Journal of Engineering and Technology

ISSN 2180-3811 eISSN 2289-814X https://jet.utem.edu.my/jet/index

OPTIMIZING TASK EFFICIENCY: PATH PLANNING FOR MOBILE ROBOTS

A. S. Ghazali*[1], M. H. Rushdi[1], M. A. M. Razib[1], K. A. Md Nor [1] and Y. M. Mustafah[1]

[1] Kulliyyah of Engineering, International Islamic University Malaysia, P.O. Box 10, 50728 Kuala Lumpur, Malaysia.

corresponding: aimighazali@iium.edu.my

Article history:
Received Date: 26 December 2023
Revised Date: 29 April 2024
Accepted Date: 30 May 2024

Keywords: Mobile robots, Path planning, ROBOCON Malaysia

Abstract— In an ideal scenario, a smart mobile robot should have precise and reliable movements to navigate paths efficiently while performing tasks swiftly, accurately, and seamlessly. However, mobile robots often encounter manoeuvrability constraints, resulting in incorrect movements with a significant margin of error. This project aims to develop a mobile robot capable in planning its path based on path generation time. The examination begins with an assessment of various path planning algorithms in MATLAB. Results show that the Dijkstra path planning algorithm is chosen for implementation, providing the shortest path with the fastest completion time in simulations compared to RRT*. Future projects may explore implementing the Dijkstra algorithm as the path planner on a real robot to analyse system performance in terms of speed, stability, and reliability.

I. Introduction

Mobile robots, operated by software, exhibit autonomous movement without human intervention and the demand in various sectors, such as industrial automation, personal services, construction, and transportation, continues to grow [1]. Despite the popularity, developing mobile robots with flawless manoeuvrability remains challenging, often resulting in incorrect movements with a substantial margin of error [2].

Building and managing mobile robots for competitions, especially those demanding precise and rapid movements, pose difficulties, particularly for autonomous robots planning their paths. Challenges in research and development (R&D) include optimizing tyre positioning for load balancing, sensor selection for positioning and coordination, and more [3]. Therefore, an in-depth understanding of engineering concepts and critical thinking skills are vital for applying hands-on knowledge in robot design, particularly for competitions.

Mapping is a crucial aspect of planning the movements of a mobile robot, involving the robot's ability to map its surrounding environment. In unfamiliar environments, the robot must adapt, and accurate mapping follows precise positioning [4]. Multiple robots can be employed as a method for effective mapping. These robots can communicate and collaborate, integrating and sharing data collected by sensors, such as ultrasonic and Time-Of-Flight sensors, to swiftly and effectively achieve the mapping goal [5]. Once the robot has determined its position and coordination, the final step in achieving an autonomous mobile robot is path planning or navigation.

Path planning involves the machine's ability to sense, strategize, and execute actions [6]. Mobile robots often need to avoid direct paths due to obstacles, making movement planning techniques essential. Path planning algorithms empower robots to navigate complex and dynamic environments both outdoors and indoors [7]. The aim is to discover an optimal or near-optimal path that avoids obstacles [8]. In the early stages, traditional algorithms aimed to provide the best-calculated path, focusing on timing. Over time, traditional algorithms were enhanced with machine learning (ML) techniques, leading to the development of supervised learning, optimal value Reinforcement Learning (RL), and policy gradient RL algorithms [9]. With the advancements in Deep Learning (DP) and RL, path planning has entered a prosperous era, capable of solving complex and non-linear problems [11]. Path planning algorithms encompass Rapidly-exploring Random Tree (RRT), Rapidly-exploring Random Tree Star (RRT*), Dijkstra, and Probabilistic Road Map (PRM) with Dijkstra [10-13].

One notable robotics competition is the Asia-Pacific Broadcasting Union (ABU) ROBOCON, founded in 2002 as an Asian-Oceanic college robot competition [1]. In ROBOCON 2022, themed 'Lagori,' participating teams design mobile robots to accomplish specific tasks within a defined time limit. The game involves two competing teams, the Red Team and the Blue Team, with each match comprising two rounds. In each round, one team assumes the role of the Seeker, while the other becomes the Hitter. For instance, if the Blue Team is the Seeker in the first round, the Red Team takes on the role of the Hitter. The Seeker's task involves using R1

1 https://en.wikipedia.org/wiki/ABU_Robocon

to throw balls to dismantle the Lagori tower, while R2 works to rebuild the tower. Simultaneously, the Hitter (R1 from the Red Team) throws balls to disrupt the activities of the opposing Blue Team. R2 (from the Blue Team) is tasked with retrieving the thrown balls and delivering them to their R1 for counterattacks against the opponents. The project focuses on developing R2, tasked with accurately rebuilding the Lagori tower while evading opponents[2].

