Journal of Engineering, Project, and Production Management
2026, 16(4), 2025-213

Dynamic Route Planning for Warehouse Robots Combining A-Star and Dynamic Window Approach

Jianjun Li[1] and Ji Xiang[2]
[1] Engineer, Alibaba Cainiao Network Company, China, E-mail: lijianjun_1983@126.com (corresponding author).
[2] Professor, College of Electrical Engineering, Zhejiang University, Hangzhou,310027, China, E-mail: jxiang@zju.edu.cn

Production Management
Received September 23, 2025; revised December 25, 2025; accepted December 25, 2025
Available online May 29, 2026

Abstract: Dynamic route planning for warehouse robots plays an important role in the development of logistics and warehousing. Accurate planning of robot routes can reduce labor costs and decrease energy consumption. However, traditional methods for warehouse robot route planning suffer from low efficiency, route deviations, and inaccurate navigation. This study utilizes the Improved Sparrow Search Algorithm (ISSA) to optimize route planning parameters and fill in missing critical information during the search process. It also combines Multi-Region Biased Sampling and a Key-Node Guidance algorithm (MRBS-KNG-A*) to highlight important areas, ultimately constructing a dynamic route planning model for warehouse robots. Experimental results show that the improved algorithm achieves a route length of 22.5m, a search time of 4.9s, and three turns in high-obstacle environments. These results outperform relative strategy optimization, Fourier Ring Correlation Resolution (FRC), and chaotic Particle Swarm Optimization (PSO) algorithms. Meanwhile, the constructed model significantly reduces the number of branch nodes, achieving 11 turns and a pathfinding success rate of 82.95%. Under the cooperation of multiple robots, the collision times, maximum curvature, iterative optimization time, and actual energy consumption of the research model for moving obstacles are 2 times, 0.72 m^{-1}, 25.69 s, and 2.62 kW/h, respectively. The results indicate that the improved dynamic route planning model enhances both planning efficiency and accuracy. This study contributes to the precise construction of multi-destination route planning models, thereby improving the mobility efficiency of the storage robot.

Keywords: A*, algorithm, dynamic window approach, warehouse robot, route planning, improved sparrow search algorithm.

Copyright © Journal of Engineering, Project, and Production Management (EPPM-Journal).
DOI 10.32738/JEPPM-2025-213

1. Background

With the rapid development of intelligent manufacturing, logistics and warehousing systems are quickly moving toward automation. Warehouse robots play a crucial role in the logistics system by significantly reducing labor costs and minimizing workforce waste (Kappagantula and Mannayee, 2024). However, the complex and changing warehouse environment, including personnel movements and equipment shifts, affects dynamic route planning for robots, thereby increasing energy consumption and time loss (Sahoo and Choudhury, 2023). Therefore, effective dynamic route planning in complex warehouse environments is highly significant. Currently, traditional methods for warehouse robot routes target static environments and cannot respond to dynamic conditions. This causes route deviations and limits global route-planning capabilities. Hence, there is an urgent need for methods that can plan dynamic routes for warehouse robots with high accuracy and efficiency (Pattyn, 2025). With rapid advances in computer technology, intelligent algorithm-based dynamic route models for robots have become a research hotspot in recent years. Intelligent algorithms offer advantages such as high efficiency, strong objectivity, and reduced reliance on manual intervention. Among them, the A-star (A*) algorithm finds the optimal route by weighting heuristic information, while the Dynamic Window Approach (DWA) algorithm generates windows to evaluate its trajectory safety and avoid obstacles (Chandra and Istiono, 2022; Lai et al., 2023). This study combines an ISSA to fill in missing key node information during search, and incorporates regional bias sampling and key node guidance algorithms to reduce route nodes and highlight critical regions. Ultimately, a dynamic route planning model for warehouse robots based on the improved hybrid of A* and DWA algorithms is constructed. This study aims to enable an efficient global dynamic route planning for warehouse robots, improve accuracy, and support the sustainable development of intelligent manufacturing. The innovation of this research lies in combining the A*-DWA and ISSA algorithms. Compared to the SSA algorithm, the improved ISSA algorithm demonstrates superior performance. It reduces collision times by 5, maximum curvature by 0.26m^{-1}, iterative optimization time by 4.05s, and actual energy

consumption by0.67kW/h, thereby significantly enhancing pathfinding efficiency. This study presents a novel approach to related research by developing an enhanced A*-DWA-based dynamic route-planning model for warehouse robots.

2. Literature Review

The hybrid A*-DWA algorithm combines two artificial intelligence algorithms, enabling it to solve complex problems across various fields. Many scholars have conducted in-depth research on it. Among them, Wang et al. (2024) proposed a low-energy multi-route planning algorithm for automated container terminal supply chains. They developed a multi-iteration route-planning model based on a collision-free scheme, which was constructed to optimize iterative pathfinding. Lin's team developed an A* route planning algorithm for unmanned vehicles. They improved the A* algorithm by enhancing redundant safety space and introduced prediction planning and corner elimination strategies to obtain the optimal driving route. Safety corridor technology was utilized to eliminate repetitive calculations and expedite the process (Lin et al., 2023). Yudha et al. (2022) proposed a route-planning model based on the A* algorithm to find the best tourist routes. They calculated the route distance and density in West Java Province, transformed the density data using heuristic values, and ultimately obtained the optimal tourism route. Li et al. (2025) improved route angles by node safety expansion and introduced a heuristic function to calculate the distance between nodes, thereby increasing the efficiency of the A* algorithm. They utilized the Floyd algorithm to smooth routes and significantly adjusted weights to enhance the DWA algorithm, thereby optimizing route planning by combining local target point selection. Vikas and Parhi (2023) proposed an improved DWA for route planning in static and dynamic terrains. They introduced a hyperbolic gravitational search algorithm to enhance the exploration and development speed of DWA. The hyperbolic gravitational search algorithm's parameters were modified using a chaotic search to improve computational speed and planning cost.

Dynamic route planning for warehouse robots plays a crucial role in logistics and warehousing systems. At present, the theories and applications of route planning methods are relatively mature, with numerous scholars having conducted thorough research. Ishihara et al. (2022) developed a model-predictive control-based route planning model for automated transport robots to improve warehouse efficiency. They optimized model predictive control route planning parameters by using Dijkstra's algorithm and optimized warehouse map information to achieve global route planning. Rasheed et al. (2022) employed rapidly exploring random trees to generate candidate nodes without collisions and optimized these nodes using particle swarm optimization. They calculated the robot's speed using reference route equations and then found the shortest collision-free route. Jathunga and Rajapaksha (2025) proposed a hybrid algorithm that combines a probabilistic roadmap and a genetic algorithm for multi-robot route planning in dynamic environments. They optimized obstacles and navigation efficiency through data collection and processing, improving route length and turn counts in complex multi-robot scenarios. Sandanika et al. (2024) realized task coordination distribution through multi-robot fusion. They divided maps into regions using grid maps, assigned tasks adaptively, and reduced redundancy by using hybrid route planning methods. They finally developed a multi-robot route-planning model with full-area coverage. Wu et al. (2023) addressed travel conflicts of warehouse robots by proposing a two-level vehicle-robot route planning model. They utilized the ACO algorithm to mitigate congestion in static environments and implemented a priority mechanism to dynamically adjust routes in real-time, resulting in optimal route planning outcomes.

