



# An Enhanced Hybrid Metaheuristic for Hierarchical Scheduling in 4WIDS Multi-robot Systems under Confined Environments

Lin Zhang, Yichen An, Tianwei Niu, Runjiao Bao, Shoukun Wang\*, Junzheng Wang

*School of Automation, Beijing Institute of Technology, Beijing 100081, China*

## ARTICLE INFO

### Keywords:

Multi-robot systems  
Task assignment  
Path planning  
Four-wheel independent drive/steering robot

## ABSTRACT

Multi-robot systems have emerged as a transformative paradigm for industrial automation. However, deploying these systems in dense, dynamic environments like ultra-dense warehouses and Ro-Ro terminals remains challenging due to simplified motion constraints, idealized models, and the tight coupling of task assignment, trajectory planning, and conflict resolution under strict spatiotemporal constraints. To address these problems, we propose a hierarchical scheduling framework for four-wheel independent drive/steering robot groups in confined environments. Firstly, at the task assignment layer, we introduce an enhanced hybrid metaheuristic for task assignment that integrates particle swarm optimization with a genetic algorithm, augmented by a problem-specific fitness function and adaptive mutation strategies to prevent premature convergence. Secondly, at the path planning layer, we develop a kinematics-aware conflict-based search path planner integrating motion primitives with improved A\* node expansion strategies, where adaptive heuristic weighting and bidirectional search acceleration are introduced to ensure computational tractability. Simulations in a near-realistic confined environment show that the proposed hierarchical scheduling algorithm reduces total execution cost by 11.0% compared to the advanced particle swarm genetic algorithm, demonstrating its superior performance in multi-robot coordination. Furthermore, field tests conducted at the Ro-Ro Terminal of Yantai Port have fully validated the feasibility of this framework for multi-robot coordination in real-world scenarios. This work lays a theoretical and practical foundation for next-generation multi-robot coordination in constrained logistics ecosystems.

## 1. Introduction

The advent of Industry 4.0 has catalyzed an urgent demand for intelligent industrial transformation, with multi-robot systems emerging as a critical enabler for achieving operational efficiency, cost reduction, and safety enhancement (Bhargava, Suhaib, & Singholi, 2024; Wu, Wang, & Qiu, 2024). These systems have demonstrated remarkable success in open environments such as unmanned factories (Wang et al., 2025), automated ports (Yang, He, & Sun, 2023), and autonomous mining (Li et al., 2023), where standardized workflows and spatial flexibility allow seamless coordination. However, their deployment in spatially confined industrial environments—exemplified by compact warehouses and Ro-Ro docks for commercial vehicles—remains a significant challenge due to restricted maneuvering areas, frequent robot interactions, and complex kinematic constraints.

To deploy multi-robot systems in spatially confined industrial environments, it is essential to focus on two interdependent dimensions: task assignment and path planning. Task assignment involves determining which tasks each robot must perform and in what order (Bolognini,

Fagiano, & Limongelli, 2023), and path planning focuses on ensuring that multiple robots can navigate the environment without colliding (Xie et al., 2024). Existing approaches typically model task assignment as a multi-constrained optimization problem, considering variables such as robot capabilities, task priorities, and energy consumption (Sun, Liu, & Cai, 2025; Zhai, Li, Wu, Hou, & Jia, 2023). Applicable algorithms, including particle swarm optimization (PSO) (Geng, Ji, & Zi, 2022), genetic algorithm (GA) (Tan, Zhou, & Qian, 2024), and hybrid metaheuristics (Su & Wang, 2021), have shown efficacy in balancing computational efficiency and solution quality. For instance, Amazon's Kiva robots leverage hierarchical task decomposition to optimize goods-to-person workflows (Allgor, Cezik, & Chen, 2023). Furthermore, automated guided vehicles at the Port of Rotterdam employ dynamic scheduling to mitigate congestion (Chen et al., 2020). Despite algorithmic progress, extant models inadequately reflect critical real-world constraints—remarkably heterogeneous robot dimensions and nonholonomic kinematics—resulting in suboptimal task assignments and potential motion conflicts in dense deployments.

\* Corresponding author.

E-mail address: [bitwsk@bit.edu.cn](mailto:bitwsk@bit.edu.cn) (S. Wang).

After task assignment, the safety and efficiency of multi-robot systems hinge on robust multi-robot path planning. Unlike single-robot paradigms, multi-robot path planning must resolve conflicts and avoid deadlocks in real-time. State-of-the-art methods fall into two categories: search-based algorithms (e.g., discontinuity bounded conflict-based search (Moldagalieva, Ortiz-Haro, Toussaint, & Höning, 2024), rolling horizon frameworks (Li et al., 2021) that prioritize computational tractability but oversimplify robot kinematics. Optimization-based approaches (e.g., spatiotemporal safety corridors (Zang et al., 2023), combined with velocity planning spatiotemporal optimization (Ma et al., 2023) that enhance trajectory smoothness at the cost of scalability in dense robot populations. Notably, most existing studies abstract robots as dimensionless agents or assume obstacle-free unstructured environments (Hiraoka et al., 2024; Ruan, Poblete, Wu, Ma, & Chirikjian, 2022), limiting their applicability to high-density industrial settings where physical dimensions and kinematic constraints (e.g., minimum turning radius) dominate system performance.

