



OPEN

Self-adaptive search algorithm for path planning based on the A* algorithm

Shiwei Lin¹✉, Xiangxi Fan¹, Zhixuan Xie¹, Yundong Wu¹ & Hongwu Yang²

The A* algorithm plays an important role in global path planning for robots, but it faces challenges such as redundant nodes and large search spaces. This paper proposes the Obstacle Density-based Dynamic Exponential A* (ODDEA*) algorithm. The ODDEA* algorithm adjusts the weights of the heuristic function based on the density of the surrounding obstacles. It uses the improved heuristic function to guide the robot toward areas with low obstacle density, employing a local dynamic penalty. The computational experiments compare the proposed ODDEA* algorithm with the Theta*, A*, and BA* algorithms, involving small-size (20×20), medium-size (40×40), and large-size (60×60) grid maps, as well as 50 random medium-size maps. The proposed ODDEA* algorithm uses fewer expanded nodes and less planning time than the other algorithms. Compared with the A* algorithm, it achieves 46.96% of the planning time and 20.33% of the search space on the three fixed grid maps.

The path planning algorithms search for the collision-free path between the start and end points while satisfying the evaluation metrics¹. These algorithms are classified as the classical approaches, heuristic algorithms, AI-based and bio-inspired approaches². The A* search algorithm is one of the most popular path planning algorithms, which has the benefits of fast path searching and computational speed^{3,4}. However, the A* search algorithm requires more search nodes and space for the complicated environment, which reduces search efficiency and wastes the computational resources⁵. The A* algorithm faces the challenges of redundant nodes during expansion, and sharp turning paths during path planning⁶.

This paper proposes a novel dynamic search algorithm based on the A* algorithm, which considers obstacle density as the dynamic heuristic function, named Obstacle Density-based Dynamic Exponential A* Algorithm (ODDEA*). The proposed ODDEA* algorithm aims to obtain the optimal path without excessive redundant nodes, and it achieves fast computational speed. Its main contributions are as below:

- A novel dynamic exponential heuristic function based on Obstacle Density to enhance the search ability.
- The proposed ODDEA* algorithm uses local dynamic penalty, adaptive weight adjustment, and multi-variable fusion to reduce the redundant search space.
- It is developed based on the Chebyshev distance, and computational experiments were performed on different sizes of grid maps.

This paper is organised as follows. Section “[Related work](#)” provides a review of the related work, and “[Preliminary knowledge](#)” explains the A* algorithm. The proposed ODDEA* algorithm is presented in “[Obstacle density-based dynamic exponential A* Algorithm \(ODDEA*\)](#)”. The computational experiments and the comparison analysis are in “[Computational experiments](#)”, and the paper is concluded in “[Conclusion](#)”.

Related work

The A* algorithm is one of the popular graph search algorithms. A shorter path can be achieved by connecting line segments between two path points to improve the A* algorithm for path planning⁷. The filter function is employed to avoid irregular paths and large turning angles, and integrated with the cubic B-spline interpolation to smooth the path, using the Euler and Manhattan distances for estimating the cost³.

⁴ presents an improved A* algorithm by expanding the search directions to 12 directions to reduce the total path length and inflection points. An improved multi-objective A* algorithm for robot path planning using the Euclidean distance is presented to reduce the number of random points, path length, and process time⁸. Five-neighborhood and eight-neighborhood searches are used with weight coefficient improvement as an improved A* algorithm⁶.

¹School of Computer Engineering, Jimei University, Xiamen 361000, Fujian, China. ²Xiamen Topstar Co., Ltd., Xiamen 361000, Fujian, China. ✉email: Shiwei.Lin@jmu.edu.cn

The A* algorithm is combined with the artificial potential field (APF) method for a home cleaning robot using the Euclidean distance¹. Another combination algorithm of global and local path planning is ASL-DWA, which combines A* and the Dynamic Window Approach (DWA), with the hybrid heuristic function of point-to-line distance and Euclidean distance⁵. An adaptive A* algorithm can reduce redundant points, and it is integrated with DWA as a fusion algorithm⁹.

A dynamic anti-collision A* algorithm is introduced in¹⁰ for complex scenarios, which considers time factors and the navigation risk cost. R5DOS model is combined with the A* algorithm to visit fewer nodes and to modify the hop search algorithm^{11 12} uses a guideline and key points for the bidirectional search to solve the problem that the A* algorithm requires a large memory space. A bidirectional alternating search strategy is combined with the A* algorithm to improve search efficiency¹³.

For AGV path planning, a modified cost function based on diverse turning angles is proposed with the artificial potential field method to address the challenges of turning points and angles¹⁴. For a hybrid A* algorithm, the Voronoi field potential is employed for the safety-enhanced design, and a multi-stage dynamic optimisation strategy is used for the efficiency-enhance design¹⁵.

Additionally, for robot path planning, there are many heuristic algorithms used to generate the optimal path. Inspired by Particle Swarm Optimisation (PSO) and Simulated Annealing (SA), PSO-SA is proposed to get rid of the local optimum and enhance the search efficiency¹⁶. Multi-Objective PSO is enhanced with a fuzzy inference system for car-like robot path planning¹⁷. The Ant Colony Optimisation (ACO) algorithm is improved by an ant communication mechanism and path selection rules¹⁸, and¹⁹ presents a new heuristic mechanism for the ACO algorithm. The sampling-based approach is also used in robot path planning, such as the Rapidly-exploring Random Tree Star (RRT*) algorithm^{20 21}.

Reinforcement learning is a recent trend in path planning.²² incorporates Q-learning and the APF method to enhance performance and learning speed. An improved Dyna-Q algorithm utilises the Dyna architecture, combining reactive navigation, Simulated Annealing (SA), and heuristic search with Q-learning²³. Long Short-Term Memory (LSTM) is used to optimise the network structure of the Deep Deterministic Policy Gradient (DDPG) algorithm to enhance the moving speed²⁴. The transformer structure is introduced into imitative reinforcement learning to enhance the policy neural networks for feature extraction²⁵.

However, the A* algorithm is widely used in industrial applications due to its completeness of path generation and fast computational speed. The A* algorithm faces the challenges of large turning angles or sharp turning paths, redundant nodes during expansion, and large search spaces. This paper aims to address these problems, and it uses a dynamic heuristic function to guide the robot to achieve the shortest path safely.