In summary, existing research has made significant progress in route planning for warehouse robots. However, limitations remain, such as weak obstacle avoidance and limited global planning capabilities. Therefore, this study introduced ISSA to optimize route-planning parameters and to fill in missing key information during search. It combined regional bias sampling and key node guidance algorithms to address problems of low sampling and navigation efficiency. Ultimately, a dynamic route planning model for warehouse robots based on an improved hybrid of A* and DWA was constructed. The model aims to plan routes efficiently, reduce route redundancy and turn counts, meeting the needs of intelligent manufacturing for robot route planning.

3. Dynamic Route Planning Strategy for Warehouse Robots Combining A* and DWA

3.1. Design of Improved A*-DWA Algorithm with ISSA Integration

With the widespread application of warehouse robots, intelligent computer algorithms play an important role in dynamic route planning for warehouse robots (Li et al., 2024). Although traditional dynamic route planning techniques still have limitations, such as low navigation efficiency and inaccurate positioning. To address these issues, this study proposes an A*-DWA hybrid algorithm applied to dynamic route planning for warehouse robots, further improving obstacle avoidance ability and reducing turn counts in complex environments (Sarbini et al., 2024). The proposed A*-DWA algorithm efficiently plans routes, reduces route redundancy and excessive turns, features strong adaptability and has high exploration capability. The operation flow of the A*-DWA algorithm.

As shown in Fig. 1, the A*-DWA algorithm consists of two parts. The first part is the A* algorithm. After initialization, the map is selected for expansion. The expanded points are divided into two lists: Open and Close. The start node and its neighboring nodes are placed into the Open list. The node with the smallest cost in the Open list is moved to the Close list, and this process repeats. The second part is the DWA algorithm. It updates the robot's position and velocity to determine its velocity space, calculates the dynamic window evaluation function, and sets the target orientation, obstacle distance, speed, and path following as adaptive adjustment weight coefficients. It then outputs the normalized target values.

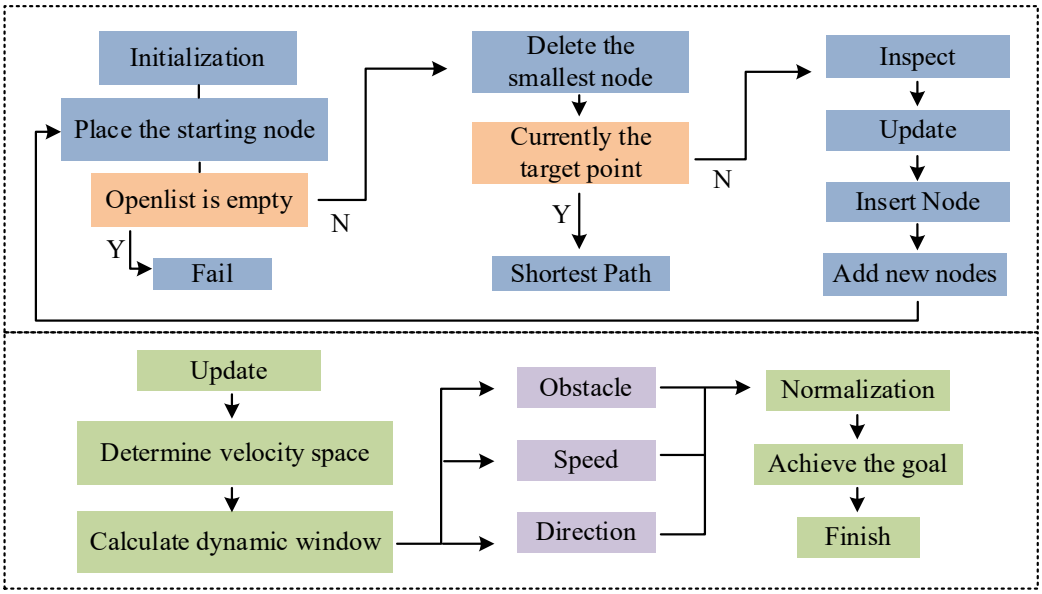


Fig. 1. Operational flowchart of the A*-DWA algorithm

The evaluation function in the A* algorithm is mathematically expressed in Eq. (1).

$$f(n) = g(n) + h(n) \tag{1}$$

In Eq. (1), $f(n)$ represents the cost function, $g(n)$ is the actual cost function, $h(n)$ is the heuristic cost function, and n represents the current node. The Euclidean distance between two nodes is calculated as shown in Eq. (2).

$$d[(x_1, y_1), (x_2, y_2)] = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{2}$$

In Eq. (2), d is the node distance, x is the x-coordinate, and y is the y-coordinate. The velocity space determination in the DWA algorithm is calculated as shown in Eq. (3).

$$V_s = \left\{(v, w) \middle| v \in [v_{\min}, v_{\max}] \wedge w \in [w_{\min}, w_{\max}]\right\} \tag{3}$$

Eq. (3), v and w represent linear velocity and angular velocity, respectively. $v_{\min}$ and $v_{\max}$ are the minimum and maximum linear velocities, where $w_{\min}$ and $w_{\max}$ are the minimum and maximum angular velocities. The trajectory is generated within the window, and its cost function is expressed in Eq. (4).

$$J(v, w) = w_{goal} \cdot J_{goal}(v, w) + w_{obs} \cdot J_{obs}(v, w) + w_{vel} \cdot J_{vel}(v, w) \tag{4}$$

In Eq. (4), J is the cost function, w_{goal} is the attraction weight, J_{goal} is the attraction cost, w_{obs} is the obstacle weight, J_{obs} is the obstacle cost, w_{vel} is the velocity weight, and J_{vel} is the velocity cost. The A*-DWA algorithm achieves precise control of the robot by adjusting weights in the function. However, in complex environments, it may lead to the omission of critical information in the search and fail to achieve global optimization. The sparrow search algorithm simulates the behavior of sparrows for cooperative global search (Hidayat et al., 2023). However, the basic algorithm is prone to falling into local optima in later iterations, and the decline in population diversity leads to insufficient search accuracy and slow convergence speed (Xue et al., 2024). Therefore, Cauchy reverse learning, sine cosine algorithm and Lévy flight strategy are introduced to improve the sparrow search algorithm for SSA. The ISSA process is shown in Fig. 2.