In parallel, while significant progress has been made in single-robot motion planning for confined spaces (Gao, Li, Liu, Ge, & Song, 2024; Peng, Ngo, & Liu, 2023), adapting these strategies to multi-robot systems reveals additional challenges. Recent methods such as composite dynamic region bias RRT\* (McBeth, Motes, Uwacu, Morales, & Amato, 2023) and Dubins curve-based deadlock resolution (Xu et al., 2023) show preliminary success in narrow corridor navigation. However, three key challenges persist. First, many algorithms simplify motion constraints by limiting robots to basic directional movements while neglecting physical dimensions and nonholonomic requirements (e.g., minimum turning radius). Second, environmental models are overly idealized, often representing complex industrial settings as topological maps or sparse obstacles (Hiraoka et al., 2024), thereby ignoring high-density equipment layouts. Third, the reliance on point-mass approximations and bidirectional mobility—even in studies incorporating some kinematic constraints (Chen et al., 2024)—results in collision avoidance strategies impractical for heterogeneous fleets. These oversimplifications ultimately limit the deployability of current multi-robot systems methods in confined industrial environments characterized by frequent interactions, strict spatial limitations, and complex obstacle distributions.

In recent years, deep reinforcement learning (DRL) has emerged as a powerful tool for multi-robot scheduling and planning. Huo, Mao, San, Li, and Zhang (2025) proposed a two-layer DRL framework with group awareness that effectively addresses conflict coordination in multi-robot pathfinding, significantly enhancing task success rates and inter-agent cooperation. Mao, Wu, Fan, Cao, and Pedrycz (2024) introduced a dual-layer DRL model based on a divide-and-conquer strategy to overcome the scalability and computational challenges in large-scale UAV task scheduling. Sun et al. (2024) extended the dual-parameter deep Q-network to jointly optimize trajectory planning, resource allocation, and task offloading in UAV-assisted MEC systems. Despite these advancements, DRL-based approaches still suffer from slow convergence, high computational cost, and limited interpretability, which limit their applicability in real-world systems. To tackle the coupling between scheduling and path planning, Yan, Chu, Hu, and Zhu (2024) designed an improved genetic algorithm incorporating customized operators and a deadlock-resolution mechanism for resource-constrained, synchronized multi-target assignment. Gao, Zuo, Lu, and Tang (2025) proposed a matrix-encoded genetic algorithm to coordinate heterogeneous agents under tightly coupled task assignment and routing constraints. However, as the scenario becomes more complex and the number of robots increases, the coupling between task assignment and path planning intensifies, posing significant challenges to computational efficiency, scalability, and practical deployment.

To address these challenges, this paper proposes a hierarchical scheduling framework tailored for the operation of multiple four-wheel independent drive/steering (4WIDS) robots in confined environments. Under spatiotemporal constraints, the upper layer handles task and

robot assignment, while the lower layer performs collision-free path planning based on the assigned tasks. This framework effectively integrates task assignment with path planning, enabling coordinated multi-robot operation. The main contributions of this work are as follows:

- A hybrid metaheuristic algorithm that combines PSO with GA is developed for efficient task assignment. The integration of adaptive mutation strategies and a problem-specific fitness function effectively mitigates premature convergence and improves solution quality.
- A kinematics-aware, conflict-based multi-robot path planner is proposed, incorporating motion primitives and enhanced A\* node expansion strategies. By leveraging adaptive heuristic weighting and bidirectional search, the planner achieves a favorable balance between computational efficiency and path optimality under motion constraints.
- The proposed hierarchical scheduling framework is thoroughly validated in near-realistic confined simulation environments. Furthermore, field tests at Yantai Port's Ro-Ro Terminal have validated this framework's feasibility for real-world multi-robot coordination.

The paper is structured as follows: Section 2 defines the problem, Section 3 details the hierarchical scheduling method, Section 4 provides simulations, Section 5 presents experiments, and Section 6 concludes.

## 2. Problem formulation

Deploying multi-robot systems presents a hierarchical scheduling challenge with two key components: task assignment and path planning. This section models these components based on an analysis of robots' motion modes.

### 2.1. Establishment of robot kinematics model

In this study, the 4WIDS robot is chosen as the research subject, as illustrated in Fig. 1. Featuring four independently driven and steered wheels, the 4WIDS robot exhibits superior driving capability and maneuverability compared to conventional wheeled robots (Xu, Zhang, Liu, Wang, & Wang, 2025; Zhang et al., 2025). It enables omnidirectional translation, zero-radius rotation, and efficient navigation in confined environments.

