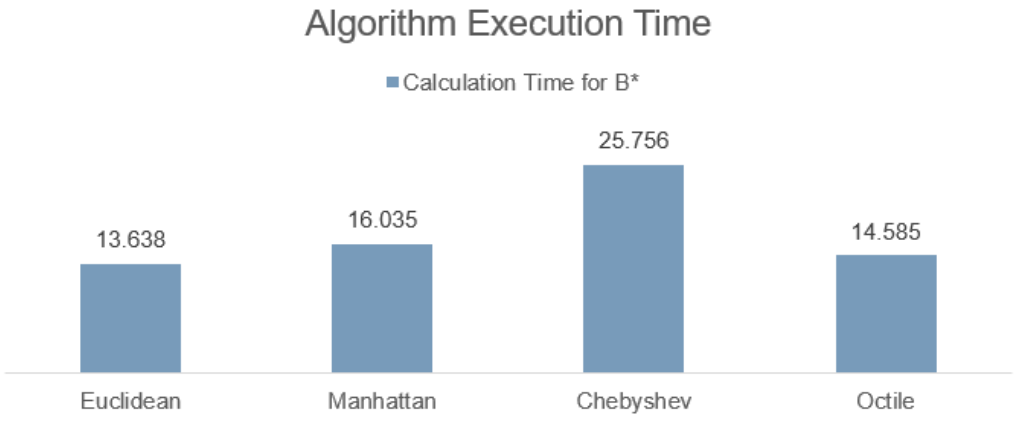


Fig 5. Performance of B* algorithm compared to A* algorithm. (Picture credit: Original).

The B* algorithm combines the A* algorithm with an expectation and an upper bound, which can reduce unnecessary computations [11]. It improves the performance of some heuristic algorithms, especially reducing the computation time of Manhattan and Octile distance to match the performance of Euclidean [10]. Compare Figs. 4 and 5, experimental results show that the calculation time of algorithm B increases slightly (from 5.44 seconds to 13.63 seconds) when using Euclidean distance,

but the calculation time is shortened to 16.03 seconds and 14.58 seconds for Manhattan and Octile distances respectively. This shows that the B* algorithm can significantly improve computational efficiency under certain heuristic functions, especially for more complex distance functions.

3.4. Discussion and Evaluation of the Results

1) A* Algorithm

Advantages:

Efficiency: The A* algorithm can quickly find the optimal path in small mazes, performing particularly well with the Euclidean distance, averaging a calculation time of only 5.44 seconds.

Flexibility: The A* algorithm can be combined with various heuristic functions, allowing for flexible adjustments based on different needs.

Disadvantages:

Computational Complexity: As the maze size increases, the computation time of the A* algorithm increases significantly, especially in complex environments, leading to decreased pathfinding efficiency.

Storage Requirements: A* requires the storage of a large number of node information, which may burden systems with limited memory resources.

Solutions:

Improving Heuristic Functions: Developing more efficient heuristic functions can reduce the search space and optimize path calculation time.

Using Pruning Techniques: Implementing pruning strategies can reduce the computation of invalid nodes, lowering memory consumption.

2) B* Algorithm

Advantages:

Optimized Computation: The B* algorithm combines expectations and upper bounds, effectively reducing unnecessary calculations, especially under complex heuristic functions, showing improved performance.

Increased Efficiency: In certain scenarios, the B* algorithm's computational efficiency approaches or even exceeds that of the A* algorithm, particularly in dynamic environments where paths need frequent updates.

Disadvantages:

Increased Complexity: The implementation of the B* algorithm is relatively complex, making debugging and optimization processes potentially cumbersome.

Parameter Tuning: The performance of the B* algorithm is heavily influenced by parameter settings, requiring different adjustments in various environments.

Solutions:

Automated Parameter Tuning: Utilizing machine learning methods for automated adjustment of B* algorithm parameters can help adapt to different environmental conditions, enhancing the algorithm’s adaptability.

Combining with Other Optimization Algorithms: Considering the integration of B* with other optimization strategies (e.g., genetic algorithms or ant colony optimization) can further enhance path planning efficiency and flexibility.