As shown in Fig. 2, ISSA first initializes the sparrow population using Cauchy opposition-based learning, calculates fitness values, and obtains the best and worst individuals in the population. The population is divided into leaders, followers, and sentinels. Then, the leader positions are updated using the sine-cosine algorithm. The follower positions are updated using the Lévy flight strategy, enabling leaders and followers to compete. Finally, the sentinel positions are updated to obtain individual and global optimal values. The process checks if the maximum number of iterations is reached; if not, it recalculates fitness values. If yes, it outputs the optimal value. The expression for initialization using Cauchy opposition-based learning is shown in Eq. (5).

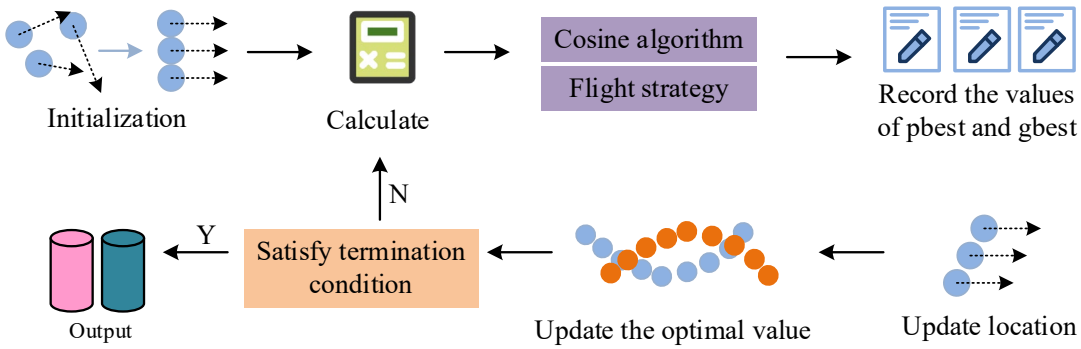


Fig. 2. Flowchart of ISSA (Icon source from: https://iconpark.oceanengine.com/)

$$OP_c = rand(\frac{a_i + b_i}{2}, x_i) \tag{5}$$

In Eq. (5), OP_c represents the opposition point matrix, $rand$ is a random number in [0,1], a_i and b_i represents two matrices, and x_i is the position of the i-th sparrow. The initialization divides sparrows with good fitness into a category. The sine-cosine algorithm then improves the algorithm's adaptive weight. The mathematical expression for the adaptive weight factor is shown in Eq. (6).

$$\lambda = \lambda_{\min} + \frac{(\lambda_{\max} - \lambda_{\min})}{(\lambda_{\max} + \lambda_{\min})} \cdot \cos(t \cdot \pi / iter_{\max}) \tag{6}$$

Eq. (6) is the weight factor, $\lambda_{\max}$ is the maximum weight, $\lambda_{\min}$ is the minimum weight, t is the current iteration, and $iter_{\max}$ is the maximum iteration. The leader position is updated by the sine-cosine factor and adaptive weight. The Lévy flight strategy is used to enhance local search ability. The Lévy flight strategy equation is shown in Eq. (7).

$$F(x) = 0.01 \cdot \frac{r_1 \cdot \sigma}{\sqrt[\varsigma]{|r_2|}} \tag{7}$$

Eq. (7) shows that F represents Lévy flight, r_1 and r_2 are discrete values in [0,1], and ς is a constant. The mathematical expression for determining the sparrow population position is shown in Eq. (8).

$$X = [c_1, c_2, c_3, ..., c_d]^T, c_i = [c_{i1}, c_{i2}, c_{i3}, ..., c_{id}] \tag{8}$$

As shown in Eq. (8), X represents position information, c is a variable, and d is the space dimension. Finally, ISSA is introduced to hybridize the A* and DWA algorithms, yielding the ISSA-A*-DWA algorithm. The ISSA-A*-DWA workflow is shown in Fig. 3.

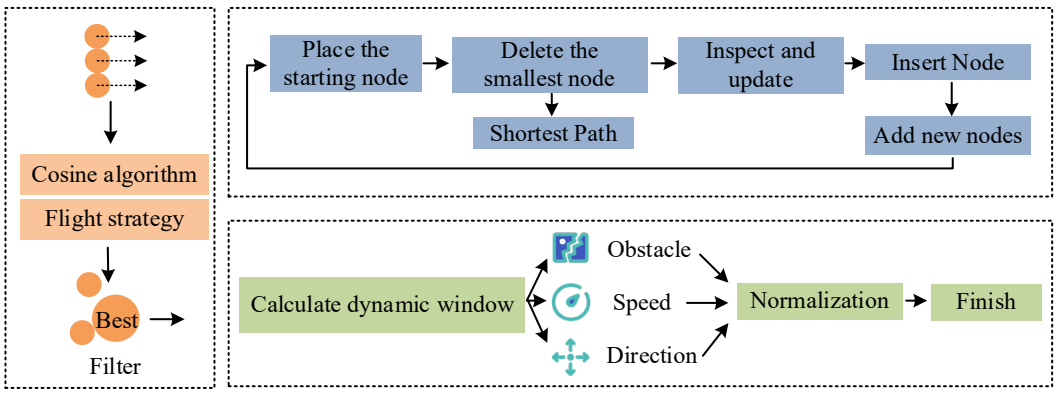


Fig. 3. Flowchart of ISSA-A*-DWA algorithm (Icon source from: https://iconpark.oceanengine.com/)

In Fig. 3, the ISSA-A*-DWA algorithm uses the A* algorithm to process map data and generate a dynamic obstacle-avoidance path for the DWA algorithm during standard navigation. The robot is surrounded by dynamic obstacles and gets stuck. The model will trigger the ISSA algorithm to optimize path parameters and eliminate limitations. The first part, ISSA, initializes population positions using Cauchy opposition-based learning, updates leader positions with the sine-cosine algorithm, and updates follower positions using the Lévy flight strategy, thereby obtaining optimal position information. The second part is the A* algorithm, which expands the map, selects the minimum route nodes in the open list, moves them to the closed list for looping, and optimizes the evaluation function. The last part is DWA, the global path generated by ISSA is disassembled into a sequence of continuous sub-targets, which are used as the local navigation target of DWA. The

prospective point closest to the robot on the path is selected as the temporary target in real time, and the direction item of the DWA track evaluation function is embedded to guide the robot to move along the global direction. This updates the position, integrates global and local optima, calculates the velocity space, dynamically adjusts the sampling space through the window evaluation function, and outputs target values.