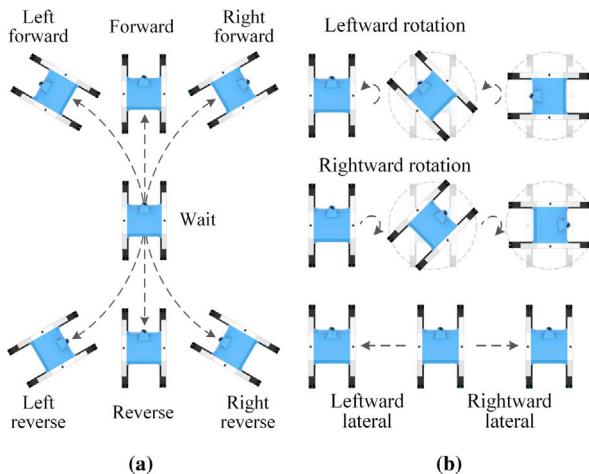
To exploit these capabilities, we extend the traditional Ackermann steering model by incorporating 11 motion modes that combine rotational and lateral movements. This design reduces the turning radius and improves maneuverability in confined spaces. The kinematic formulation of the Ackermann model is as follows:

$$\begin{cases} x_{t+1} = x_t + v_t \cos \theta_t \Delta t, \\ y_{t+1} = y_t + v_t \sin \theta_t \Delta t, \\ \theta_{t+1} = \theta_t + \theta_t \frac{2 \tan \delta_t}{L} \Delta t, \end{cases} \quad (1)$$

where  $x_t$  and  $y_t$  denote the robot's position in the X and Y directions at time  $t$ ,  $\theta_t$  represents the robot's heading angle at time  $t$ , and  $v_t$  and  $\delta_t$  denote the robot's speed and steering angle, respectively, at time  $t$ .  $L$  is the wheelbase of the robot, and  $\Delta t$  is the time step.

### 2.2. Task assignment problem analysis and modeling

The objective of task assignment is to assign a set of tasks to multiple robots, determining both the task-to-robot assignments and the execution sequence for each robot. The goal is to complete all tasks efficiently while minimizing overall resource consumption. This problem is typically formulated as an optimization problem.



**Fig. 1.** The 4WIDS robot motion modes. (a) Traditional motion modes. (b) New motion modes.

Let  $\mathcal{R} = \{r_1, r_2, \dots, r_m\}$  be the set of  $m$  robots,  $\mathcal{I} = \{\pi_1, \pi_2, \dots, \pi_n\}$  be the set of  $n$  tasks, and  $x_{ij}^k = 1$  be the task assignment decision variable indicating that robot  $r_k$  needs to execute task  $\pi_j$  after completing task  $\pi_i$ . Furthermore, let  $y_{kj} = 1$  be the decision variable, indicating that task  $\pi_j$  is executed by robot  $r_k$ . The aim is for all robots to minimize resource consumption and complete all tasks as quickly as possible. Therefore, the objective function for multi-robot task assignment is divided into three components:

$$\mathcal{F} = \alpha_1 \cdot \mathcal{F}_{max\_d} + \alpha_2 \cdot \mathcal{F}_{all\_d} + \alpha_3 \cdot \mathcal{F}_{sw}, \quad (2)$$

where  $\mathcal{F}_{max\_d}$  denotes the maximum travel cost of a single robot, introduced to avoid uneven task distribution and prevent individual overload.  $\mathcal{F}_{all\_d}$  represents the total travel cost of all robots, aiming to reduce system-wide energy consumption and time cost.  $\mathcal{F}_{sw}$  accounts for the cost of motion mode switching (e.g., forward, reverse, turning), which penalizes unnecessary transitions that may increase control difficulty and energy loss. The weights  $\alpha_1, \alpha_2, \alpha_3$  are used to balance these objectives.

The above costs can be further defined as:

$$\mathcal{F}_{max\_d} = \max_{1 \leq k \leq m} \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij}^k, \quad (3)$$

$$\mathcal{F}_{all\_d} = \sum_{k=1}^m \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij}^k, \quad (4)$$

$$\mathcal{F}_{sw} = \sum_{k=1}^m \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}^k, \quad (5)$$

where  $d_{ij}$  represents the calculated distance cost from task  $\pi_i$  to task  $\pi_j$ ,  $c_{ij}$  represents the motion-mode switching cost from task  $\pi_i$  to task  $\pi_j$ .

In summary, the multi-objective optimization problem of task assignment can be formulated as follows:

$$\min \quad \alpha_1 \cdot \mathcal{F}_{max\_d} + \alpha_2 \cdot \mathcal{F}_{all\_d} + \alpha_3 \cdot \mathcal{F}_{sw}, \quad (6)$$

Subject to:

$$\sum_{k=1}^m \sum_{i=1}^n \sum_{j=1}^n x_{ij}^k = n, \quad (7)$$

$$\sum_{k=1}^m y_{kj} = 1, \quad \forall j, \quad (8)$$

$$\sum_{i=1}^n x_{ij}^k = y_{kj}, \quad \forall j, k, \quad (9)$$

$$\sum_{j=1}^n x_{ij}^k = y_{ki}, \quad \forall i, k, \quad (10)$$

where constraint (7) requires the robots to complete all tasks, constraint (8) ensures that each task is performed by a robot, and constraints (9) and (10) ensure that only one robot performs each task.

### 2.3. Path planning problem analysis and modeling

Path planning aims to generate optimal, collision-free trajectories for multiple robots, ensuring they reach their task points safely. This study models the environment as a grid-based map, treating fixed infrastructure as impassable and planning robot movements within navigable areas.