In sum, this project is dedicated to evaluating the performance of the Seeker's R2 robot through the selection of path planning algorithms.

II. Methodology

After reviewing and selecting several path planning algorithms as potential options for R2, several algorithms such as Dijkstra, PRM and RRT* are simulated and compared to determine the optimal one for implementation in the competition. The primary parameter to be evaluated is the time taken by each algorithm to complete the path from Point A to Point B (based on Lagori discs' locations) and the distance of the generated path. The test involves placing three goal coordinates on the test field before the algorithm initiates the generation of the optimal path to these goals. The three different goals, namely Goal 1, Goal 2, and Goal 3, are positioned at coordinates [750,450], [450,750], and [150,150], respectively, with [50,450] set as the starting point. Two different configurations will be used for testing the algorithms. The first test requires algorithms to generate the path only from the Start node to Goal 1. In the subsequent test, the algorithms must generate a path from the Start node to Goal 1, then to Goal 2, and finally to Goal 3.

Figure 1 illustrates the start and goal coordinates on the test field.

[2] https://www.youtube.com/playlist?list=PLcmcmEF6geRDb9oR9W-_1A7bD0rEFtpos

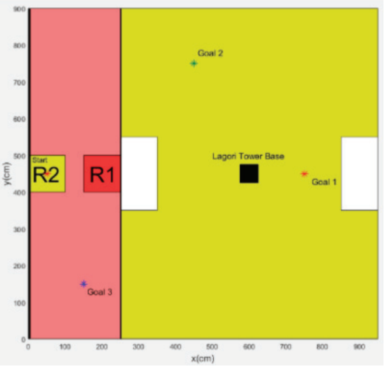


Figure 1: Test field with start and goal nodes

III. Results and Discussion

A. Test field

The MATLAB environment was utilized to construct the proposed test field. Initially, the original test field without any grids was generated, serving as a foundational map, as depicted in Figure 2. MATLAB's provided function was employed to partition the original test field into grids of 50 x 50 cm and 100 x 100 cm throughout the entire test area. The original test field is suitable for implementing RRT* and PRM algorithms, whereas the gridded test field is designed for the Dijkstra algorithm.

B. Path planning

The simulation of various path planning algorithms has led to the identification of the optimal algorithm for Robot R2.

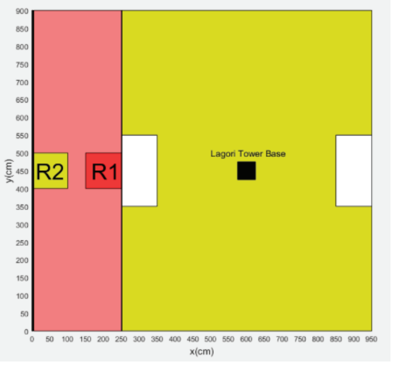


Figure 2: Simplified ROBOCON 2022 field

Completion time (duration) of the algorithms

Table 1 provides a breakdown of the completion time taken by three different algorithms (Dijkstra, PRM + Dijkstra, and RRT*) for two different goals (1 goal vs. 3 goals). Additionally, a comparison of Dijkstra is presented, considering two different field grid sizes (50 x 50 cm vs. 100 x 100 cm).

The average time required to generate optimal paths varies significantly across each algorithm. For both the scenarios of generating paths for 1 goal and 3 goals, the Dijkstra algorithm on the 100 x 100 cm grid test field emerged as the fastest, with an average time below 1 second compared to the other algorithms.

Table 1: Completion path planning based on the algorithms

Number of goals	Algorithms	Path generation time (s)			
		1st test	2nd test	3rd test	Average
1	Dijkstra 50 x 50 cm	45.793	47.131	46.388	46.437
	Dijkstra 100 x 100	0.909	0.926	0.889	0.908
	PRM + Dijkstra	63.733	64.747	61.059	63.180
	RRT*	4.632	6.039	5.464	5.378
3	Dijkstra 50 x 50 cm	43.663	44.906	44.665	44.411
	Dijkstra 100 x 100	0.844	0.822	0.799	0.827
	PRM + Dijkstra	60.875	61.653	61.777	61.435
	RRT*	17.465	15.471	13.707	15.548

The second fastest algorithm in producing optimal paths is RRT*, taking around 5 seconds for each goal coordinate. Considering the competitive context, a lengthy path generation time poses a disadvantage. Consequently, the Dijkstra algorithm for the 50 x 50 cm gridded test field and PRM integrated with Dijkstra are excluded from further tests due to their impracticality for use in competitions.