3.2. Construction of Dynamic Route Planning Model for Warehouse Robots

ISSA-A*-DWA enhances the adaptability of robot route planning but is prone to falling into local optima. This leads to excessive global guiding route nodes and certain limitations (Rizki et al., 2025). The algorithm samples regions too evenly and does not emphasize sampling near the start and end points or around obstacles, resulting in low efficiency (Shi et al., 2023). Therefore, this study combines Multi-Region Biased Sampling (MRBS) and Key-Node Guidance (KNG) to improve navigation efficiency and constructs a warehouse robot route planning model, named MRBS-KNG-A*-DWA. MRBS classifies sampling probabilities of map areas, improving route quality and generation speed. The sampling probability distribution is shown in Eq. (9).

$$P_{start} + P_{end} + P_{obs} + P_{norm} = 1 \qquad (9)$$

Eq. (9), P_{start}, P_{end}, P_{obs}, and P_{norm} represent the spatial sampling probabilities for the start point, end point, obstacles, and important regions, respectively. The detailed process of MRBS is shown in Fig. 4.

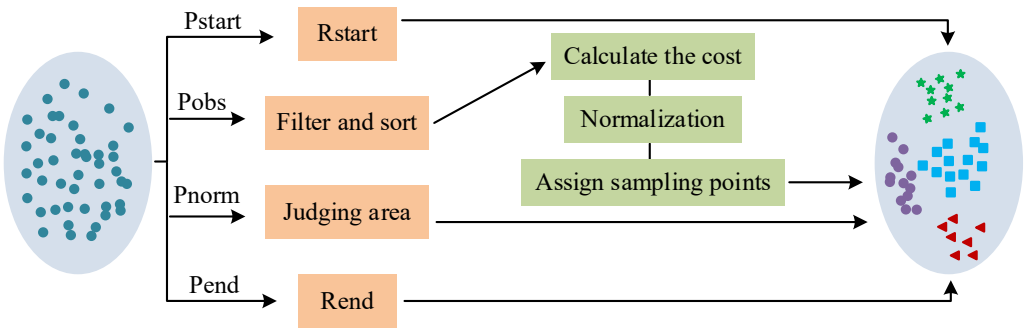


Fig. 4. MRBS operation process diagram

As shown in Figure 4, MRBS divides the map space into start-point, obstacle, important-region, and end-point areas based on a probability distribution. The start point area calculates the sampling radius by planning distance and iteration count to reduce computation and storage space. In obstacle space sampling, obstacles are inflated, and 3D geometry is used to calculate avoidance routes. Normalization is applied to obtain obstacle cost distribution. Important regions are handled by deciding whether to apply biased sampling for route planning. The endpoint space sampling calculates the sampling radius and density, which accelerates route generation. The equation for calculating the start point, stop sampling radius is shown in Eq. (10).

$$R_{start} = \frac{\sum_{i=1}^{n} \left\| \vec{x}_{start,tree(n)} \right\|}{n} \qquad (10)$$

In Eq. (10), n is the number of nodes, and $\vec{x}_{start,tree(n)}$ is the tree node. Inflated obstacles are treated as spheres to avoid collisions. The cost of avoiding obstacles is expressed in Eq. (11).

$$C_{obs} = \sum_{i=1}^{n} C^{i}_{obsSingle} \qquad (11)$$

Eq. (11) is the total obstacle cost for the route, and $C^{i}_{obsSingle}$ is the cost for a single obstacle. The radius of the endpoint area is calculated by Eq. (12).

$$\mathrm{R}_{end} = \frac{\min_{i=1}^{n} \left\{ \left\| \vec{x}_{start,tree(n)} \right\| \right\}}{n} \qquad (12)$$

Eq. (12) is the radius of the end point area. Determining the endpoint radius predicts the optimal route in that region. Then, KNG is introduced to further filter nodes, reduce route length, and improve efficiency. The KNG algorithm process is shown in Fig. 5.

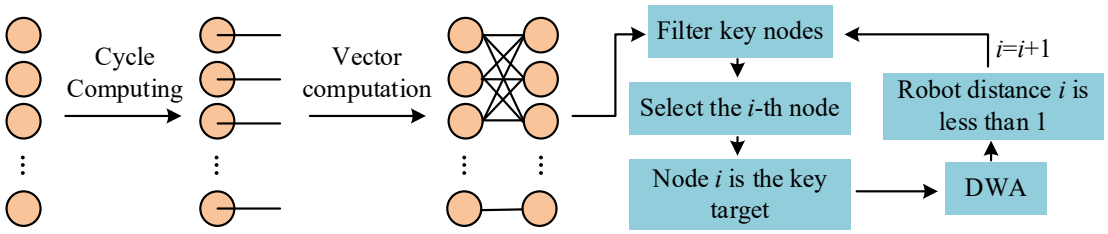


Fig. 5. KNG operation flow chart

In Fig. 5, the KNG algorithm clusters generated nodes. It creates route segments between adjacent nodes, calculates route directions in a loop, and computes angle differences using direction vectors. Nodes with angle differences less than 20° are deleted, and the remaining key nodes are kept and passed into the DWA algorithm. Finally, the algorithm checks if the key target has been reached; if so, it ends. If not, it returns to run DWA again until the robot is within 1m of the target. The mathematical expression for the direction vector is shown in Eq. (13).

$$d_i = \frac{p_{i+1} - p_i}{\left\| p_{i+1} - p_i \right\|} \tag{13}$$

Eq. (13) is the number of nodes, d is the direction vector, and p is the node route. The angle calculation equation of the direction vector is shown in Eq. (14).

$$\theta_i = atan2(d_{i1}, d_{i2}) \tag{14}$$

In Eq. (14), θ is the direction vector angle, d_{i1} is the first component of the direction vector, and d_{i2} is the second component. The angle difference equation between adjacent node routes is shown in Eq. (15).

$$\Delta\theta_i = \theta_{i+1} - \theta_i \tag{15}$$

The calculated angle differences are filtered by deleting nodes with angle differences of less than 20°, retaining the remaining key nodes, which are then introduced into DWA. However, ISSA-A*-DWA still generates too many branch routes and has low navigation efficiency. Therefore, this study combines MRBS and KNG algorithms to build the MRBS-KNG-A*-DWA model and applies it to dynamic route planning for warehouse robots. The detailed process is shown in Fig. 6.

