

Low-carbon Path Optimization for Intelligent Material Transport in Indoor Environments

Yu-Jen Hsu,¹ Ting-En Wu,² and Shu-Hsien Huang^{3*}

¹Department of Computer Science & Information Engineering, National Kaohsiung University of Science and Technology, No. 415, Jiangong Rd., Sanmin Dist., Kaohsiung City 807618, Taiwan (R.O.C.)

²Department of Aeronautics and Astronautics, National Cheng Kung University,
No. 1, University Road, Tainan City 70101, Taiwan (R.O.C.)

³Department of Information Management, National Chin-Yi University of Technology,
No. 57, Sec. 2, Zhongshan Rd., Taiping Dist., Taichung 411030, Taiwan (R.O.C.)

(Received July 28, 2025; accepted November 6, 2025)

Keywords: A*, RRT, RRT*, low-carbon path planning, autonomous mobile robot, energy-aware navigation

In this study, we conducted a low-carbon-oriented performance comparison of three mainstream path planning algorithms, A-star (A*), Rapidly exploring random tree (RRT), and Rapidly exploring random tree-star (RRT*), under a unified platform. Through image-based map simulation and obstacle inflation processing, a consistent experimental environment is established with fixed start and goal points set to ensure comparability. The experimental evaluation metrics include total path length, algorithm computation time, node expansion behavior, and path smoothness. The results show that A* performs best in structured environments, producing the shortest path. Although RRT has fast exploration capabilities, it tends to generate irregular trajectories. RRT* improves path quality through a node rewiring mechanism, making it suitable for carbon-sensitive scenarios, but with higher computational cost. Overall, the results of this study fill the gap in previous carbon-sensitive navigation research by providing a cross-algorithm comparison. It offers an empirical foundation and visual reference for selecting appropriate strategies in future low-carbon autonomous mobility systems.

1. Introduction

With the rapid development of fields such as smart manufacturing, urban logistics, autonomous mobile robots (AMRs), and smart agriculture, robotic autonomous navigation technology is attracting increasing attention.⁽¹⁾ Recent advances in smart manufacturing have shown that deep learning can significantly enhance defect detection. Lai proposed a welding defect identification system using AI, demonstrating the effectiveness of intelligent sensing in industrial environments.⁽²⁾ Traditionally, the goal of path planning has focused on the shortest distance or minimum time, with representative algorithms primarily oriented toward geometric shortest paths and minimal turning costs. However, in response to global climate change and

*Corresponding author: e-mail: sh.huang@ncut.edu.tw
<https://doi.org/10.18494/SAM5865>

increasing carbon emission regulations, the carbon sensitivity and energy efficiency of robotic paths are gradually being incorporated into the core of planning and optimization design.⁽³⁾

In practical applications, mobile robots consume electrical energy during operation and indirectly generate carbon emissions. Particularly in enclosed environments such as warehouses, factories, or campuses, their travel distances and frequencies are often considerable. Without proper navigation strategies, this can lead to cumulative energy consumption and environmental burden.⁽⁴⁾ Therefore, beyond traditional geometric costs, researchers have begun incorporating energy models, carbon estimation, and spatial awareness models to design navigation strategies that balance operational efficiency and sustainability.⁽⁵⁾ Among these, carbon-aware path planning is regarded as a key breakthrough for robotic carbon reduction applications.

In recent years, numerous studies have indicated that systems capable of integrating carbon costs into planning algorithms often outperform traditional models in energy consumption control and sustainability performance. For example, Natarajan *et al.*⁽⁶⁾ proposed a modified A-star (A*) model with carbon emissions as the objective function, significantly reducing total carbon consumption in both simulations and real-world settings.⁽⁶⁾ The integration of real-time visual sensing technologies, as demonstrated by Nasir *et al.*,⁽⁷⁾ offers promising directions for improving path planning reliability and contextual awareness in urban or semi-structured environments.