Problem statement

The grid map is employed to represent the environment, as shown in Fig. 1. The white grids stand for the accessible area, while the black grids represent the obstacles. They are modelled as a binary map, where 1 indicates obstacles, while 0 denotes accessible areas. The sizes of the grid maps in the computational experiments include 20x20, 40x40, and 60x60 maps. M stands for the map matrix in Eq.(1), and the grid at the i th row and the j th column is represented in Eq.(2).

$$M = \begin{bmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,j} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,j} \\ \vdots & \vdots & \ddots & \vdots \\ m_{i,1} & m_{i,2} & \cdots & m_{i,j} \end{bmatrix} \quad (1)$$

$$m_{i,j} \in \{0, 1\} \quad (2)$$

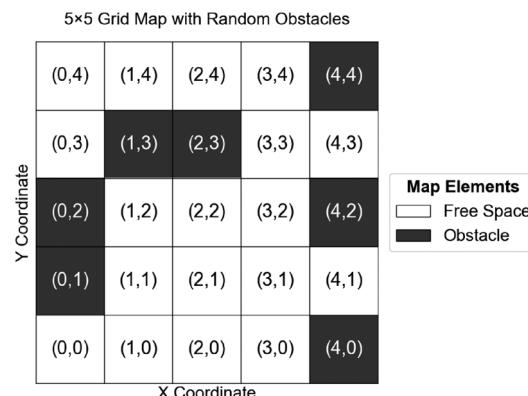


Figure 1. The 5x5 grid map.

Preliminary knowledge

A* algorithm

A* is a heuristic search algorithm, which introduces information related to the target point to improve the search efficiency, and it weighs the estimated cost and known cost to explore the path. The traditional A* algorithm uses Eq.(3) as the heuristic function to find the optimal path.

$$f(n) = h(n) + g(n) \quad (3)$$

where $h(n)$ is the heuristic cost from the current path point to the goal, and $g(n)$ represents the cost from the start to the current path point.

Heuristic function

There are three commonly used heuristic functions: Manhattan distance, Euclidean distance, and Chebyshev distance. The difference between the three heuristic functions is shown in Fig. 2.

The traditional A* algorithm supports four search directions, using Manhattan distance, which allows vertical and horizontal movement. As a result, it may overestimate the actual distance because of expanding to more nodes and overestimating the cost in the oblique direction. Manhattan distance between nodes $S(x_s, y_s)$ and $T(x_T, y_T)$ is calculated based on Eq.(4). It is suitable for simple scenarios, such as city blocks.

$$d(S, T) = |x_s - x_T| + |y_s - y_T| \quad (4)$$

The Euclidean distance calculates the straight-line distance between two nodes, and it is suitable for a barrier-free environment or continuous space. It supports moving in any direction, but it may result in lower calculation efficiency in large-scale scenarios due to the square root involved. It is calculated by Eq.(5).

$$d(S, T) = \sqrt{(x_s - x_T)^2 + (y_s - y_T)^2} \quad (5)$$

The Chebyshev distance allows horizontal, vertical, and diagonal movements for 8-neighbour searching. It supports free diagonal movement and is suitable for AGV path planning. It is implemented in this paper as the heuristic function, in Eq.(6).

$$d(S, T) = \max(|x_s - x_T|, |y_s - y_T|) \quad (6)$$

Obstacle density-based dynamic exponential A* Algorithm (ODDEA*)

For mobile robots, path planning faces the challenges of estimating the density and types of obstacles, and giving a reasonable path to pass through obstacles to reach the target smoothly²⁶. This paper proposes a new heuristic function based on the consideration of obstacle density and a dynamic exponential weight heuristic function to improve the A* algorithm, called the Obstacle Density-based Dynamic Exponential A* Algorithm (ODDEA*). The flowchart of the ODDEA* algorithm is shown in Fig. 3.

The proposed ODDEA* algorithm adjusts the weight of the heuristic function based on the obstacle density, and it uses the Chebyshev distance. The Chebyshev distance supports eight-directional movement on two-dimensional grid maps and efficiently estimates the movement cost from a node to the target, thus reducing the possibility of redundant search. It uses multi-variable fusion, adaptive weight adjustment, and local dynamic penalty, and $h(n)$ is determined by Eq. (7).

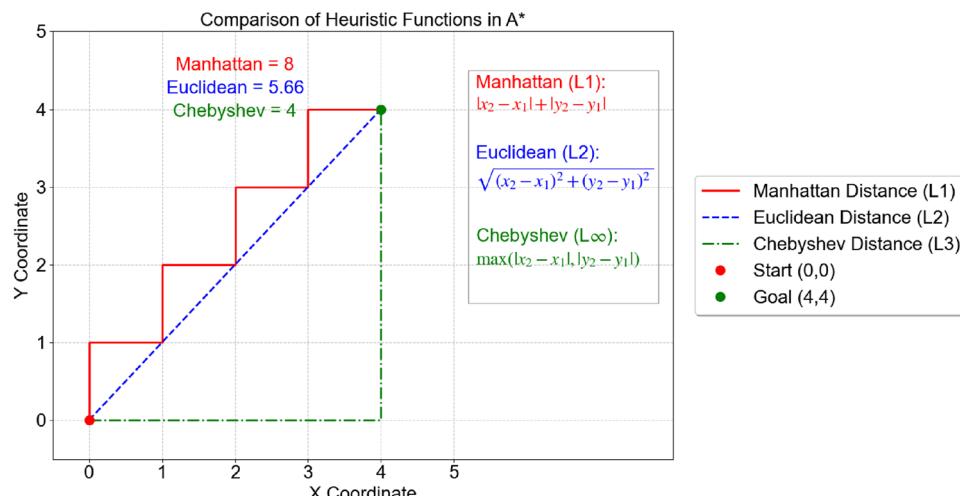


Figure 2. Three heuristic functions.