Specifically, in a shared, bounded 2D confined space  $\mathcal{W}$  with impassable regions  $\mathcal{O}$ , where each obstacle  $o \in \mathcal{O}$  is a closed subset of  $\mathcal{W}$ . The free workspace for multiple robots is  $F = \mathcal{W} \setminus \mathcal{O}$ , and the area occupied by each robot is  $\mathcal{R}(r_k)$ . For robot  $r_k$ , the starting point  $s_k^i$  of task  $\pi_i$  is either its initial position or the endpoint of its previous task, while the endpoint  $g_k^i$  is the start of the next task or the final destination. The planned path from the initial position to the final task point is represented as  $\mathbf{p}_k = \{z_k^1, z_k^2, \dots, z_k^{t_k}\}$ , where  $z_k^t = [x_k^t, y_k^t, \theta_k^t]^T$  denotes the state of robot  $r_k$  at time  $t$ , and  $t_k$  is the time step when robot  $r_k$  reaches the final task point. The final trajectory  $\mathcal{V}_k$  is a time-parameterized path  $\mathbf{p}_k(t)$ . Under the kinematic constraints defined by Eq. (1), each robot  $r_k$  must travel from its initial task point to its final destination while avoiding collisions with obstacles and other robots. The associated constraints are expressed as follows:

$$z_k^1 = s_k^1, \quad 1 \leq k \leq m, \quad (11)$$

$$g_k^i = s_k^{i+1}, \quad 1 \leq k \leq m, 1 \leq i \leq n-1, \quad (12)$$

$$z_k^{t_k} = g_k^n, \quad 1 \leq k \leq m, \quad (13)$$

$$\mathcal{R}(\mathbf{p}_k(t)) \cap o = \emptyset, \quad 1 \leq k \leq m, 0 \leq t, o \in \mathcal{O}, \quad (14)$$

$$\mathcal{R}(\mathbf{p}_k(t)) \cap \mathcal{R}(\mathbf{p}_p(t)) = \emptyset, \quad 1 \leq k \neq p \leq m, 0 \leq t, o \in \mathcal{O}. \quad (15)$$

## 3. Methods

In this section, we propose a hierarchical scheduling framework for 4WIDS robots in confined environments (see Fig. 2). Comprising a task assignment layer and a path planning layer, the framework enables seamless multi-robot collaboration.

### 3.1. Task assignment method

Considering the efficiency of bio-inspired methods in high-dimensional, multi-constraint scenarios, we enhance the conventional PSO by refining encoding/decoding, proposing a precise fitness function, and integrating an adaptive update strategy with GA optimization. The pseudocode of the proposed adaptive particle swarm genetic algorithm (APSGA) is shown in Alg. 1.

To streamline encoding and decoding, we adopt an integer encoding scheme where each particle represents a task assignment strategy. Each particle consists of  $n$  dimensions corresponding to tasks, which are evenly distributed among  $m$  robots. Given the number of robots  $m$  and task dimensionality  $n$ , we determine task assignment and execution order. Fig. 3 illustrates the encoding and decoding process.

To assess task assignment quality, we define a multi-objective optimization function as the fitness metric for particle positions. Given the spatial constraints and mode-switching requirements in confined environments, the A\* search algorithm sequentially computes the distance from the robot's start position to each task point. Motion-mode

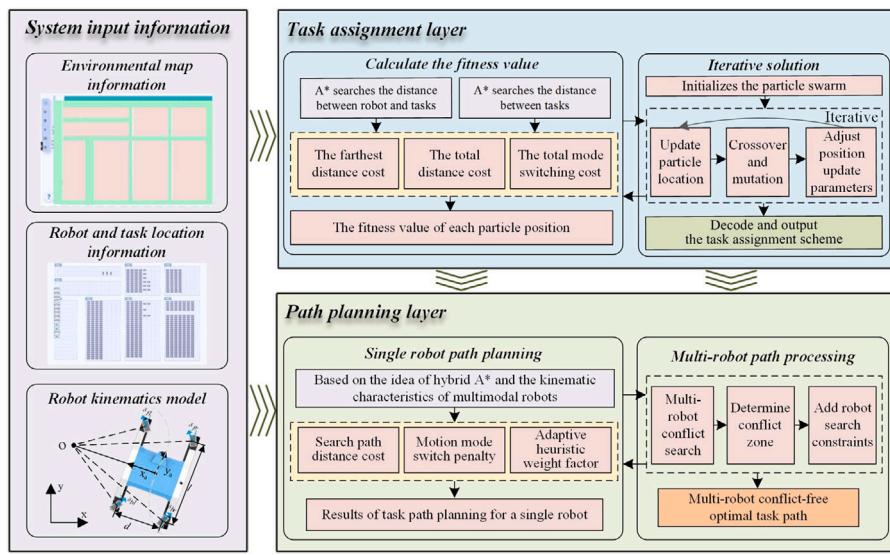


Fig. 2. Hierarchical scheduling framework for multi-robot systems.