However, although the incorporation of carbon cost considerations into navigation strategies has been attempted in various methods, performance comparisons of mainstream algorithms such as A*, Rapidly exploring random tree (RRT), and Rapidly exploring random tree-star (RRT*) in the same environment remain rare. Previous studies mostly involved individual designs and lacked a unified map and visual feedback platform, which hindered cross-comparison and the establishment of a basis for selection.⁽⁸⁾ Therefore, in this study, we aim to build a consistent image-based simulation platform. Through image preprocessing and algorithm modularization, low-carbon-oriented experimental design and performance analysis are conducted for three common algorithms. Through this study, more empirical and visual evidence can be provided for the development of robot carbon-sensitive navigation systems, filling the research lack of comparative perspectives in existing literature.

2. Theoretical Framework

2.1 Demands and challenges of low-carbon-oriented robot path planning

Against the backdrop of rapid development in smart manufacturing, warehouse logistics, and agricultural automation, the deployment of AMRs has emerged as a vital strategy to enhance productivity and reduce human labor demands. The integration of lightweight structural components,⁽⁹⁾ the application of energy-efficient propulsion systems,⁽¹⁰⁾ and the advancement of materials science for agricultural and environmental sensors⁽¹¹⁾ have collectively accelerated the need for precise and adaptive path planning. Consequently, research on trajectory optimization

for such robots in structured and semi-structured environments has gained momentum. Against the backdrop of rapid growth in smart factories, warehouse logistics, and agricultural automation, the introduction of autonomous navigation robots has become an important means to improve operational efficiency and reduce labor costs. However, traditional navigation systems mostly focus only on minimizing the geometric shortest distance and operation time, neglecting their cumulative impact on energy consumption and carbon emissions over long-term operation.⁽¹²⁾ With the promotion of sustainable development and carbon neutrality policies, low-carbon-oriented navigation strategies are receiving increasing attention from the research community and industry, especially in application scenarios involving high-frequency operations within fixed environments, where balancing navigation efficiency and carbon emission costs becomes a key design challenge.⁽¹³⁾

However, these algorithms are mostly designed for specific application scenarios or individual optimized models and lack a horizontally integrated and fair comparative experimental platform, making it difficult for users to choose the most suitable algorithm framework based on task requirements. Particularly in carbon cost-sensitive applications, the lack of standardized visual feedback and quantitative comparison mechanisms limits the promotion and adoption of this technology in broader environments.⁽¹⁴⁾ Therefore, constructing a unified and open-source simulation testing platform that integrates mainstream methods, such as A*, RRT, and RRT*, and evaluates them on the basis of carbon-sensitive metrics, will help fill the gaps in existing research and provide a basis for selection in practical deployment.

2.2 Exploration of A* algorithm and carbon-aware path optimization application

Since its introduction, the A* algorithm has been widely applied in autonomous navigation and shortest path planning on image-based maps owing to its combination of shortest distance guarantee and heuristic search efficiency. Its core operation mechanism is shown by Eq. (1).

$$S = (k_1 + k_2) \iint q \, ds \, d\theta \quad (1)$$

Here, $g(v)$ represents the path cost from the start point to v (Dijkstra's algorithm). $h(v)$ is the heuristic path estimate from v to the goal (BFS algorithm). The algorithm prioritizes nodes with the smallest $f(v)$. It continuously updates the path information of neighboring nodes until the shortest path to the goal is found. Typically, Euclidean distance or Manhattan distance is used as the heuristic evaluation.⁽¹⁵⁾ In static and structured environments, such as inside factories and automated warehouses, A* can provide highly reliable and consistent shortest paths, and can incorporate obstacle expansion to ensure navigation safety.

However, the traditional A* has limited sensitivity to energy consumption and carbon emissions. In recent years, various groups have proposed incorporating energy consumption models or carbon weighting into its heuristic function, transforming it into a carbon-aware path optimization problem. For example, Wang *et al.*⁽¹⁶⁾ transformed the estimated carbon emission rate into spatial weight distributions, incorporated these weights into the $h(v)$ evaluation, and

designed the Energy-Aware A* algorithm, which effectively reduced energy consumption by nearly 15% in simulated warehouse environments.⁽¹⁶⁾ Han *et al.* performed feature recognition on hotspots and high-energy-consumption areas within indoor spaces, combining heatmaps with grayscale images to adjust the A* costmap, enhancing its ability to identify low-carbon paths.⁽¹⁷⁾

Overall, the A* algorithm possesses both implementation and simulation advantages in carbon-aware navigation because of its high adjustability and simple structure. In the future, further integration of real-time sensing feedback mechanisms combined with reinforcement learning for costmap self-learning holds promise to become one of the most stable and scalable core modules in low-carbon autonomous navigation systems.