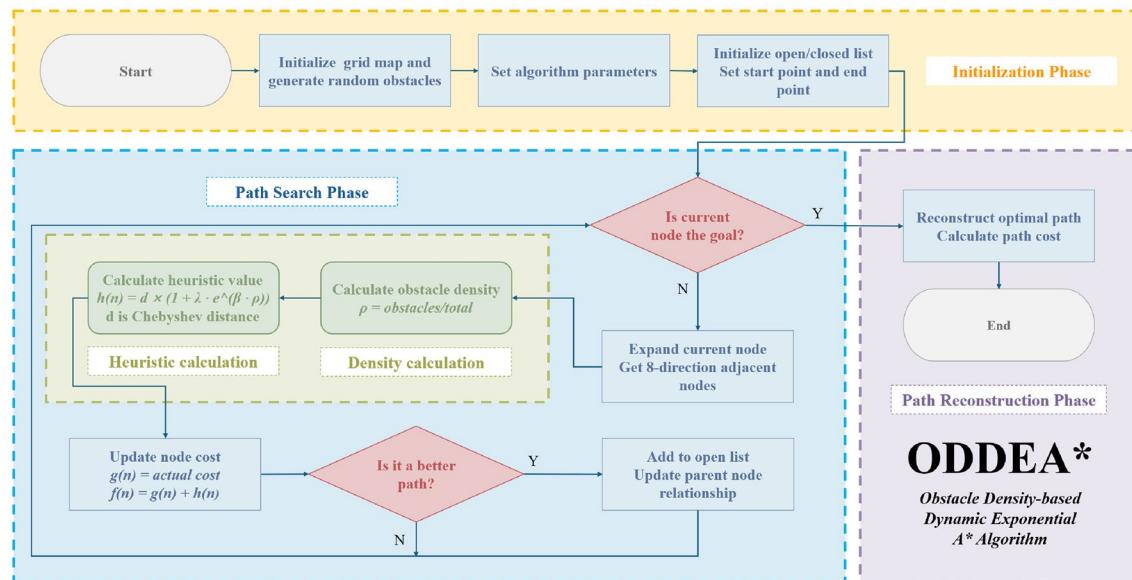


Figure 3. The flowchart of the ODDEA* algorithm.

$$h(n) = \max(|x_n - x_g|, |y_n - y_g|) \cdot (1 + \lambda \cdot e^{\beta \cdot D(n)}) \quad (7)$$

where $D(n)$ denotes the normalized obstacle density within a radius R around node n (with $D(n) \in [0, 1]$), λ represents the density-sensitive coefficient ($\lambda > 0$, controlling the strength of density influence), and β denotes the exponential decay coefficient ($\beta > 0$, indicating the nonlinearity degree of the density response).

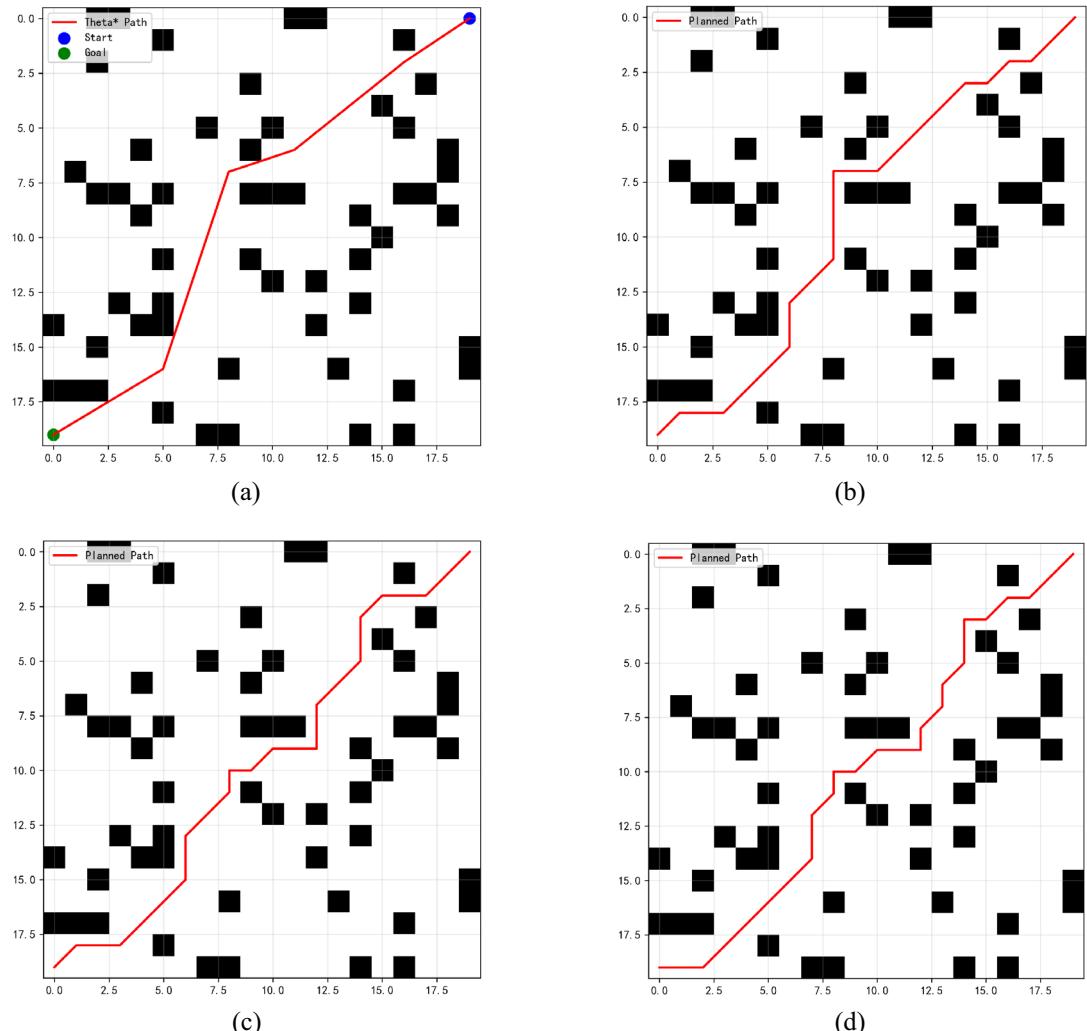
The ODDEA* algorithm dynamically adjusts the weight of the Chebyshev heuristic function based on the density of surrounding obstacles, as demonstrated in Algorithm 1. When the robot moves to an area with a high density of obstacles, $D(n)$ in its surrounding environment will increase significantly, as $D(n) \rightarrow 1$. Therefore, $e^{\beta \cdot D(n)}$ will increase significantly to guide the path planning process away from areas with dense obstacles. When the robot moves into an area with a low density of obstacles, $D(n) \rightarrow 0$. In this case, the heuristic function returns to the Chebyshev distance, thereby maintaining the efficiency of searching for the shortest path. The exponential function is used to achieve a nonlinear amplification of the density effect. Compared with the linear obstacle-density method, it addresses the problem of response hysteresis in high-density obstacle areas and insufficient risk differentiation in high-risk areas.