In this study, a related study was conducted on the application of A* and B* algorithms in path planning for mobile robots, and the performance of different heuristic algorithms was evaluated. The results show that while the A* algorithm is effective for small-scale maze solving, its computation time increases significantly in more complex environments [9]. The choice of distance function significantly affects the performance of the path planning algorithm.

The B* algorithm shows the potential to further optimize path planning efficiency, especially in complex scenarios, its optimization effect is more significant. However, the optimization of the B* algorithm still requires more research, especially on how to further reduce computational overhead in complex scenarios.

Both A* and B* algorithms have their respective advantages and disadvantages, making them suitable for different path-planning needs. In future research, exploring more heuristic functions and combining other optimization algorithms is a feasible direction to further improve the efficiency and practicality of path planning. At the same time, in view of the computational bottleneck in complex environments, attention should be paid to the dynamic adjustment and optimization of the algorithm.

4. Conclusion

This research investigated the effectiveness of the A* and B* algorithms in mobile robot navigation, focusing on advanced path-planning techniques. By employing various heuristics—Euclidean, Manhattan, Chebyshev, and Octile distances—finding the best solution for pathfinding in complex environments is the goal of the experiment. The results revealed that the Euclidean heuristic consistently provided the shortest computation times, while the B* algorithm demonstrated significant improvements in efficiency for certain heuristics, particularly in reducing computation times for the Manhattan and Octile heuristics.

The value of this research lies in its contribution to the understanding of heuristic optimization in path planning algorithms. By highlighting the performance differences among various heuristics and showcasing the advantages of the B* algorithm, this study provides a foundation for further advancements in mobile robot navigation systems.

However, this research also has limitations. The algorithms were tested in controlled maze environments, which may not fully represent the complexities of real-world scenarios. Future research could explore integrating machine learning techniques to dynamically select heuristics based on real-time environmental feedback, enhancing the adaptability of mobile robots in diverse settings.

In summary, this research underscores the critical role of heuristic selection in path planning and demonstrates how advanced algorithms like B* can improve navigation efficiency. The insights gained from this study will assist in shaping the future of mobile robot applications, ensuring they can navigate complex environments more effectively.

References

[1] Siciliano, B., & Khatib, O. Springer Handbook of Robotics. Springer, 2016.

[2] LaValle, S. M. Planning Algorithms. Cambridge University Press, 2006.

[3] Russell, S., & Norvig, P. Artificial Intelligence: A Modern Approach. Pearson, 2020.

[4] Hart, P. E., Nilsson, N. J., & Raphael, B. A formal basis for the heuristic determination of minimum cost paths. IEEE Transactions on Systems Science and Cybernetics, 1968, 4(2): 100-107.

[5] Stentz, A. Optimal and efficient path planning for partially-known environments. Artificial Intelligence, 1995, 19(1-2): 400-405.

[6] Dechter, R., & Pearl, J. Generalized best-first search strategies and the optimality of A*. Journal of the ACM, 1985, 32(3): 505-536.

[7] Thrun, S., Burgard, W., & Fox, D. Probabilistic Robotics. MIT Press, 2005.

[8] Zha, H., & Jiang, X. A survey of robot path planning algorithms. Journal of Robotics, 2015, 726969.

[9] Koenig, S., & Likhachev, M. D* Lite. AAAI Conference on Artificial Intelligence, 2002, 476-483.

[10] Zhou, Z., & Hansen, E. Breadth-first heuristic search. Artificial Intelligence, 2004, 170(4-5): 385-408.

[11] Nilsson, N. J. Principles of Artificial Intelligence. Tioga Publishing Company, 1980.

[12] LaValle, S. M., & Kuffner, J. J. Rapidly-Exploring Random Trees: Progress and Prospects. In Algorithmic and Computational Robotics: New Directions, 2001, 293-308.

[13] GPS-based Navigated Autonomous Robot, Last accessed 2024/10/06, https://www.semanticscholar. Org/paper/GPS-based-Navigated-Autonomous-Robot-Al-Faiz-Mahameda.