2.3 Application exploration of RRT, RRT* algorithm, and carbon-aware path optimization

RRT is a well-known sampling-based path planning algorithm with the ability to rapidly explore feasible regions in high-dimensional spaces. It primarily expands the tree structure by repeatedly sampling random points q_{rand} , finding the nearest node q_{near} , and extending toward it to a new node q_{new} , until the goal is reached or the maximum number of samples is met.⁽¹⁸⁾ RRT has high exploration efficiency and simple implementation, making it particularly suitable for handling complex geometric and dynamic environment problems. However, its random nature causes the generated paths to be suboptimal and often discontinuous and jagged, which increases the actual travel distance and energy consumption.

Furthermore, Yuan and colleagues developed adaptive sampling RRT, which integrates environmental energy monitoring data to dynamically adjust the sampling density in real time. Nonetheless, challenges remain for RRT in low-carbon applications. Its unstable expansion directions and nonreproducible sampling results may cause the system to generate differing or even inefficient paths within the same environment.⁽¹⁹⁾ Additionally, RRT lacks local optimization and node rewiring capabilities, limiting its performance in avoiding carbon hotspots and in final path optimization.⁽²⁰⁾ Therefore, subsequent research has been focused on improving RRT stability and energy efficiency sensitivity, laying a crucial foundation for the evolution and application of RRT*.

3. Experimental Design

To verify the performance differences and application potential of the three algorithms A*, RRT, and RRT* under the same environmental conditions, we constructed an experimental platform equipped with an “image-based map environment”, “carbon-sensitive parameter simulation”, and “visualized path analysis” functions. The experimental design is aimed at investigating the path quality, total length, computational efficiency, and environmental adaptability of the three methods in static 2D spaces and further analyzing their feasibility in low-carbon-oriented robot mobility tasks.

The environment modeling adopted in this study is a grayscale image (PNG format) of a real warehouse map with a resolution of 640×480 pixels. The original map undergoes image

processing procedures, including grayscale binarization (with a fixed threshold of 127) and morphological dilation (using a 5×5 convolution kernel to simulate safety distance), to form a binary obstacle map suitable for algorithm operation. Obstacle areas are represented by a grayscale value of 0, indicating non-traversable regions; the remaining areas have a value of 255, representing traversable regions.

In each experiment, the start and goal points are fixed at coordinates $S = (100, 400)$ and $G = (550, 150)$, respectively, with coordinate transformation applied so that the bottom-left corner serves as the origin. Each algorithm is executed under the same environment to ensure fair comparison. Each algorithm is run five times to reduce bias caused by randomness, and the average is taken as the evaluation basis. The platform hardware comprises an Intel Core i7 processor with 16 GB of RAM running on Python 3.10 and OpenCV framework.

The A* algorithm is configured with a grid unit of 2 pixels, expanding in eight directions (up, down, left, right, and diagonals). The heuristic function is the Euclidean distance, aiming to minimize $f(v) = g(v) + h(v)$. RRT and RRT* are sampling-based algorithms in continuous space, with a maximum step size set to 20 pixels and a sampling limit of 2000 iterations. RRT* additionally sets a rewiring radius $r(v) = \gamma(\log n/n)^{1/2}$, where γ is empirically set to 30, enabling semi-local optimization.

The evaluation indicators include the following four items.

- (1) Total Path Length: Calculate the sum of Euclidean distances between all adjacent points along the path from the start to the goal, simulating actual movement energy consumption.
- (2) Computation Time: Computation time is defined as the elapsed time (in seconds) from the start of the algorithm to the acquisition of the first feasible path.
- (3) Expanded Nodes or Sampled Nodes: Represent the computational resources and redundancy required by the algorithm within the search space.
- (4) Path Smoothness and Number of Direction Changes: The number of directional changes between segmented path lines is used as an approximate measure of path tortuosity, supplemented by visual representation.