**Algorithm 1** An Adaptive Particle Swarm Genetic Algorithm for Task Assignment.

**Require:** Task information, robot position, basic parameters.  
**Ensure:** Task assignment result  $D$ .

- 1: Initialize particle position  $x_{i,0}^d$  and velocity  $v_{i,0}^d$ ;
- 2: Calculate fitness  $F_{i,0}$  by Equ. (2);
- 3: Determine global best  $g_{best}^d$  and local best  $p_{0,best}^d$ ;
- 4:  $t \leftarrow 1$ ;
- 5: **while**  $t \leq T$  **do**
- 6:   **for**  $i \leftarrow 1$  to  $N$  **do**
- 7:     Update particle position  $x_{i,t+1}^d$  by Equ. (16);
- 8:     Update particle velocity  $v_{i,t+1}^d$  by Equ. (16);
- 9:   **end for**
- 10:   Crossover and Mutation;
- 11:   **for**  $i \leftarrow 1$  to  $N$  **do**
- 12:     Recalculate fitness  $F_{i,t}$  by Equ. (2);
- 13:     Update global best  $g_{best}^d$  and local best  $p_{0,best}^d$ ;
- 14:   **end for**
- 15:   Adjust parameters by Equ. (18) and (20);
- 16:    $t \leftarrow t + 1$ ;
- 17: **end while**
- 18: Decode the global best position  $g_{best}^d$ ;
- 19: **return**  $D$ ;

Particles :	12	13	3	6	7	16	1	9	11	15	10	8	14	2	4	5																
	Robot 1's task				Robot 2's task				Robot 3's task				Robot 4's task																			
Execution sequence:	Robot 1:	12	→	13	→	3	→	6	Robot 2:	7	→	16	→	1	→	9	Robot 3:	11	→	15	→	10	→	8	Robot 4:	14	→	2	→	4	→	5

Fig. 3. Coding and decoding criteria of particle swarm.

switching costs are estimated based on orientation angle differences between the start point and the first task and between consecutive tasks in the execution sequence. As a minimization problem, a lower fitness value indicates a more optimal task assignment scheme.

When updating particle velocity and position in the PSO process, if a non-integer value appears in a dimension of the computed position, its rounding direction is probabilistically determined by its proximity to the two nearest integers. The velocity and position update, along with

the rounding process, is as follows:

$$\begin{cases} v_{i,t+1}^d = \omega v_{i,t}^d + c_1 r_1 (p_{i,best}^d - x_{i,t}^d) + c_2 r_2 (g_{best}^d - x_{i,t}^d), \\ x_{i,t+1}^d = x_{i,t}^d + v_{i,t}^d, \end{cases} \quad (16)$$

$$\begin{cases} P_{down} = \lceil x_{i,t+1}^d \rceil - x_{i,t+1}^d, \\ P_{up} = x_{i,t+1}^d - \lfloor x_{i,t+1}^d \rfloor, \end{cases} \quad (17)$$

where  $x_{i,t}^d = (x_{i,t}^1, x_{i,t}^2, \dots, x_{i,t}^d)$  and  $v_{i,t}^d = (v_{i,t}^1, v_{i,t}^2, \dots, v_{i,t}^d)$  denote the position and velocity of the  $i$ th particle at iteration  $t$ . The local and global best positions are given by  $p_{i,best}^d = (p_{i,best}^1, p_{i,best}^2, \dots, p_{i,best}^d)$  and  $g_{best}^d = (g_{best}^1, g_{best}^2, \dots, g_{best}^d)$  respectively. Here,  $r_1$  and  $r_2$  are uniformly distributed in  $[0, 1]$ ,  $\omega$  is the inertia weight, and  $c_1$ ,  $c_2$  are learning factors,  $P_{down}$  and  $P_{up}$  represent the probabilities of rounding down and up, respectively.

To prevent premature convergence to local optima and enhance the search efficiency for the global optimum, the inertia parameter and learning factors are adaptively adjusted based on the iterative evolution process and particle aggregation degree. As shown in Fig. 4, leveraging the strengths of the particle swarm genetic algorithm (PSGA), particles from the current population are randomly paired during the crossover and mutation stages to form  $N/2$  parent pairs. Two random fragments are exchanged for each pair to generate offspring particles, which undergo mutation with predefined probabilities. If a particle contains duplicate values in different dimensions (representing redundant task assignments), one duplicate is randomly removed, and an unassigned task is selected to maintain task integrity. The adaptive particle adjustment process is as follows:

$$\omega = \omega_{max} - \frac{(\omega_{max} - \omega_{min}) \cdot t}{T} + \gamma \frac{F_{g,t}}{F_{a,t}}, \quad (18)$$

$$F_{a,t} = \frac{1}{N} \sum_{i=1}^N F_{i,t}, \quad (19)$$

$$\begin{cases} c_1 = c_{max} - \frac{(c_{max} - c_{min})t}{T}, \\ c_2 = c_{min} + \frac{(c_{max} - c_{min})t}{T}, \end{cases} \quad (20)$$

where  $\omega_{max}$  and  $\omega_{min}$  denote the upper and lower bounds of the inertia weight, respectively;  $T$  represents the maximum number of iterations;  $N$  is the total number of particles;  $F_{g,t}$  is the fitness of the global best position at iteration  $t$ ;  $F_{a,t}$  is the average fitness of the swarm at iteration  $t$ ; and  $\gamma$  is the particle aggregation coefficient.