Input: start node X_{start} , end node X_{end} , OPEN, CLOSE
Output: Path = $[X_{\text{start}}, P_1, \dots, P_n, X_{\text{end}}]$

- 1: OPEN = $[X_{\text{start}}]$, where $f(X_{\text{start}}) = h(X_{\text{start}})$
- 2: **while** OPEN $\neq \emptyset$ **do**
- 3: Take node X_n from OPEN where $f(X_n) = \min f(X)$
- 4: **if** $X_n = X_{\text{end}}$ **then**
- 5: **break**
- 6: **end if**
- 7: Remove X_n from OPEN, add to CLOSE
- 8: X_i of $X_n \leftarrow$ child node selection method
- 9: **for** each child node X_i of X_n **do**
- 10: Update $g(X_i)$
- 11: Calculate $h(X_i) = \max(|x_i - x_g|, |y_i - y_g|) \cdot (1 + \lambda \cdot e^{\beta \cdot D(x_i)})$ ▷ Improved evaluation
- 12: Evaluate X_i with $f(X_i) = g(X_i) + h(X_i)$
- 13: Update OPEN list
- 14: **end for**
- 15: Update CLOSE list
- 16: **end while**
- 17: **return** PATH

Algorithm 1. ODDEA* Algorithm

Algorithm	Path length	Planning time (ms)	Number of expanded nodes	Search coverage
Theta* ²⁷	28.49	1.44	215	53.75%
A*	30.38	2.21	250	62.50%
BA* ²⁸	30.97	2.35	162	40.50%
ODDEA*	30.97	1.38	76	19.00%

Table 1. The small-size grid map (20×20).**Figure 4.** The simulation results of the small-size grid map (a) The Theta* algorithm²⁷. (b) The A* algorithm. (c) The BA* algorithm²⁸. (d) The ODDEA* algorithm.

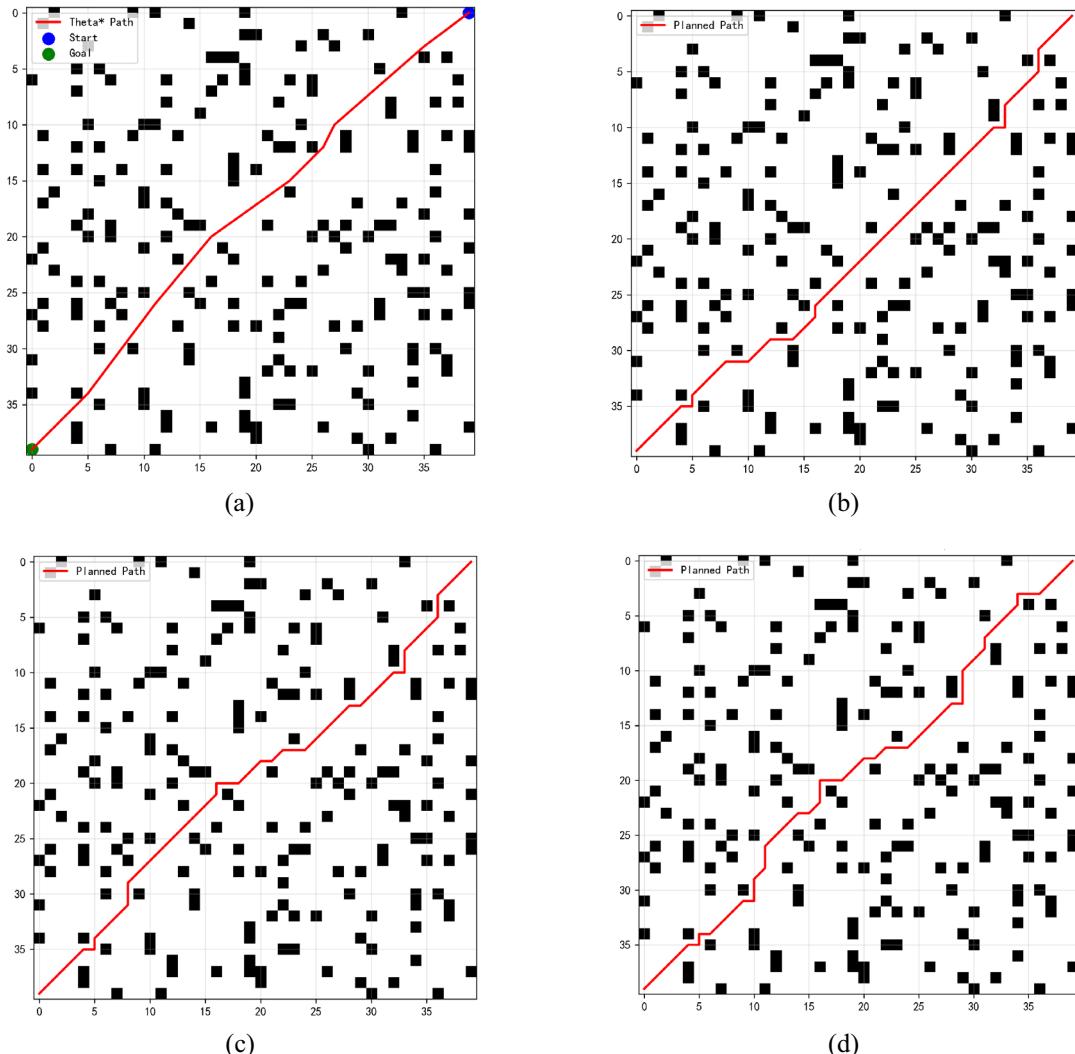
Computational experiments

The computational experiments utilize Python 3.11, running on an Intel i5-13600KF processor with 32GB of memory and a GeForce RTX 4070 graphics card. Three different scales of small, medium, and large two-dimensional grid maps are constructed in “The small-size map”, “The medium-size map”, “The large-size map” separately. The comparison analysis includes the Theta* algorithm²⁷, the A* algorithm, the Bidirectional A* algorithm²⁸, and the proposed ODDEA* algorithm. Each algorithm runs 20 times to obtain the average performance for comparison. Additionally, we generate 40x40 random maps 50 times for comparison. In the experiments, the proposed ODDEA* algorithm uses the parameters R is 2, λ is 0.5, and β is 2.0 for the small-size map, R is 2, λ is 0.6, and β is 3.0 for the medium-size map, and R is 2, λ is 0.7, and β is 3.0 for the large-size map.