In addition, in this study, we design a “carbon sensitivity simulation mechanism” by defining virtual high-carbon hotspot zones on the map (simulated using grayscale values from 128 to 200). The algorithm must avoid these areas as much as possible to align with low-carbon planning objectives. This configuration is applicable to the extended versions of the RRT* and A* algorithms and can serve as a foundation for incorporating energy cost functions in future research.

The overall experimental framework is illustrated in Fig. 1. The paths generated by each algorithm are visualized as colored trajectory lines using OpenCV, and their execution results and statistical data are recorded. Final analysis is conducted using integrated tables and path diagrams. This experimental design effectively demonstrates the differences among algorithms in low-carbon-oriented applications and their potential application scenarios.

4. Results and Discussion

To evaluate the performance and behavioral characteristics of different path planning algorithms, we conducted a series of experiments under various map configurations, with fixed

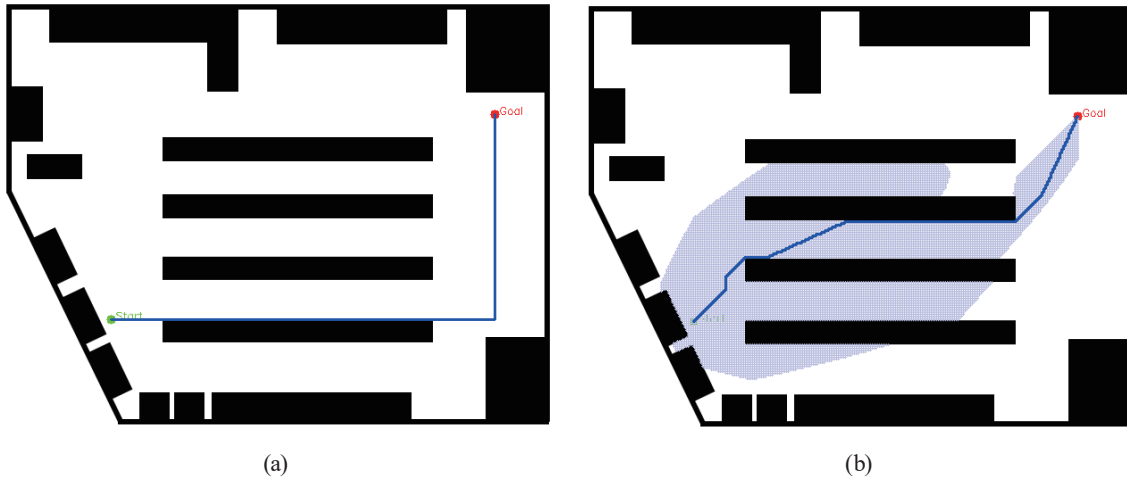


Fig. 1. (Color online) Path planning comparison. (a) Original path. (b) A* algorithm.

start and goal positions in each setup. The following methods were compared in the experiments: baseline (original) path, A*, RRT, and RRT*. Each method was evaluated on the basis of the total path length as well as qualitative aspects such as path smoothness, spatial efficiency, and adherence to free-space corridors.

4.1 Case 1

The RRT* algorithm is an improved version developed to overcome the lack of optimality and rewiring capability in traditional RRT.⁽³⁾ Its core lies in the node rewiring mechanism, which searches the set of neighboring nodes around the newly added node q_{new} and evaluates whether rewiring these neighbors through q_{new} can reduce the total cost. This semilocal optimization mechanism enables RRT* to converge to the shortest path under sufficient time and sampling conditions, possessing asymptotic optimality.

Figure 1(a) shows the original path result applied to the same map configuration, serving as the comparison baseline. Although the path is valid and collision-free, it does not account for the actual spatial structure or distribution of obstacles in the environment. As a result, the path traverses longer segments than necessary and fails to utilize more direct shortcuts. Figure 1(b) shows the result of applying the A* algorithm to the same map configuration. The blue shaded area represents the set of explored nodes, illustrating the spatial breadth of the algorithm's search. The final path is more compact and closely aligned with the geometric center of the free-space corridor. The results of RRT and RRT* path planning are shown in Fig. 2.