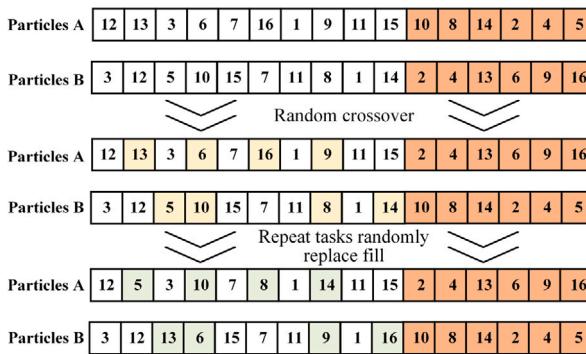


Fig. 4. Diagram of the particle cross-mutation process.

### 3.2. Path planning method

Building on task assignment results, we propose an enhanced conflict-based search algorithm incorporating motion constraints in confined environments. This method extends the traditional conflict-based search by considering the 4WIDS robot's shape and multiple motion modes, while adaptive heuristic weights accelerate the search process. The pseudocode is shown in Alg. 2.

---

#### Algorithm 2 An Enhanced Conflict-Based Search Algorithm for Path Planning.

**Require:** Map and task information, robot location, and robot kinematics model.  
**Ensure:** Conflict-free paths for each robot.

- 1: Initialize binary body conflict tree  $OPEN = \emptyset$ , root node  $N_0.constraints = \emptyset$ ;
- 2:  $N_0.solution, N_0.cost \leftarrow \text{Lower-level\_search}$ ;
- 3:  $OPEN \leftarrow N_0$ ;
- 4: **while**  $OPEN \neq \emptyset$  **do**
- 5:    $N \leftarrow \min_{cost} N, \forall N \in OPEN$  ;
- 6:    $OPEN \leftarrow OPEN \setminus N$ ;
- 7:    $C \leftarrow \text{search for first body conflict in } N.\text{path}$ ;
- 8:   **if** ( $C = \emptyset$ ) **then**
- 9:     **return**  $N.solution$ ;
- 10:   **else**
- 11:     **for**  $C$  in  $N.constraints$  **do**
- 12:        $N' \leftarrow N$ ;
- 13:        $N'.constraints \leftarrow \text{conflict robot constraints}$ ;
- 14:        $N'.solution \leftarrow \text{Lower-level\_search}$ ;
- 15:       **if** ( $N'.solution \neq \emptyset$ ) **then**
- 16:          $OPEN \leftarrow N'$ ;
- 17:       **end if**
- 18:     **end for**
- 19:   **end if**
- 20: **end while**

---

At the lower level, we integrate A\* with robot kinematics, using motion primitives as state nodes while considering mode-switching costs. The  $g$ ,  $h$ , and  $f$  functions retain the traditional A\* structure but incorporate kinematic constraints across different motion modes. In the cost function  $g$ , steering, and mode-switching penalties are introduced. For  $h$ , we use heuristic costs that account for non-holonomic constraints in obstacle-free areas and holonomic constraints when obstacles are present. The heuristic function holistically considers the Reeds-Shepp path length at the endpoint, the Manhattan distance to the goal, and the shortest A\* path from the current node. Additionally, to minimize mode switching and driving space, the orientation difference between the current position and the goal is included in  $h$ . The high-level stage

**Table 1**  
Algorithm parameter setting.

Parameters	Value
The number of particles $N$	30
The maximum number of iterations $T$	200
The maximum value of learning factors $c_{max}$	2.5
The minimum value of learning factors $c_{min}$	0.5
The maximum value of inertia weight $\omega_{max}$	1
The minimum value of inertia weight $\omega_{min}$	0
The particle aggregation coefficient $\gamma$	0.2
The turning penalty factor $\lambda_{turn}$	2
The motion mode switch penalty factor $\lambda_{switch}$	3

detects conflicts in planned paths and iteratively imposes constraints for replanning until all paths are conflict-free.

Building on this, the algorithm checks whether the current node is near the endpoint. If so, it generates rectilinear and transverse Reeds-Shepp curves to compute a path to the endpoint, ignoring static obstacles. If the generated path is collision-free, it is considered feasible. To enhance path-solving speed in confined spaces, the algorithm introduces an adaptive heuristic weight  $\xi$ , which adjusts based on the robot's position relative to the target. This balances the trade-off between optimality and computational efficiency. The calculation formula is as follows:

$$g(s) = g(s_f) + \lambda_t \cdot d(s_f, s) + \lambda_{sw} \cdot d(s_f, s), \quad (21)$$

$$h(s, s_g) = \max(h_{rs}(s, s_g), h_o(s, s_g), h_{A^*}(s, s_g)) + h_h(s, s_g), \quad (22)$$

$$f(s, s_g) = g(s) + \xi \cdot h(s, s_g), \quad (23)$$

where  $f(s, s_g)$  denotes the total cost from the start to the goal position,  $g(s)$  denotes the accrued cost, and  $h(s, s_g)$  denotes the estimated cost to the goal;  $g(s_f)$  is the parent's cumulative cost,  $d(s_f, s)$  denotes the distance cost to the parent,  $\lambda_t$  and  $\lambda_{sw}$  denote the turning and mode-switch penalties respectively, while the heuristics  $h_{rs}(s, s_g)$ ,  $h_o(s, s_g)$ ,  $h_{A^*}(s, s_g)$ , and  $h_h(s, s_g)$  correspond to the Reeds-Shepp curve length, Manhattan distance, A\* distance, and orientation cost, with the weight  $\xi$  balancing optimality ( $\xi = 1$ ) against computational speed ( $\xi > 1$ ).