The small-size map

The simulation results of different algorithms on a small-size grid map are shown in Table 1 and Fig. 4, and the best performance is highlighted in bold. Figure 4 presents the 5th path for each algorithm. For the planning

Algorithm	Path length	Planning time (ms)	Number of expanded nodes	Search coverage
Theta* ²⁷	55.59	7.00	995	62.19%
A*	58.67	12.18	1089	68.06%
BA* ²⁸	59.84	9.70	641	40.06%
ODDEA*	62.18	5.27	214	13.38%

Table 2. The medium-size grid map (40x40).**Figure 5.** The simulation results of the medium-size grid map (a) The Theta* algorithm²⁷. (b) The A* algorithm. (c) The BA* algorithm²⁸. (d) The ODDEA* algorithm.

time, the ODDEA* algorithm costs 62.44% of the A* algorithm. The Theta* algorithm produces the shortest path while requiring the second-least computation time. In terms of the number of expanded nodes, the traditional A* algorithm has a search coverage rate of 62.50% due to the invalid expansion in multiple directions in the search strategy of the heuristic function. In contrast, the ODDEA* algorithm based on the Chebyshev obstacle-density-weighted heuristic function expands only 76 search nodes on a 20×20 map, which is 30.40% of the traditional A* algorithm.

The medium-size map

Table 2 and Fig. 5 demonstrate the simulation data of the algorithms for comparison analysis on a medium-size map. For the path length, the Theta* is 55.59, the traditional A* algorithm is 58.67, the ODDEA* algorithm is 62.18, and the Bidirectional A* algorithm is 59.84. The search time of the traditional A* algorithm is 12.18ms, while the search time of the proposed ODDEA* algorithm is 5.27ms, which is 43.27% of the traditional A*

Algorithm	Path length	Planning time (ms)	Number of expanded nodes	Search coverage
Theta* ²⁷	84.91	16.10	2393	66.47%
A*	90.47	36.10	2880	80%
BA* ²⁸	90.47	24.15	1456	40.44%
ODDEA*	93.40	9.45	315	8.75%

Table 3. The large-size grid map (60 × 60).

algorithm. The planning time of the Theta* algorithm ranks second only to the ODDEA* algorithm, and the Theta* algorithm achieves the shortest path length.

For the number of expanded nodes, the traditional A* algorithm has 1089 expanded nodes, which is 68.06% of the entire map. However, the proposed ODDEA* algorithm has only 214 search nodes, which is only 19.65% of the traditional A* algorithm, with a coverage rate of only 13.38%. For the Bidirectional A* algorithm, the number of expanded nodes is significantly reduced compared to the traditional A*, which is 58.86% of the traditional A*.

The large-size map

In terms of a large-size map, the path length is 90.47 for the traditional A* algorithm, 93.40 for the ODDEA* algorithm, 90.47 for the Bidirectional A* algorithm, and 84.91 for the Theta* algorithm. The planning time of the traditional A* algorithm is 36.10ms, while the ODDEA* algorithm takes 9.45ms, which is only 26.18% of the traditional A* algorithm and 58.70% of the Theta* algorithm. The Bidirectional A* algorithm takes 27.08ms, which is 75.01% of the traditional A* algorithm.

Considering the number of nodes to be expanded, the traditional A* algorithm's number of expanded nodes reaches 2880, and the search coverage rate is 80.00% of the map, which consumes a lot of search space. The number of nodes expanded by the ODDEA* algorithm is only 315, and the search coverage rate reaches 8.75%, which is only 10.94% of the search coverage rate of the traditional A* algorithm. The Bidirectional A* algorithm's coverage rate is also only 40.44%. The simulation results are indicated in Table 3 and Fig. 6.

Random maps

Each algorithm was used to generate 50 random maps on a medium-size grid map, and the average values of path length, planning time, expanded nodes, and search coverage rate were calculated, as shown in Table 4. Each algorithm has the same starting position (39, 0) and end point (0, 39), and uses the same heuristic function based on Chebyshev distance. The generated paths by each algorithm in the 20th and 40th maps are shown in Figs. 7, 8, 9 and 10.

From the 50 random maps, the average path planning time for the proposed ODDEA* algorithm is 5.10ms, which is 41.26% of the A* algorithm, and it is the minimum planning time among the five algorithms. The average number of expanded nodes is 215.62, which is only 18.29% of the A* algorithm, and the coverage rate is only 13.48% of the map. The Theta* algorithm achieves the shortest path with 62.95% search coverage.

Conclusion

The proposed ODDEA* algorithm dynamically adjusts the heuristic function based on obstacle density, and it guides the search direction to avoid invalid expansion. It can reduce the possibility of excessive redundant node expansion and has faster search ability than the traditional A* algorithm. The planning time of the ODDEA* algorithm is 53.87%, 55.54%, and 31.47% of the traditional A* algorithm in the small-size (20 × 20), medium-size (40 × 40), and large-size (60 × 60) grid maps. For the search space, the ODDEA* algorithm only requires 30.40%, 19.65%, and 10.94% of the traditional A* algorithm in three different maps. For the 50 random maps, the average path planning time of the ODDEA* algorithm is 5.10ms, the average number of expanded nodes is 215.62, and the coverage rate is only 13.48%. Compared to the other algorithms (A*, BA*, Theta*) in the experiments, the ODDEA* algorithm has better performance in search ability.

However, the proposed ODDEA* may result in a longer path during planning, and its parameters need to be adjusted based on the environmental settings. In future work, reinforcement learning will be considered to enable the agent to choose the direction and strategy of movement through the perception of the surrounding environment and the current motion state of the robot, while the reward mechanism will be used to guide the robot's behavior. By combining the global search capability of the improved A* algorithm with the dynamic adaptability of the reinforcement learning model, the neural network can be trained to achieve fast convergence in complex environments.