Figure 2(a) shows the path generated by the RRT algorithm. This algorithm employs a rapidly exploring tree structure based on uniform random sampling. Because of the lack of global cost optimization, the path exhibits high curvature, especially in the middle and final segments. The algorithm tends to prioritize rapid expansion over refinement, which, while enabling effective

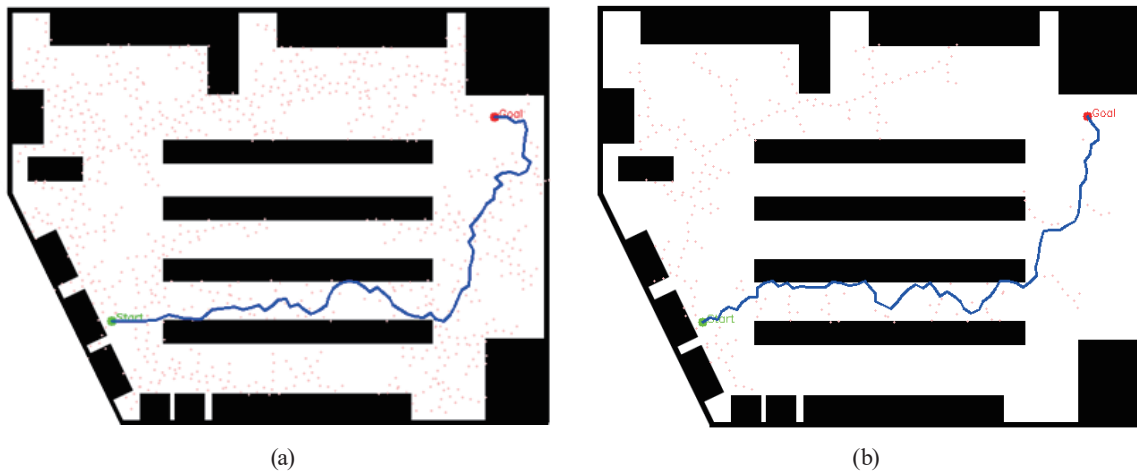


Fig. 2. (Color online) Path planning results: (a) RRT algorithm and (b) RRT* algorithm.

spatial coverage, also leads to irregular path quality. Additionally, the distribution of red sampling points throughout the environment highlights the algorithm's randomness; many nodes are sampled but, because of early pruning or collisions with obstacles, ultimately do not contribute to the final path. Figure 2(b) shows the results of RRT*. RRT* retains the core structure of RRT while incorporating a local rewiring mechanism to improve path efficiency. The results show that the trajectory is noticeably smoother and more direct than that obtained using RRT with fewer abrupt direction changes and better alignment. The total path length evaluation is presented in Table 1.

The original path has a total length of 660.00 pixel units and does not consider environmental constraints. Although this path appears geometrically simple and requires low computational effort, its lack of environmental awareness can lead to inefficiency in dynamic or constrained environments. The path generated by the A* algorithm is the shortest among all methods, with a length of 561.59 pixel units. This result demonstrates the strong advantage of the A* algorithm in producing optimal solutions within structured grid environments. By combining actual movement costs and heuristic estimates, the algorithm efficiently explores the most promising areas of the map, minimizing redundant expansions while avoiding close proximity to obstacles. Using the A* algorithm reduces the path length by 14.90% compared with the original path. The trajectory generated by the RRT-based method is noticeably longer, with a total length of 704.00 pixel units. This result aligns with the inherent randomness of the algorithm and its lack of global optimization. Although RRT demonstrates strong capability in rapidly exploring free space, its greedy expansion strategy often produces irregular and suboptimal paths. The path generated by the RRT* algorithm is even longer, reaching 756.70 pixel units. Although RRT* includes a local rewiring mechanism aimed at improving path efficiency, this result highlights a practical limitation: under constrained iteration budgets and dense obstacle distributions, the rewiring process may fail to sufficiently converge to the optimal path.

Table 1
Total path length evaluation.