### 4. Simulation

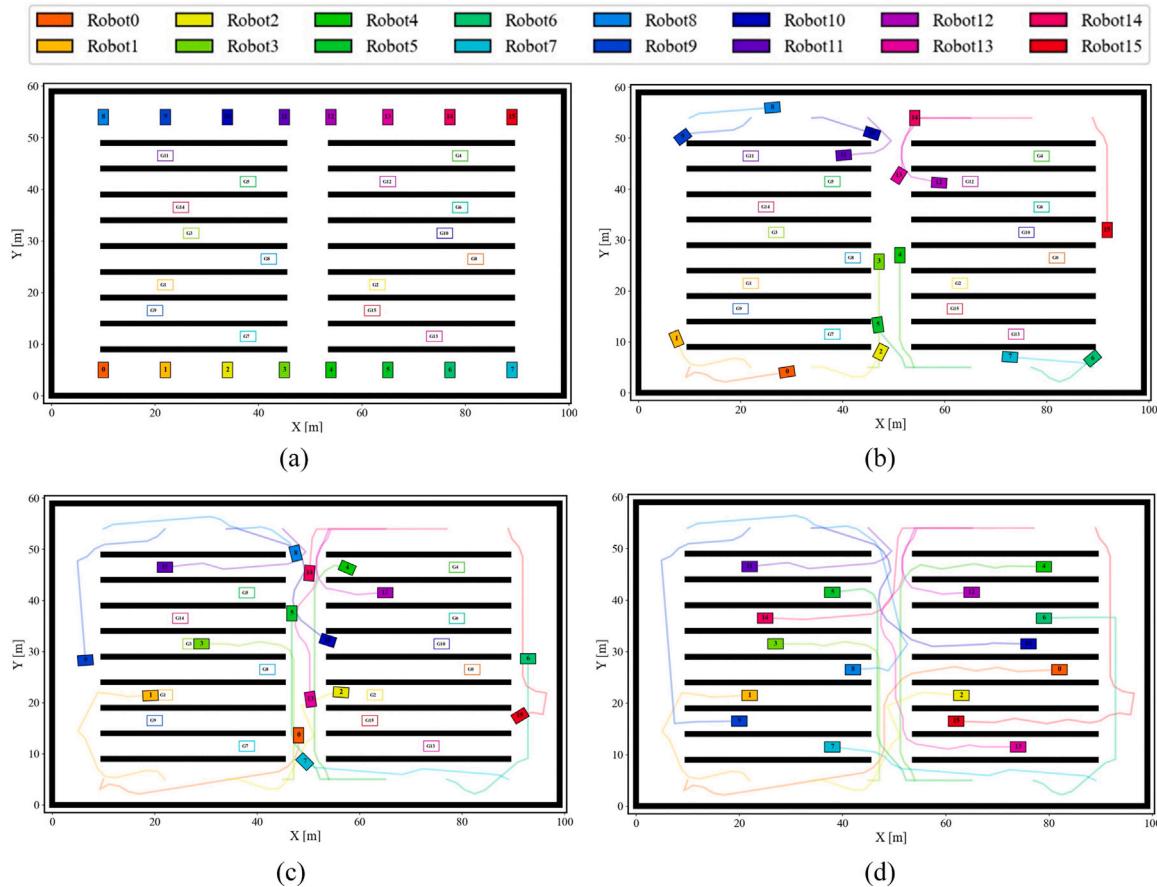
To validate our proposed algorithm, we conducted a simulation in a confined warehouse environment to evaluate both the hierarchical scheduling algorithms and to examine the impact of robot and task quantities on overall operational performance.

#### 4.1. Implementation details

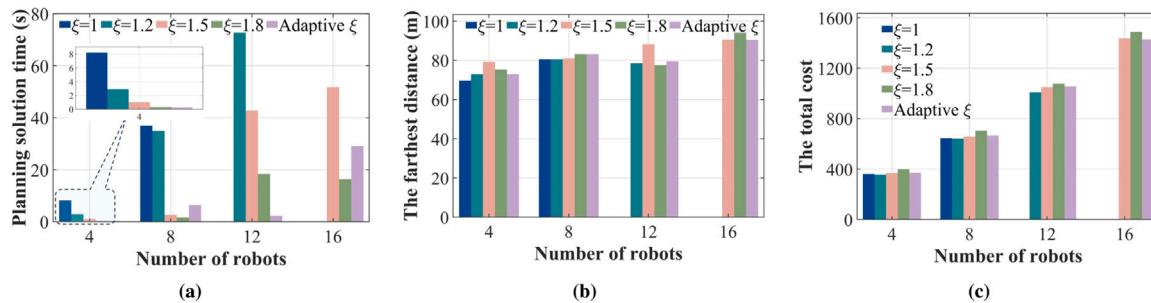
In our study, we designed the experimental environment illustrated in Fig. 5. The scene spans 100 m by 60 m and features 18 shelf obstacles, each measuring 35 m by 1.5 m. The robots have dimensions of 2 m by 3 m and a minimum turning radius of 3 m. The algorithm parameters were determined based on real-world conditions (see Table 1). Simulations were conducted on a Ubuntu 20.04 PC with an AMD Ryzen 7 3700X processor and 16 GB of RAM. Additionally, we recorded the average and optimal fitness values during the task assignment process, the farthest, average, and total travel distances, and the total cost during the path planning process.