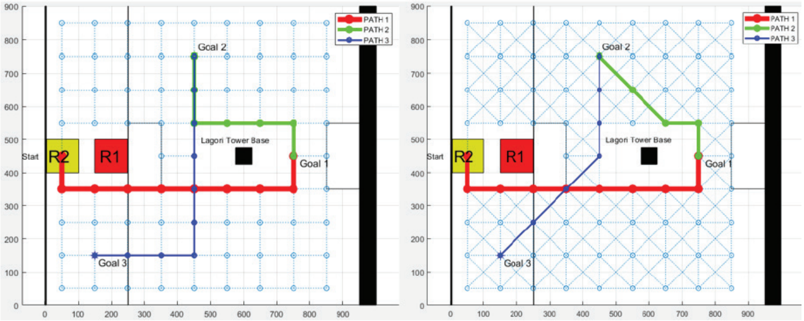
*Enhancement of RRT**

The RRT* algorithm involves three crucial parameters: the maximum number of random nodes generated, the distance between neighbouring nodes, and the distance of enhanced neighbouring nodes. The algorithm initiates by generating random nodes (random coordinates) on the test field. These nodes connect to nearby nodes (neighbouring nodes) to form a path, with nodes recognized as neighbours only if their distance falls within the user-set limit. The enhanced neighbouring nodes work similarly, but new nodes are considered neighbours when generated in proximity. This parameter optimizes the algorithm by creating shorter paths. The number of random nodes serves as a limit, and once reached, the algorithm stops. Increasing this number extends the time for optimal path generation. A higher distance value between neighbouring nodes connects more random nodes, yielding improved optimal paths. Depending on these parameters, the generated optimal path varies. Given limited research on these parameters, the algorithm is simulated with different sets and run three times to find the best path.

The simulation reveals that increasing the number of random nodes extends completion time, and 150 nodes result in over 10 seconds. This poses a significant disadvantage in competition. Simulations with too few nodes may generate paths ignoring obstacles. Hence, 150 random nodes and 50 nodes are omitted due to performance issues. The middle ground of 100 maximum random nodes is selected. With 100 maximum random nodes, the optimal configuration for the RRT* algorithm is determined to be 200 cm for normal and 400 cm for enhanced for the distance of neighbouring nodes.

Enhancement of gridded test field

The 100 x 100 cm gridded test field, which had already shown positive results, can be further optimized by connecting nodes diagonally. This modification takes advantage of R2's holonomic motion system, allowing diagonal movements within the test field. Diagonal movements can shorten the distance between points and, consequently, reduce the length of the optimal path. After running the Dijkstra algorithm on both the original and the improved test fields, the resulting optimal paths are illustrated in Figure 3. A technical analysis of the algorithm's performance on both test fields is presented in Table 2.



(a) (b)

Figure 3: (a) Path generated for the old gridded test field, (b) Path generated for the new gridded test field

Table 2: Distance and completion time for Dijkstra using the old vs new test fields

Test field	Distance (m)			Total distance (m)	Time taken (s)
	Path 1	Path 2	Path 3		
Old	9.00	6.00	9.00	24.00	0.926
New	9.00	4.83	7.24	21.07	1.045

Both Figure 3 and Table 2 clearly indicate that the new gridded test field produces a shorter optimal path compared to the old test field. Although there is a slight increase in the time taken to run the algorithm, it remains within an acceptable range for competition purposes.

Comparison of RRT and Dijkstra algorithms*

In terms of the average distance covered, Dijkstra for the 100 x 100 cm gridded test field outperforms RRT*, with 21.07 m compared to 24.52 m. Additionally, Dijkstra completes the task in a shorter amount of time, with an average time taken of 1.05 s, whereas RRT* takes 8.82 s. In short, Dijkstra for the 100 x 100 cm gridded test field provides a shorter optimal path and completes the path planning algorithm in less time compared to RRT*. Given that the opponent team will target R2 during the competition, R2's swift response becomes a crucial factor for success. Importantly, while the randomness of the path generated by the RRT* algorithm may aid R2 in evading the opposing team's attack, the 8-second window taken by the algorithm to generate its path (during which R2 remains static) is ample time for the opponent to launch an attack. Therefore, it is demonstrated that the Dijkstra algorithm implemented on the 100 x 100 cm gridded test field is the best algorithm for application to R2.