Path search	Total path length
Original path	660.00
A*	561.59
RRT	704.00
RRT*	756.70

4.2 Case 2

Figure 3(a) shows that the route fails to utilize available diagonal corridors or more direct paths to avoid excessive turning. The rigid path structure may be suitable for highly controlled or static environments but has limited adaptability and often results in increased travel distance especially in spaces where obstacle configurations change or more efficient diagonal routes can be formed. In Fig. 3(b), the A* algorithm result demonstrates a more efficient utilization of the navigable space. The shaded area represents the region explored by the algorithm, indicating a focused and heuristic-driven expansion strategy. The final path fully leverages the open corridors between obstacles and minimizes total distance by aligning more closely with the geometric center of the free space.

In Fig. 4(a), the path generated by RRT exhibits significant irregularity in its trajectory, especially in the early and middle segments. The path shows sharp turns and inconsistent directionality, with considerable deviation from the geometric center of the available free space. These characteristics stem from the RRT algorithm's tendency to explore random samples without considering the cumulative path cost. Although the algorithm ultimately finds a feasible solution, its reactive nature and lack of optimization result in an extended final path length. In Fig. 4(b), the result of the RRT* algorithm shows significant improvement in both path smoothness and trajectory compactness. The route avoids unnecessary oscillations and aligns more consistently with the global direction between the start and goal points. This improvement is primarily attributed to RRT*'s rewiring mechanism, which gradually reduces local costs through neighborhood evaluation and tree restructuring. Although the sampling distributions of RRT and RRT* are roughly similar, the trajectory obtained by RRT* is more refined, with smoother transitions between points and overall lower curvature.

The original path with a total length of 700.00 pixel units does not consider environmental constraints. Although feasible, this result does not reflect adaptability to surrounding spatial limitations. The rigid shape introduces unnecessary detours and fails to utilize more direct open areas between obstacles. The A* algorithm demonstrates significant improvement, producing a final path length of 593.39 pixel units. Using the A* algorithm reduces the path length by 15.23% compared with the original path. The A* algorithm effectively prunes less desirable paths and focuses computational effort on directions likely to yield shorter routes. The RRT algorithm generated a path of 802.40 pixels. Although the resulting trajectory is valid, it exhibits issues of random curvature and inefficient path segments. The RRT* algorithm shortens the total path

length to 726.26 pixels through local optimization via rewiring, as show in Table 2. Although this result is still longer than the output of A*, it represents a significant improvement over RRT. The path generated by RRT* is smoother and better aligned with the target direction.

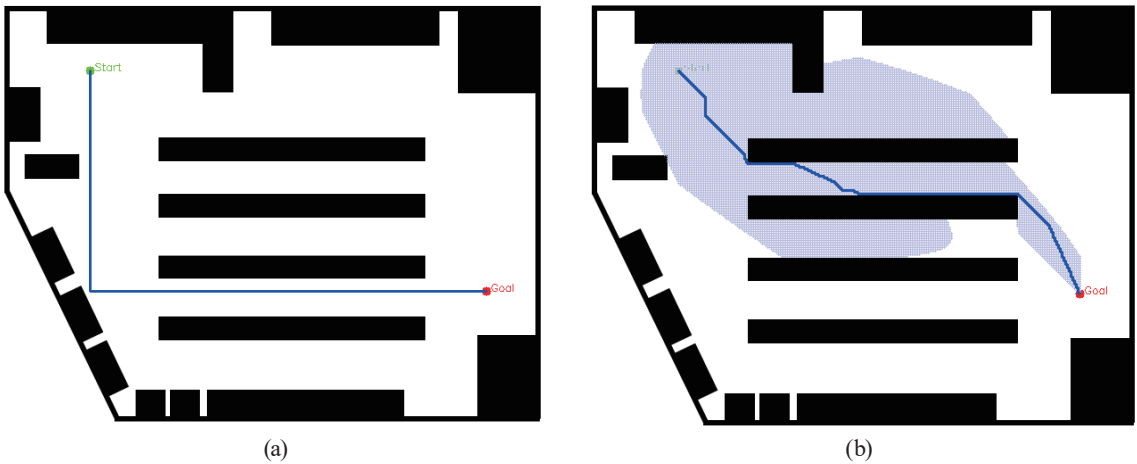


Fig. 3. (Color online) Path planning results. (a) Original path. (b) A* algorithm.