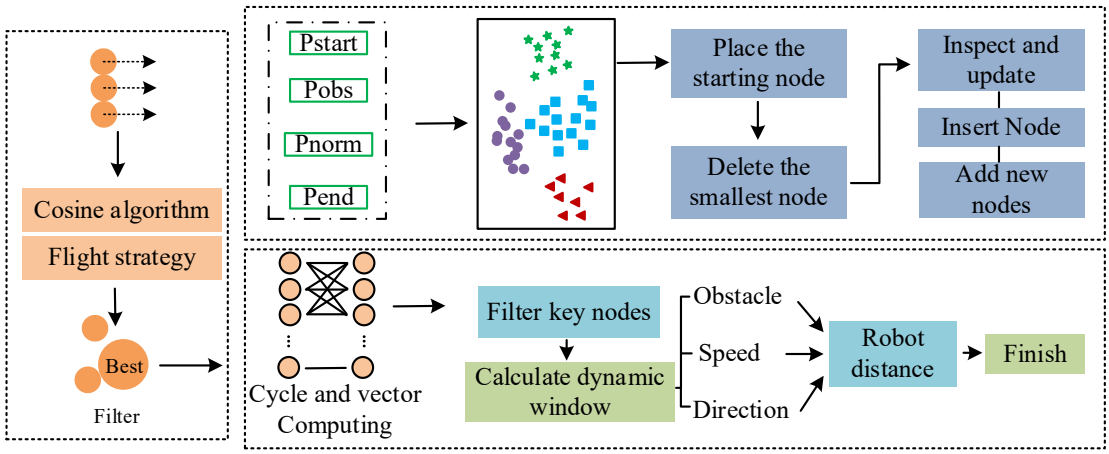


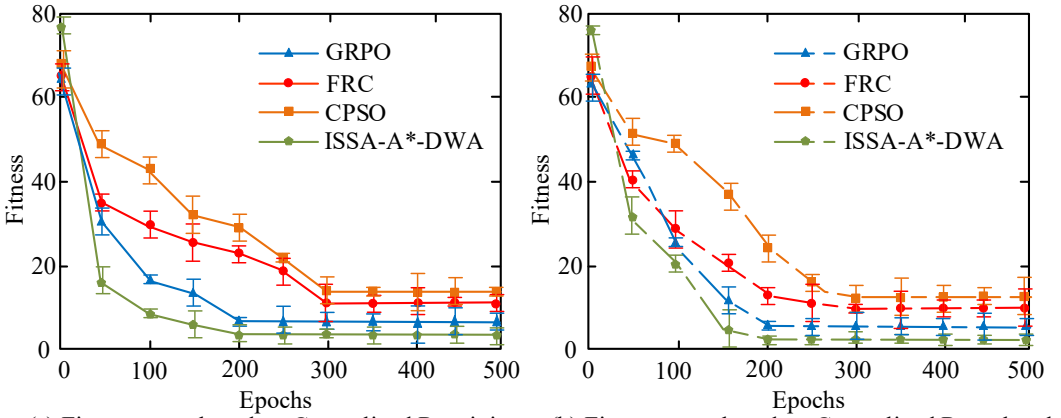
Fig. 6. Proposed dynamic route planning process for warehouse robots

In Fig. 6, the model initializes population positions and updates them using Cauchy opposition-based learning, sine-cosine algorithm, and Lévy flight strategy to get the optimal value. Then, MRBS divides the map into regions, calculates the sampling radius, plans the distance and iteration times, inflates obstacles, normalizes the data, and obtains obstacle-avoidance routes, thereby speeding up route generation. The expanded map route selects the minimum nodes in the open list and moves them to the closed list for looped processing, optimizing the evaluation function. Finally, KNG clusters generate nodes, create route segments, calculate direction vector angle differences, filter and keep key nodes, and pass them to DWA. DWA updates the robot's position, calculates velocity space, and uses the dynamic window evaluation function to output target values.

4. Performance Analysis of Dynamic Route Planning Strategy for Warehouse Robots

4.1. Validation of ISSA-A*-DWA Effectiveness

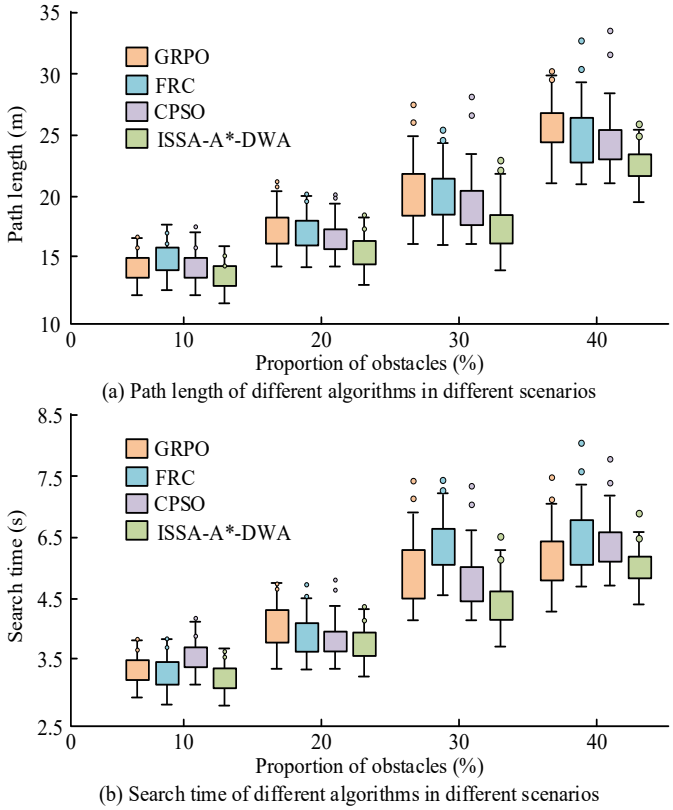
To verify the effectiveness of the proposed algorithm, the study compared it with Group Relative Policy Optimization (GRPO), FRC, and CPSO. The experiments were conducted on Windows 11 and Ubuntu 18.04, utilizing an Intel(R) Core(TM) i7-5500 processor, 8GB RAM, MATLAB R2023a as the software platform, and the Robot Operating System (ROS) as the framework. The learning rate was set to 0.005 with 500 iterations, and the maximum iteration time is 500s. The study evaluated the optimal fitness of ISSA-A*-DWA, GRPO, FRC, and CPSO using the Generalized Rastrigin and Generalized Rosenbrock functions. The results are shown in Fig. 7.



(a) Fitness curve based on Generalized Rastrigin (b) Fitness curve based on Generaliaed Rosenbrock

Fig. 7. Comparison of fitness curves of different functions

As shown in Fig. 7(a), on the Generalized Rastrigin function, the fitness curve of ISSA-A*-DWA dropped rapidly within the first 50 iterations. The decline then slowed and stabilized after about 120 iterations, reaching an optimal fitness value of 6.2. Fig. 7(b) shows that, on the Generalized Rosenbrock function, the fitness curve of the proposed algorithm dropped sharply from 0 to 20 iterations, slowed down between 20 and 130 iterations, and then stabilized. Compared to other algorithms, the proposed algorithm achieved a significantly better fitness curve, likely because ISSA's cooperative mechanism enabled the discovery of better routes and a broader search range. Overall, compared with GRPO, FRC, and CPSO, the proposed algorithm exhibited superior fitness curves and faster convergence speed, demonstrating that ISSA-A*-DWA can find optimal fitness values more quickly and possesses excellent optimization ability. To further demonstrate the route length and search time of ISSA-A*-DWA, it was compared with GRPO, FRC, and CPSO. The testing results are shown in Fig. 8.