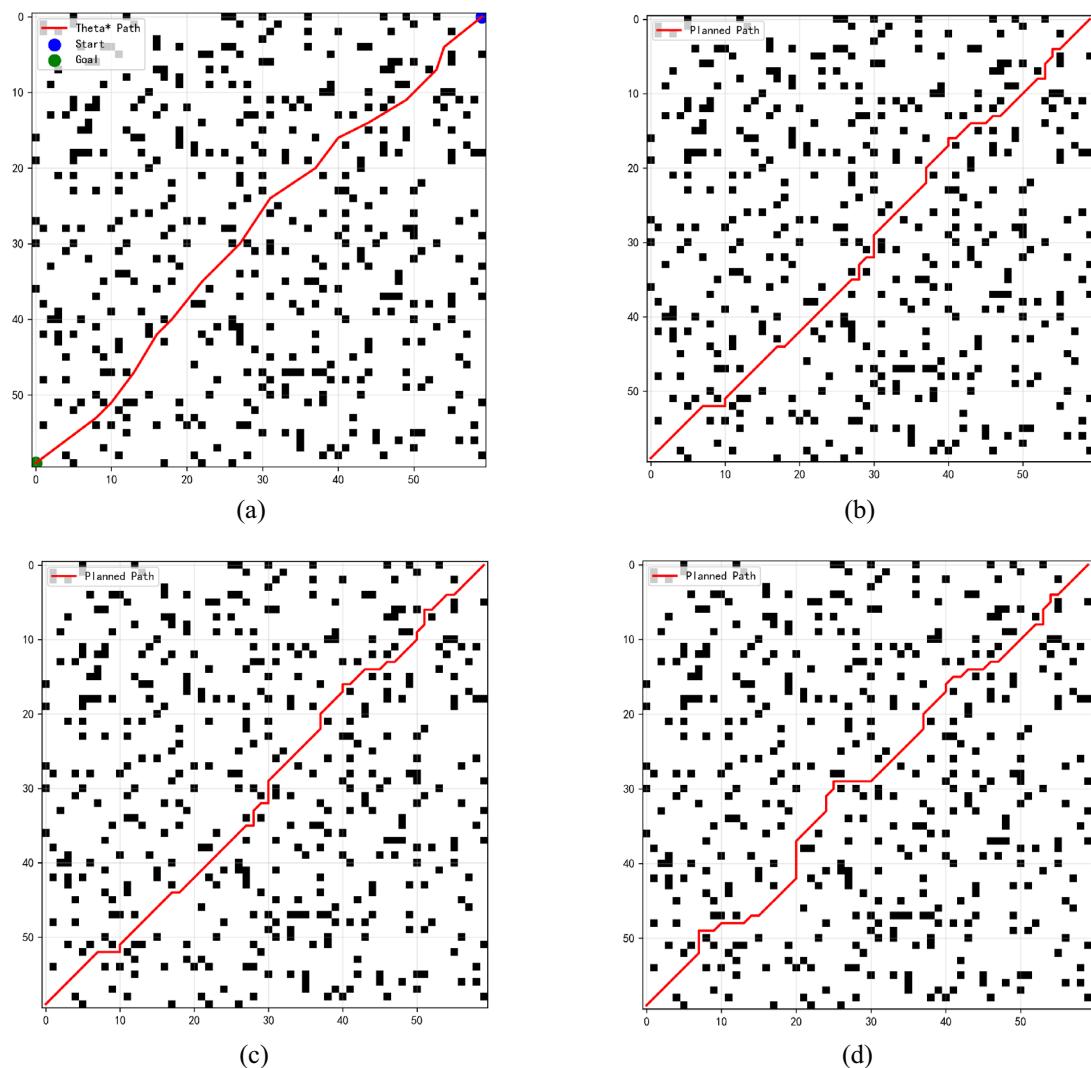


Figure 6. The simulation results of the large-size grid map (a) The Theta* algorithm ²⁷. (b) The A* algorithm. (c) The BA* algorithm ²⁸. (d) The ODDEA* algorithm.

Algorithm	Avg path length	Avg planning time (ms)	Avg number of expanded nodes	Avg search coverage
Theta* ²⁷	56.71	6.65	1007.14	62.95%
A*	60.40	12.36	1178.94	73.68%
BA* ²⁸	61.01	8.74	610.64	38.15%
ODDEA*	62.64	5.10	215.62	13.48%

Table 4. 40 × 40 random maps.

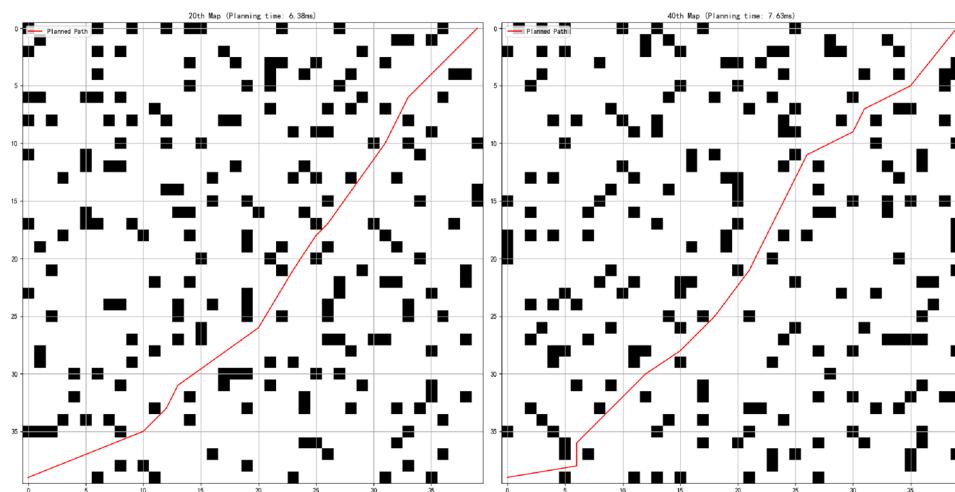


Figure 7. The path generated by the Theta* algorithm in the 20th and 40th map.