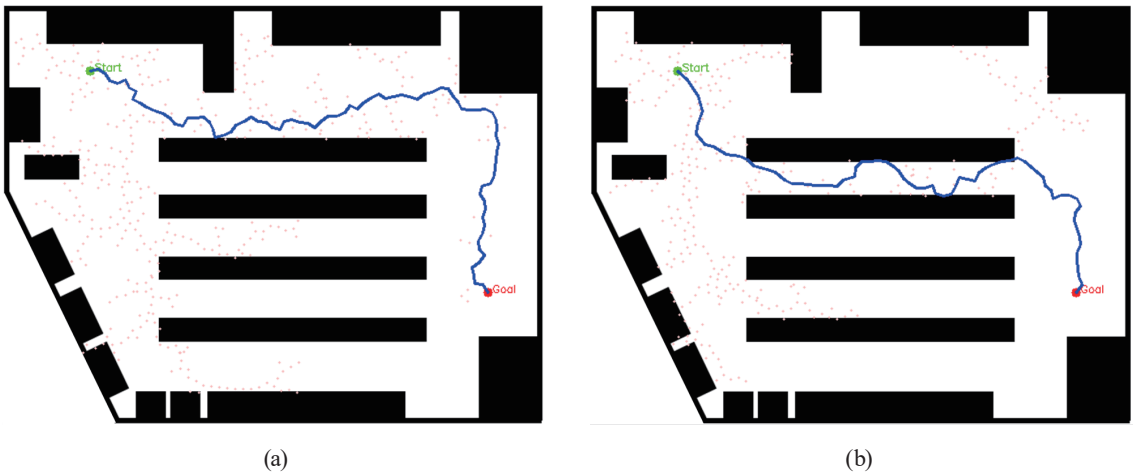


Fig. 4. (Color online) Path planning results. (a) RRT algorithm. (b) RRT* algorithm.

Table 2
Total path length evaluation.

Path search	Total path length
Original path	700.00
A*	593.39
RRT	802.40
RRT*	726.26

5. Conclusions

In this study, we conducted a comprehensive comparison of four path planning strategies—original path, A*, RRT, and RRT*—across various obstacle-dense environments. Through visual inspection and quantitative evaluation of the total path length, the results revealed the trade-offs between planners based on deterministic rules and those utilizing heuristic or random sampling algorithms. From the results of the above analyses, the following conclusions are drawn.

- (1) The A* algorithm generated the shortest path in both test cases, producing routes that are the shortest and most spatially efficient.
- (2) The RRT algorithm demonstrated rapid exploration capability, but because of its greedy expansion behavior, it often resulted in suboptimal paths.
- (3) RRT* improved upon the results of RRT by incorporating a rewiring mechanism, producing smoother and more compact trajectories, albeit with increased computational load.

In the future, incorporating dynamic obstacle scenarios will help in the evaluation of the efficiency of replanning and real-time adaptability. Integrating motion constraints such as speed, curvature, or kinematic models can better simulate the real-world limitations of robots for practical deployment. Implementing the evaluation framework on physical robots will help validate planning efficiency, smoothness, and robustness in real-world applications involving sensor noise, localization uncertainty, and actuator constraints.