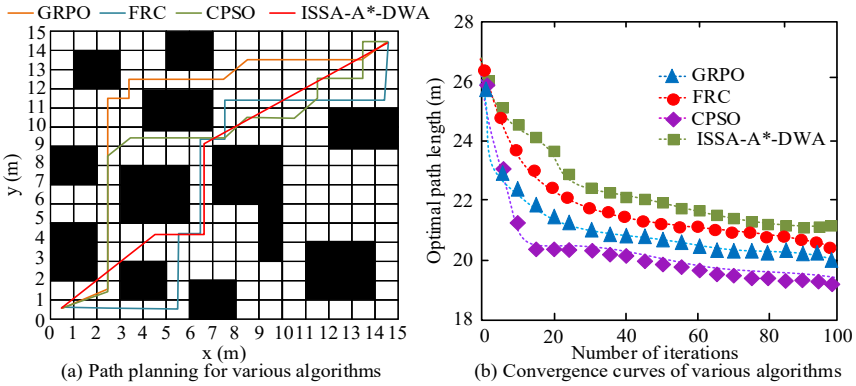


(a) Path length of different algorithms in different scenarios

(b) Search time of different algorithms in different scenarios

Fig. 8. Comparison of route length and search time experimental results

Fig. 8(a) shows that in scenarios with different obstacle ratios, ISSA-A*-DWA achieved the shortest route length of 13m at a 10% obstacle ratio. With an increasing obstacle ratio, the route length grew to 22.5m. The route lengths and trends were overall smaller than those of the other algorithms. Fig. 8(b) shows that ISSA-A*-DWA has a significant time advantage, with a search time of 3.1 s at 10% obstacle ratio, rising to 4.9 s at 40% obstacle ratio. The results indicated better pathfinding efficiency and lower sensitivity to scenario changes, due to reduced redundant nodes and optimized computational layers. To further demonstrate the route-planning and convergence performance in simulation, the study tested ISSA-A*-DWA alongside GRPO, FRC, and CPSO. Results are shown in Fig. 9.



(a) Path planning for various algorithms (b) Convergence curves of various algorithms

Fig. 9. Comparison of route planning and convergence performance

Fig. 9(a) shows that ISSA-A*-DWA had only 3 turns, while GRPO, FRC, and CPSO had 7, 7, and 10 turns, respectively. This demonstrates superior route smoothness for the proposed algorithm. Fig. 9(b) shows that after 10 iterations, the optimal route length was 21m. From 10 to 60 iterations, the length steadily decreased to 18m, then stabilized. After 100 iterations, the loss value reached 17m, with a trend better than the other three algorithms. This indicated the proposed algorithm could predict collision possibilities, encouraging the robot to choose the correct direction. In summary, the proposed algorithm had good route optimization accuracy and obstacle avoidance ability.

4.2. Application Effect Analysis of the Dynamic Route Planning Model

After validating ISSA-A*-DWA's performance, the study further tested the warehouse robot dynamic route planning model against the Improved Snake Optimizer-Time Window (ISO-TW), Novel Artificial Bee Algorithm with Adaptive Disturbance (IGABC), and Hybrid Particle Swarm Optimization (HPSO). The experiments used a Linux-based Robot Operating System simulation, MATLAB for map simulation, Gmapping for mapping, C++ as the programming language, and VM Ware as the virtual machine. The detailed parameters of the experimental device are shown in Table 1.

Table 1. Experimental setting parameters

Experimental setup	Parameter
Obstacle distribution	Regular grid shelf
Obstacle passage width	2m
Robot model	Ackerman steering
Maximum speed of robot movement	1.5m/s
Robot moving acceleration	0.3m/s
Map resolution	0.2m

The study constructed a 1000×1000 3D space with three obstacles as a virtual map and tested MRBS-KNG-A*-DWA, ISO-TW, IGABC, and HPSO models for pathfinding performance. Results are shown in Fig. 10.

Fig. 10(a), 10(b), and 10(c) show that the three comparison models generated search trees with many branches near the route, and the number of branch nodes increased to varying degrees. The number of route nodes was 60.25, 58.41, and 24.69, respectively. Fig. 10(d) shows that the proposed model's route search tree consisted of regular long edges, with significantly fewer branch nodes. The node count was 4.65. The few branches improved sampling probability in key areas, creating more key nodes. Moreover, branches mainly appeared near obstacles, indicating better node utilization. Overall, MRBS-KNG-A*-DWA demonstrated superior route search performance. To further analyze turn counts during pathfinding, the study compared the proposed model with ISO-TW, IGABC, and HPSO. Results are shown in Fig. 11.

Fig. 11 shows that at a 10% obstacle ratio, the proposed model had the lowest turn count of 6, while ISO-TW, IGABC, and HPSO had 10, 8, and 12 turns, respectively. As the obstacle ratio increased, the number of turns increased to 11 for MRBS-KNG-A*-DWA and rose to 18, 17, and 21 for the other three models. The overall growth trend of the proposed algorithm was lower than that of ISO-TW, IGABC, and HPSO. The results indicated good route optimization ability. To further demonstrate pathfinding success rates under different obstacle ratios, the study evaluated the performance of MRBS-KNG-A*-DWA, ISO-TW, IGABC, and HPSO models. Results are shown in Table 2.