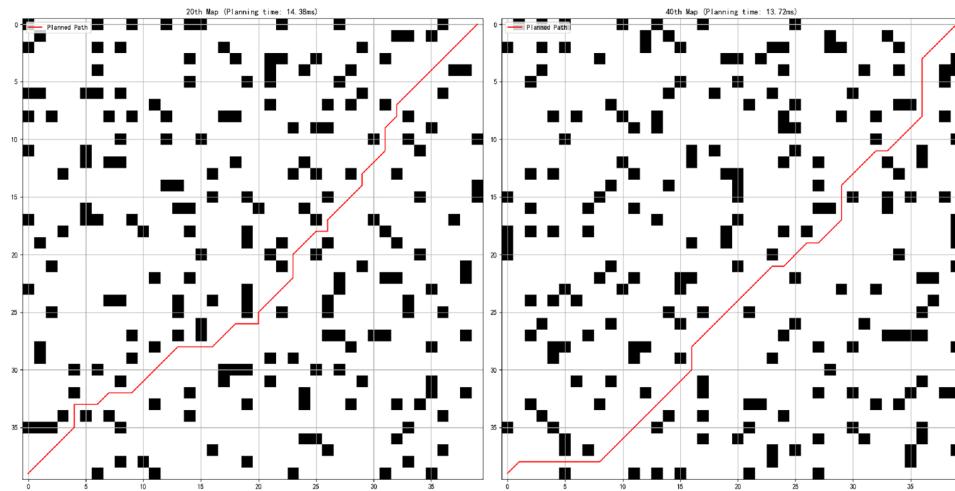


Figure 8. The path generated by the A* algorithm in the 20th and 40th map.

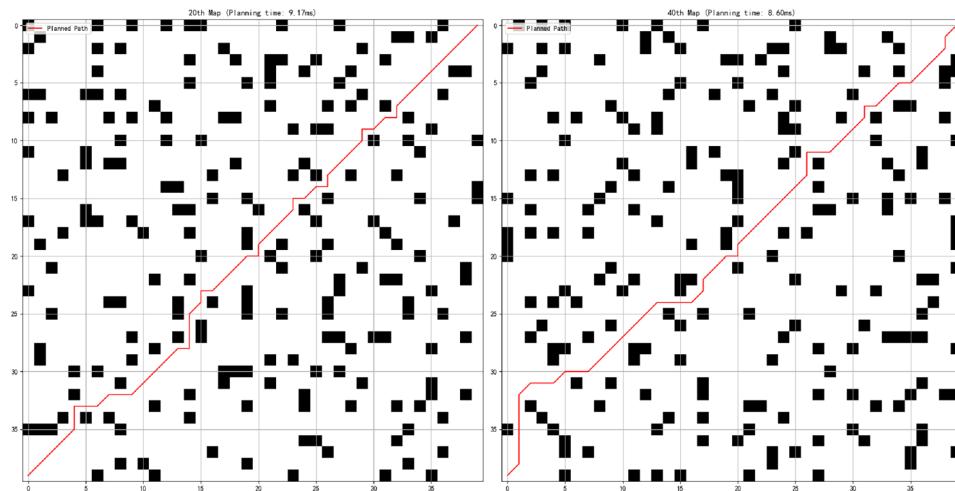


Figure 9. The path generated by the BA* algorithm in the 20th and 40th map.

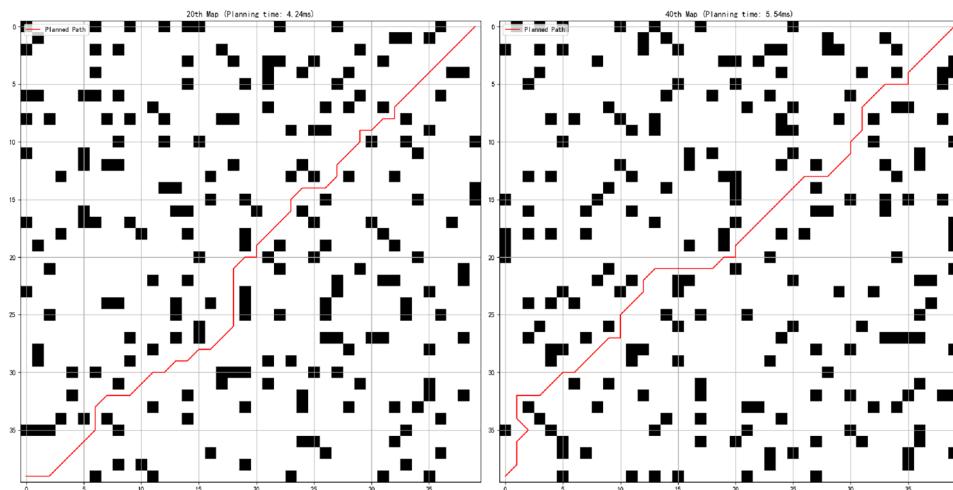


Figure 10. The path generated by the ODDEA* algorithm in the 20th and 40th map.

Data availability

Data is provided within the manuscript.