References

- 1 J. Korczak and K. Kijewska: *Transp. Res. Procedia* **39** (2019) 201. <https://doi.org/10.1016/j.trpro.2019.06.022>.
- 2 C. H. Lai: *J. Chin. Soc. Mech. Eng.* **45** (2024) 453. <https://doi.org/10.29979/JCSME>
- 3 L. Chen, Z. Chen, Y. Zhang, Y. Liu, A. I. Osman, M. Farghali, J. Hua, A. Al-Fatesh, I. Ihara, and D. W. Rooney: *Environ. Chem. Lett.* **21** (2023) 2525. <https://doi.org/10.1007/s10311-023-01617-y>
- 4 A. M. Omer: *J. Renewable and Sustainable Energy* **1** (2009) 053101. <https://doi.org/10.1063/1.3220701>
- 5 B. Thirumalaiyammal, P. Steffi, and P. Bhambri: *Handbook of Technological Sustainability*, P. Bhambri and P. Bajdor Eds. (CRC Press, 2024) pp. 292–304.
- 6 S. Natarajan, J. Padget, and L. Elliott: *Energy Build.* **43** (2011) 2602.
- 7 Y. C. Leng, C. L. Tai, and C. L. Chen: *J. Chin. Soc. Mech. Eng.* **37** (2016) 201. <https://doi.org/10.29979/JCSME>
- 8 J. Nasir, F. Islam, U. Malik, Y. Ayaz, O. Hasan, M. Khan, and M. S. Muhammad: *Int. J. Adv. Rob. Syst.* **10** (2013) 299. <https://doi.org/10.5772/56718>
- 9 M. D. Ding, Z. H. Fan, J. Y. Wang, B. Liu, F. Q. Zhai, and L. Li: *J. Chin. Soc. Mech. Eng.* **44** (2023) 231. <https://doi.org/10.29979/JCSME>
- 10 G. K. Deshmukh, A. Rehman, and R. Gupta: *J. Chin. Soc. Mech. Eng.* **43** (2022) 347. <https://doi.org/10.29979/JCSME>
- 11 N. M. Dwidiani, N. P. G. Suardana, I. N. G. Wardana, N. W. Satrio, I. G. K. Puja, I. Santhiarsa, W. N. Septiadi, and A. A. A. Suryawan: *J. Chin. Soc. Mech. Eng.* **45** (2024) 375. <https://doi.org/10.29979/JCSME>
- 12 Z. Javanmard and C. Nava: *Green Technol. Sustainability* **3** (2025) 100222. <https://doi.org/10.1016/j.grets.2025.100222>
- 13 S.-J. Heo and W. S. Na: *Electron.* **14** (2025) 227. <https://doi.org/10.3390/electronics14020227>
- 14 A. Muhebwaa and K. K. Osman: *ACM J. Comput. Sustainable Soc.* **3** (2025) 1. <https://doi.org/10.3390/sul7219529>
- 15 J. Zhang, J. Wu, X. Shen, and Y. Li: *Int. J. Adv. Rob. Syst.* **18** (2021).
- 16 X. Wang, Y. Cai, G. Liu, M. Zhang, Y. Bai, and F. Zhang: *Ecol. Inf.* **70** (2022) 101759. <https://doi.org/10.3390/land14020245>
- 17 Z. Han, H. Sun, J. Huang, J. Xu, Y. Tang, and X. Liu: *World Electric Vehicle J.* **15** (2024) 322. <https://doi.org/10.3390/wevj15070322>

- 18 T. Xu: Heliyon **10** (2024) e32451. <https://doi.org/10.1016/j.heliyon.2024.e32451>
- 19 G. Benedetti, O. Solarin, C. Miller, G. Tystahl, W. Enck, C. Kästner, A. Kapravelos, A. Merlo, and L. Verderame: 2025 IEEE/ACM 47th Int. Conf. Software Engineering (ICSE) (2025) 673. <https://doi.org/10.1109/ICSE55347.2025.00136>
- 20 Q. Guo, R. Li, C. Zheng, and G. Jeon: PeerJ Computer Science **11** (2025) e2611. <https://doi.org/10.7717/peerj-cs.2611>

About the Authors



Yu-Jen Hsu received her Ph.D. degree from the National University of Tainan, Taiwan, in 2012. From 2012 to 2015, she was an assistant professor at the Department of e-Learning Design & Management, National Chiayi University, Taiwan. Since 2018, she has been a postdoctoral fellow at the Graduate Institute of Digital Learning and Education, National Taiwan University of Science and Technology. Her research interests are in NLP & dialogue management, AI-guided learning system design, and virtual character-based facilitation platforms. (yuren925@gmail.com)



Ting-En Wu is currently studying for his M.S. degree at the Department of Aeronautics and Astronautics, National Cheng Kung University. His research direction lies in gear design, image recognition, and vibration analysis. (p46144382@gs.ncku.edu.tw)



Shu-Hsien Huang is currently an assistant professor in the Department of Information Management at National Chin-Yi University of Technology. She received her Ph.D. degree from National Cheng Kung University, Taiwan, in 2014. Her research interests are in Internet of Things, sensors, virtual reality, and cyber-physics. (sh.huang@ncut.edu.tw)