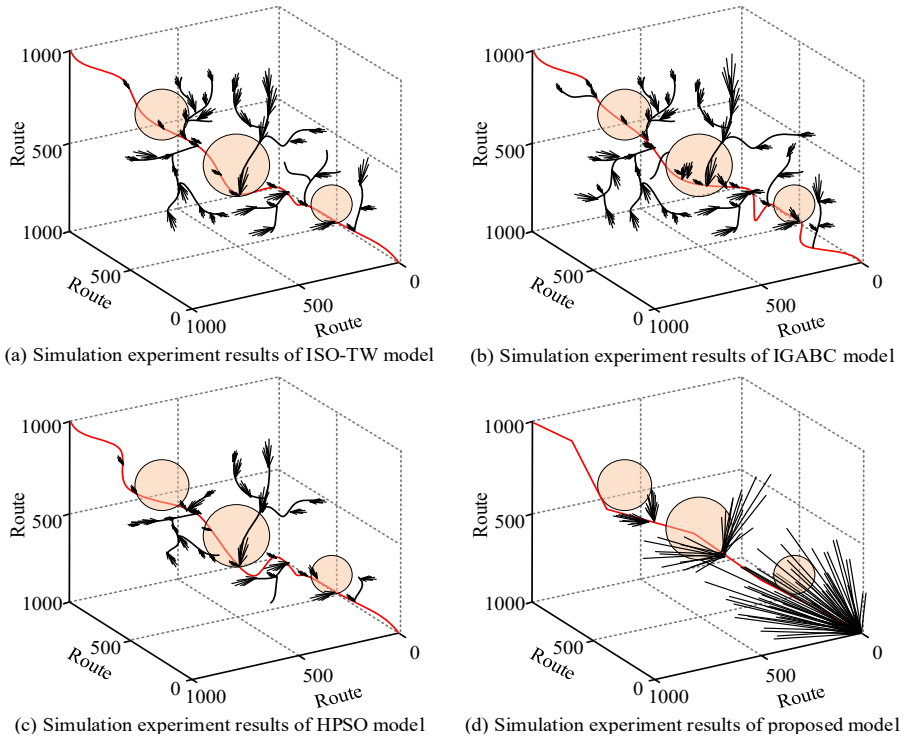


Fig. 10. Comparison of pathfinding effects

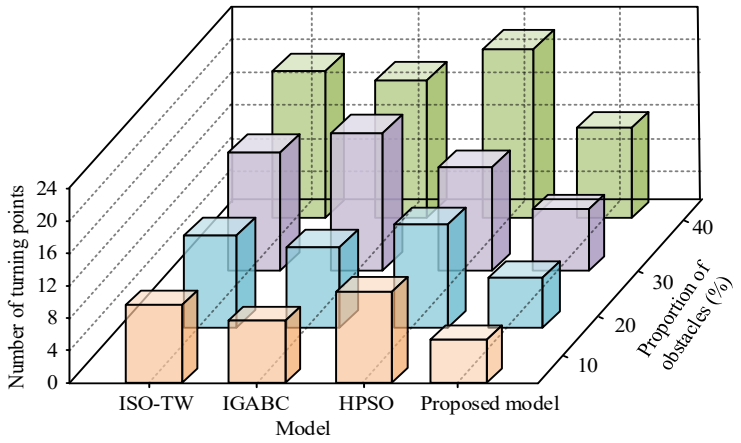


Fig. 11. Comparison of the number of turns during pathfinding

Table 2. Pathfinding success rate in different obstacle scenarios

Proportion of obstacles (%)	Success rate (%)			
	ISO-TW	IGABC	HPSO	Proposed model
10	91.25±1.94**	89.36±1.58**	93.25±1.01*	97.35±0.58
15	88.69±2.31**	85.45±1.94**	91.65±1.66**	95.66±1.02
20	81.36±1.58**	81.69±2.58**	88.55±1.87**	92.57±1.15
25	74.65±2.36**	79.44±3.47**	80.47±2.08**	89.69±2.04
30	66.25±2.64***	71.25±3.94**	74.25±2.69**	88.57±2.21
35	61.89±3.41***	65.24±4.87***	66.27±3.44**	85.63±2.69
40	54.62±5.69***	60.87±5.02***	61.88±4.52***	82.95±3.46

Note: * represents $P<0.05$, ** represents $P<0.01$, and *** represents $P<0.001$.

Table 2 shows that at a 10% obstacle ratio, success rates were 97.35%, 91.25%, 89.36%, and 93.25% for MRBS-KNG-A*-DWA, ISO-TW, IGABC, and HPSO, respectively. As the obstacle ratio increased, the success rates decreased for

MRBS-KNG-A*-DWA, reaching 82.95% at a 40% obstacle ratio. Despite the decrease, the success rates remained higher than those of the other three models, improving by 28.33%, 23.08%, and 21.07%, respectively. The overall decreasing trend was also less than that of ISO-TW, IGABC, and HPSO. In summary, the proposed model achieved high success rates across various obstacle scenarios, guiding robots to avoid them and demonstrating excellent route-planning performance effectively. In order to further verify the path planning performance of the model under dynamic moving obstacles, the research adopts the multi robot cooperation mode, sets the number of robots to 5, and the moving speed of obstacles to 1m/s and 1.5m/s, and conducts comparative analysis on the collision times, maximum curvature, iterative optimization time, and actual energy consumption of the research model, Rapid Expanding Random Tree Star (RRT *), and SSA model, respectively. The results are shown in Table 3.

Table 3. Experimental results of each model

Obstacle speed	Model	Number of collisions	Curvature (m^{-1})	Iteration time (s)	Energy consumption (kW/h)
1 m/s	Research model	2±1	0.72±0.09	25.69±1.69	2.62±0.01
	SSA	$11±2^{***}$	$1.06±0.18^{*}$	$34.51±2.64^{*}$	$3.69±0.04^{**}$
	ISSA	$7±1^{**}$	$0.80±0.10^{*}$	$30.46±1.77^{*}$	$3.02±0.09^{*}$
	RRT*	$8±1^{**}$	$1.54±0.20^{**}$	$32.12±2.71^{*}$	$3.52±0.10^{**}$
1.5 m/s	Research model	3±1	0.81±0.12	29.54±1.41	3.44±0.03
	SSA	$15±2^{***}$	$1.99±0.25^{**}$	$40.21±3.65^{**}$	$4.58±0.12^{**}$
	ISSA	$8±1^{**}$	$1.14±0.26^{*}$	$34.28±2.64^{**}$	$3.98±0.14^{*}$
	RRT*	$11±2^{**}$	$2.02±0.32^{**}$	$45.30±3.74^{**}$	$4.69±0.19^{**}$

Note: * represents $P<0.05$, ** represents $P<0.01$, and *** represents $P<0.001$.

As shown in Table 3, the collision times, maximum curvature, iterative optimization time, and actual energy consumption of the research model are 2 times, 0.72m-1, 25.69s, and 2.62kW/h, respectively, at the moving speed of 1 m/s obstacles, which are better than those of the SSA, ISSA, and RRT* models. As the moving speed of obstacles increases, the collision times, maximum curvature, iterative optimization time, and actual energy consumption of the MRBS-KNG-A*-DWA model increase to three times, 0.81m-1, 29.54s, and 3.44kW/h. Although they increase, they are still superior to other comparison models, and the growth trend of the proposed model's index is smaller than that of the comparison model. In conclusion, in the multi-robot cooperation mode, the proposed model can guide the robot to avoid moving obstacles effectively, reduce the number of collisions, optimize the iteration time, and reduce energy consumption at the same time, indicating that the research model shows excellent path planning performance in moving obstacles in the multi-robot cooperation mode.

5. Conclusions and Recommendations

To address the challenges of high manual dependency and low diagnostic efficiency in dynamic route planning for warehouse robots, a dynamic route planning model based on a hybrid of A* and DWA algorithms was constructed. This model utilized ISSA to optimize operating parameters and enhance information search capabilities. Regional bias sampling and the KNG algorithm were employed to improve the efficiency of route node generation and accelerate route planning, ultimately achieving a dynamic route planning model for warehouse robots. Experimental results demonstrated that the A* and DWA hybrid algorithm outperformed the GRPO, FRC, and CPSO algorithms in a scenario with a 40% obstacle ratio, achieving a route length of 22.5 m, a search time of 4.9 s, and three turns in a simulated environment. Furthermore, the constructed MRBS-KNG-A*-DWA dynamic route planning model significantly reduced the generation of bifurcations. With a 40% obstacle ratio, the model achieved 11 turns and an 82.95% pathfinding success rate, surpassing the ISO-TW, IGABC, and HPSO models across all metrics. In addition, at the moving speeds of 1m/s and 1.5m/s, the collision times, maximum curvatures, iterative optimization time, and actual energy consumption indicators are all lower than those of the comparison model. Overall, the A*-DWA algorithm-based dynamic route planning model efficiently planned optimal paths, reduced route distance and search time, and improved the accuracy of dynamic route planning for warehouse robots. In this experiment, the dynamic path planning of the storage robot is limited to a single obstacle scene and does not involve research on real-time performance and robustness in a highly dynamic pedestrian environment. Therefore, in the future, the path planning with deep reinforcement learning can be studied in depth.