Received: 6 August 2025; Accepted: 12 November 2025

Published online: 29 December 2025

References

1. Liu, L., Wang, B. & Xu, H. Research on path-planning algorithm integrating optimization a-star algorithm and artificial potential field method. *Electronics* **11**, 3660 (2022).
2. Lin, S., Liu, A., Wang, J. & Kong, X. A review of path-planning approaches for multiple mobile robots. *Machines* **10**, 773 (2022).
3. Tang, G., Tang, C., Claramunt, C., Hu, X. & Zhou, P. Geometric a-star algorithm: An improved a-star algorithm for agv path planning in a port environment. *IEEE Access* **9**, 59196–59210 (2021).
4. Zhang, Z., Wang, S. & Zhou, J. A-star algorithm for expanding the number of search directions in path planning. In *2021 2nd International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT)*, 208–211 (IEEE, 2021).
5. Liu, H. & Zhang, Y. Asl-dwa: An improved a-star algorithm for indoor cleaning robots. *Ieee Access* **10**, 99498–99515 (2022).
6. Han, C. & Li, B. Mobile robot path planning based on improved a* algorithm. In *2023 IEEE 11th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, vol. 11, 672–676, <https://doi.org/10.1109/ITAIC58329.2023.10408799> (2023).
7. Ju, C., Luo, Q. & Yan, X. Path planning using an improved a-star algorithm. In *2020 11th international conference on prognostics and system health management (PHM-2020 Jinan)*, 23–26 (IEEE, 2020).
8. Martins, O. O., Adekunle, A. A., Olaniyan, O. M. & Bolaji, B. O. An improved multi-objective a-star algorithm for path planning in a large workspace: Design, implementation, and evaluation. *Sci. Afr.* **15**, e01068 (2022).
9. Liao, T. et al. Research on path planning with the integration of adaptive a-star algorithm and improved dynamic window approach. *Electronics* **13**, 455 (2024).
10. He, Z., Liu, C., Chu, X., Negenborn, R. R. & Wu, Q. Dynamic anti-collision a-star algorithm for multi-ship encounter situations. *Appl. Ocean Res.* **118**, 102995 (2022).
11. Li, J., Liao, C., Zhang, W., Fu, H. & Fu, S. Uav path planning model based on r5dos model improved a-star algorithm. *Appl. Sci.* **12**, 11338 (2022).
12. XiangRong, T., Yukun, Z. & XinXin, J. Improved a-star algorithm for robot path planning in static environment. *J. Phys. Conf. Ser.* **1792**, 012067 (2021) (IOP Publishing).
13. Li, C., Huang, X., Ding, J., Song, K. & Lu, S. Global path planning based on a bidirectional alternating search a* algorithm for mobile robots. *Comput. Ind. Eng.* **168**, 108123. <https://doi.org/10.1016/j.cie.2022.108123> (2022).
14. Zhang, D., Chen, C. & Zhang, G. Agv path planning based on improved a-star algorithm. In *2024 IEEE 7th Advanced Information Technology, Electron. Autom. Control Conf. (IAEAC)* **7**, 1590–1595 (2024) (IEEE).
15. Meng, T. et al. Improved hybrid a-star algorithm for path planning in autonomous parking system based on multi-stage dynamic optimization. *Int. J. Automot. Technol.* **24**, 459–468. <https://doi.org/10.1007/s12239-023-0038-1> (2023).
16. Lin, S., Liu, A., Wang, J. & Kong, X. An intelligence-based hybrid pso-sa for mobile robot path planning in warehouse. *J. Comput. Sci.* **67**, 101938. <https://doi.org/10.1016/j.jocs.2022.101938> (2023).
17. Sathiya, V., Chinnadurai, M. & Ramabalan, S. Mobile robot path planning using fuzzy enhanced improved multi-objective particle swarm optimization (fimopso). *Expert Syst. Appl.* **198**, 116875. <https://doi.org/10.1016/j.eswa.2022.116875> (2022).
18. Hou, W., Xiong, Z., Wang, C. & Chen, H. Enhanced ant colony algorithm with communication mechanism for mobile robot path planning. *Robot. Auton. Syst.* **148**, 103949. <https://doi.org/10.1016/j.robot.2021.103949> (2022).
19. Wu, L., Huang, X., Cui, J., Liu, C. & Xiao, W. Modified adaptive ant colony optimization algorithm and its application for solving path planning of mobile robot. *Expert Syst. Appl.* **215**, 119410. <https://doi.org/10.1016/j.eswa.2022.119410> (2023).
20. Ding, J. et al. An improved rrt* algorithm for robot path planning based on path expansion heuristic sampling. *J. Comput. Sci.* **67**, 101937. <https://doi.org/10.1016/j.jocs.2022.101937> (2023).
21. Wang, J., Li, B. & Meng, M. Q. H. Kinematic constrained bi-directional rrt with efficient branch pruning for robot path planning. *Expert Syst. Appl.* **170**, 114541. <https://doi.org/10.1016/j.eswa.2020.114541> (2021).
22. Orozco-Rosas, U., Picos, K., Pantrigo, J. J., Montemayor, A. S. & Cuesta-Infante, A. Mobile robot path planning using a qapf learning algorithm for known and unknown environments. *IEEE Access* **10**, 84648–84663. <https://doi.org/10.1109/ACCESS.2022.3197628> (2022).

23. Pei, M., An, H., Liu, B. & Wang, C. An improved dyna-q algorithm for mobile robot path planning in unknown dynamic environment. *IEEE Trans. Syst. Man Cybern. Syst.* **52**, 4415–4425. <https://doi.org/10.1109/TSMC.2021.3096935> (2022).
24. Gong, H., Wang, P., Ni, C. & Cheng, N. Efficient path planning for mobile robot based on deep deterministic policy gradient. *Sensors* **22**, 3579 (2022).
25. Chen, L. et al. Transformer-based imitative reinforcement learning for multirobot path planning. *IEEE Trans. Industr. Inf.* **19**, 10233–10243. <https://doi.org/10.1109/TII.2023.3240585> (2023).
26. Tang, Y., Zakaria, M. A. & Younas, M. Path planning trends for autonomous mobile robot navigation: A review. *Sensors* **25**, 1206 (2025).
27. Daniel, K., Nash, A., Koenig, S. & Felner, A. Theta^{*}: Any-angle path planning on grids. *J. Artif. Intell. Res.* **39**, 533–579 (2010).
28. Sun, Y., Yuan, Q., Gao, Q. & Xu, L. A multiple environment available path planning based on an improved a^{*} algorithm. *Int. J. Comput. Intell. Syst.* **17**, 172 (2024).

Author contributions

Conceptualization, S.L. and X.F.; methodology, S.L. and X.F.; software, X.F.; validation, X.F.; formal analysis, X.F.; investigation, S.L. and X.F.; resources, S.L. and X.F.; data curation, X.F.; writing—original draft preparation, S.L., X.F. and Z.X.; writing—review and editing, S.L., X.F., Z.X. and Y.W.; visualization, S.L., X.F. and Z.X.; supervision, S.L., Y.W. and H.Y.; project administration, S.L., Y.W. and H.Y.; funding acquisition, S.L. and H.Y.

Funding

This work was supported in part by the Natural Science Foundation of Fujian Province, China, under Grant 2024J01723, Grant 2024J01721 and Grant 2024J01115; in part by the Key Project of Natural Science Foundation of Fujian Province, China, under Grant 2022J02045; and in part by the Startup Fund of Jimei University under Grant ZQ2024002 and Grant ZQ2024034.

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to S.L.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025