IV. Conclusion

The study evaluates several paths planning algorithms, including Rapidly-expanding Random Tree Star (RRT*), Dijkstra, and Probabilistic Road Map (PRM), to assess their performance. These algorithms are simulated in MATLAB to generate optimal paths for specific goal coordinates on the test field. The simulations compare the algorithms based on the time taken to generate the optimal path and the distance of the path produced. After eliminating impractical algorithm setups and refining the remaining options, the Dijkstra algorithm emerges as the best-performing algorithm, particularly when applied to the 100 x 100 cm gridded test field. The added value of this paper lies specifically in providing insights and guidance for new researchers or beginners in path planning algorithms for robots, particularly in the specialized domain of ROBOCON competitions. Future work may involve utilizing parallel computing and GPU to reduce computation time, particularly when handling large datasets and complex algorithms for path planning. Additionally, it is crucial to note that while this algorithm demonstrates strong performance in simulations, its effectiveness in real-world scenarios remains unproven. To validate the superiority of the Dijkstra algorithm, it must be implemented on a real robot in future experiments.

V. Acknowledgement

The authors gratefully acknowledge the financial support provided by SPG22-039-0039 grant.

VI. References

[1] L. Antonyshyn, J. Silveira, S. Givigi, and J. Marshall, "Multiple mobile robot task and motion planning: A survey," *ACM*

Computing Surveys, vol. 55, no. 10, pp. 1-35, 2023.

[2] F. Rubio, F. Valero, and C. J. I. J. o. A. R. S. Llopis-Albert, "A review of mobile robots: Concepts, methods, theoretical framework, and applications," *International Journal of Advanced Robotic Systems,* vol. 16, no. 2, p. 1729881419839596, 2019.

[3] Y. Yan, S. Wang, Y. Xie, H. Wu, S. Zheng, and H. Li, "Obstacle-circumventing adaptive control of a four-wheeled mobile robot subjected to motion uncertainties," *Frontiers of Mechanical Engineering,* vol. 18, no. 3, pp. 1-21, 2023.

[4] L. Dong, Z. He, C. Song, and C. Sun, "A review of mobile robot motion planning methods: from classical motion planning workflows to reinforcement learning-based architectures," *Journal of Systems Engineering,* vol. 34, no. 2, pp. 439-459, 2023.

[5] A. N. Jati and R. E. Saputra, "Coordination control for simple autonomous mobile robot," in *2017 5th International Conference on Instrumentation, Control, and Automation (ICA)*, 2017, pp. 93-98: IEEE.

[6] X. Xiao, B. Liu, G. Warnell, and P. Stone, "Motion planning and control for mobile robot navigation using machine learning: a survey," *Autonomous Robots,* vol. 46, no. 5, pp. 569-597, 2022.

[7] L. Chang, L. Shan, C. Jiang, and Y. Dai, "Reinforcement based mobile robot path planning with improved dynamic window approach in unknown environment," *Autonomous Robots,* vol. 45, pp. 51-76, 2021.

[8] A. Vinayak, M. Zakaria, and K. J. J. o. E. Baarath, "A Comprehensive Review on Different Path Planning Methods for Autonomous Vehicles," *Journal of Engineering Technology,* vol. 14, no. 2, 2023.

[9] J. Oh *et al.*, "Discovering reinforcement learning algorithms," *Advances in Neural Information Processing Systems,* vol. 33, pp. 1060-1070, 2020.

[10] C. Zhou, B. Huang, and P. Fränti, "A review of motion planning algorithms for intelligent robots," *Journal of Intelligent Manufacturing,* vol. 33, no. 2, pp. 387-424, 2022.

[11] Y. Ming, Y. Li, Z. Zhang, and W. Yan, "A survey of path planning algorithms for autonomous vehicles," *SAE International Journal of Commercial Vehicles,* vol. 14, no. 02-14-01-0007, pp. 97-109, 2021.

[12] M. Dirik and F. Kocamaz, "Rrt-dijkstra: An improved path planning algorithm for mobile robots," *Journal of Soft COmputing Artificial Intelligence,* vol. 1, no. 2, pp. 69-77, 2020.

[13] R. Geraerts and M. H. Overmars, "A comparative study of probabilistic roadmap planners," in *Algorithmic Foundations of Robotics V*: Springer, 2004, pp. 43-57.