Author Contributions

Jianjun Li contributed to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, supervision, project administration, and funding acquisition. Ji Xiang contributed to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, supervision, project administration, and funding acquisition.

Declaration of Artificial Intelligence (AI) Tools

The authors used AI tools solely for language editing and readability improvement. The authors reviewed and verified all content and take full responsibility for the accuracy and integrity of the manuscript.

References

Chandra, O. R. and Istiono, W. (2022). A-star Optimization with Heap-sort Algorithm on NPC Character. *Indian Journal of Science and Technology,* 15(35), 1722-1731.

Hidayat, A. R., Gernowo, R., and Sugiharto, A. (2023). Comparison Analysis of Nearest Road Calculations on Dijkstra Algorithm and A*(A-Star) Algorithm for Mapping BTS Tower Area. *Journal of Social Research,* 2(10), 3419-3427.
Ishihara, S., Kanai, M., Narikawa, R., and Ohtsuka, T. (2022). A proposal of path planning for robots in warehouses by model predictive control without using global paths. *IFAC-PapersOnLine,* 55(37), 573-578.
Jathunga, T. and Rajapaksha, S. (2025). Improved Path Planning for Multi-Robot Systems Using a Hybrid Probabilistic Roadmap and Genetic Algorithm Approach. *Journal of Robotics and Control,* 6(2), 715-733.
Kappagantula, S. and Mannayee, G. (2024). Dynamic path planning algorithm for mobile robots: Leveraging reinforcement learning for efficient navigation. *Journal of Internet Services and Information Security,* 14(2), 226-236.
Lai, X., Wu, D., Wu, D., Li, J., H., and Yu, H. (2023). Enhanced DWA algorithm for local path planning of mobile robot. *Industrial Robot: The International Journal of Robotics Research and Application,* 50(1), 186-194.
Li, C., Yao, L., and Mi, C. (2025). Fusion Algorithm Based on Improved A* and DWA for USV Path Planning. *Journal of Marine Science and Application,* 24(1), 224-237.
Li, Q., Ma, Q., and Weng, X. (2024). Dynamic path planning for mobile robots based on artificial potential field enhanced improved multi-objective snake optimization (APF-IMOSO). *Journal of Field Robotics,* 41(6), 1843-1863.
Lin, Z., Wu, K., Shen, R., Yu, X., and Huang, S. (2023). An efficient and accurate A-star algorithm for autonomous vehicle path planning. *IEEE Transactions on Vehicular Technology,* 73(6), 9003-9008.
Pattyn, F. (2025). The value of generative AI for qualitative research: A pilot study. *Journal of Data Science and Intelligent Systems,* 3(3), 184-191.
Rasheed, A., A., A., Al-Araji, A., S., and Abdullah, M., N. (2022). Static and dynamic path planning algorithms design for a wheeled mobile robot based on a hybrid technique. *International Journal of Intelligent Engineering and Systems,* 15(4), 167-181.
Rizki, A., C., Athallah, K. R., and Siddiq, S. M. (2025). Performance Analysis of Dijkstra and A-star Algorithm in Maritime Navigation Pathfinding. *Journal of Mechatronics and Artificial Intelligence,* 2(2), 31-50.
Sahoo, S. K. and Choudhury, B., B. (2023). A review of methodologies for path planning and optimization of mobile robots. *Journal of Process Management and New Technologies,* 11(1-2), 122-140.
Sandanika, W., A., H., Wishvajith, S., H., Randika, S., Thennakoon, D., A., Rajapaksha, S., K., and Jayasinghearachchi, V. (2024). Ros-based multi-robot system for efficient indoor exploration using a combined path planning technique. *Journal of Robotics and Control,* 5(5), 1241-1260.
Sarbini, R., N., Ahmad, I., Bura, R., O., and Simbolon, L. (2024). Development of pathfinding using a-star and d-star lite algorithms in video game. *Journal of Theoretical and Applied Information Technology,* 102(3), 832-840.
Shi, K., Wu, Z., Jiang, B., and Karimi, H. R. (2023). Dynamic path planning of mobile robot based on improved simulated annealing algorithm. *Journal of the Franklin Institute,* 360(6), 4378-4398.
Vikas and Parhi, D., R. (2023). Multi-robot path planning using a hybrid dynamic window approach and modified chaotic neural oscillator-based hyperbolic gravitational search algorithm in a complex terrain. *Intelligent Service Robotics,* 16(2), 213-230.
Xue, J. and Shen, B. (2024). A survey on sparrow search algorithms and their applications. *International Journal of Systems Science,* 55(4), 814-832.
Wang, J., Zhao, N., and Mi, C. (2024). ART path planning method based on the 3D steering angle weighted A-star algorithm. *International Journal of Vehicle Information and Communication Systems,* 9(3), 256-275.
Wu, P., Zhong, L., Xiong, J., and Zeng, Y. (2023). Two-level vehicle path planning model for multi-warehouse robots with conflict solution strategies and improved ACO. *Journal of Intelligent and Connected Vehicles,* 6(2), 102-112.
Yudha, M. H. P., Supian, S., and Napitupulu, H. (2022). Optimization Route to Tourism Places in West Java Using A-STAR Algorithm. *CAUCHY: Jurnal Matematika Murni dan Aplikasi,* 7(3), 464-473.

Jianjun Li was born in HeBei, China, in 1983. From 2009 to 2012, he studied at South China University of Technology and received his master's degree in 2012. He currently works for Cainiao Network, an Alibaba company. His research interests include multi-sensor fusion perception, robot navigation algorithms, distributed multi-agent collaborative scheduling, and dynamic path planning. He has rich experience in the engineering implementation of large-scale complex systems.

Ji Xiang (Senior Member, IEEE) received his Ph.D. in Control Science and Engineering from Zhejiang University, Hangzhou, China, in 2005. He is currently a professor in the Department of Systems Science and Engineering at Zhejiang University. From 2005 to 2007. He was a postdoctoral researcher at Zhejiang University and visited the City University of Hong Kong for 3 months. With support from a Gledden Senior Visiting Fellowship, he visited the University of Western Australia in Crawley, WA, Australia, in 2008. From 2013 to 2014, he was a visiting scholar at the University of Sydney, Sydney, NSW, Australia. His research interests include manipulator control, autonomous surface vehicles, and microgrids.