#### 4.2. Path planning algorithm verification

To evaluate the effectiveness of our path-planning method, we conducted experiments with 4, 8, 12, and 16 robots. In each experiment, the robots were deployed from both sides of a narrow corridor, with target positions randomly assigned along its midpoint. Given



**Fig. 5.** Path planning results for 16 robots. (a) Position status of 16 robots at 0 s (robots are represented by filled color squares, and their corresponding target positions are hollow boxes of the same color with target numbers labeled). (b) Position status of 16 robots at 8 s. (c) Position status of 16 robots at 18 s. (d) Position status of 16 robots at 36 s.



**Fig. 6.** Simulation results for path planning under different scenarios. (a) Solution time for path planning. (b) The farthest travel distance. (c) Total cost of path planning.

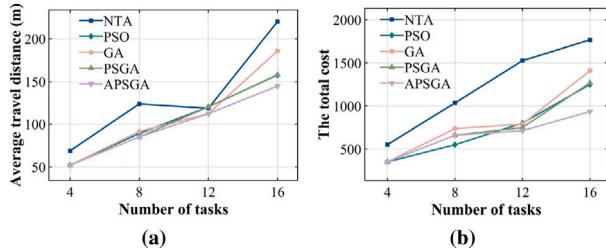
the specificity of the application scenario, research on 4WIDS robots incorporating kinematic constraints is relatively scarce. Consequently, we focused on testing various heuristic weight factors. Each scenario was repeated 30 times. Fig. 5 displays the planned paths for 16 robots, while Fig. 6 illustrates the simulation results across different scenarios.

As shown in Fig. 5, 16 robots start from their initial positions, navigate around obstacles and other robots, and employ diverse motion modes to successfully reach their designated targets. Throughout this process, no collisions or deadlocks occurred, demonstrating the effectiveness of the path-planning solution for multi-robot navigation in a confined space. However, due to the heuristic weight factors used to accelerate the solution process, some paths deviate from the optimal trajectory.

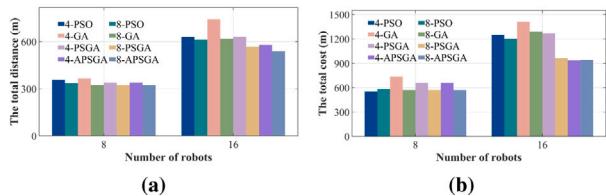
As shown in Fig. 6, solution time and total cost rise significantly as the number of robots increases. For scenarios with 12 and 16 robots, an optimal complete solution cannot be obtained within 100 s in a confined environment. Since target positions are randomly assigned, the number of robots minimizes the maximum travel distance. However, total cost varies considerably depending on the heuristic weight factors. Smaller weights keep the total cost close to the optimal solution, while larger weights increase costs but significantly reduce solution time. The proposed adaptive heuristic weight method sacrifices a small amount of solution quality to reduce runtime, substantially balancing cost and efficiency. For instance, in the scenario with 8 robots, the total cost increases by only 3.2%, while the solution time decreases by 82.5% compared to the optimal complete solution, demonstrating

**Table 2**  
Simulation results of the hierarchical scheduling algorithm under different scenarios.

Scenarios (robots-tasks)	Method	Average fitness	Optimal fitness	The farthest travel distance (m)	Average travel distance (m)	Total travel distance (m)	Total cost
4-12	NTA	—	—	189.63	164.77	659.08	1526.28
	PSO	587.25	549.56	124.92	119.24	476.98	801.03
	GA	620.10	556.56	<b>122.61</b>	118.58	474.32	789.90
	PSGA	560.47	<b>546.56</b>	134.93	120.21	480.85	749.83
	APSGA	<b>557.14</b>	<b>546.56</b>	132.24	<b>112.21</b>	<b>448.82</b>	<b>713.57</b>
4-16	NTA	—	—	262.97	220.14	880.55	1767.01
	PSO	833.02	784.84	184.40	157.61	630.45	1249.32
	GA	858.72	820.56	196.06	185.92	743.66	1409.43
	PSGA	752.03	<b>699.57</b>	<b>164.35</b>	158.02	632.10	1267.73
	APSGA	<b>746.10</b>	<b>699.57</b>	170.41	<b>144.69</b>	<b>578.76</b>	<b>934.59</b>
8-16	NTA	—	—	148.92	99.33	794.64	1642.44
	PSO	728.20	629.85	96.76	76.65	613.20	1200.89
	GA	745.98	659.85	100.52	77.34	618.72	1287.89
	PSGA	642.60	<b>599.85</b>	88.10	70.87	566.96	960.95
	APSGA	<b>632.63</b>	<b>599.85</b>	<b>78.40</b>	<b>67.40</b>	<b>539.20</b>	<b>941.61</b>



**Fig. 7.** Simulation results of different algorithms for the same robots performing different tasks. (a) The result of average travel distance. (c) The result of the total cost.



**Fig. 8.** Simulation results of different algorithms for different robots performing the same tasks. (a) The result of average travel distance. (c) The result of the total cost.

the effectiveness of the adaptive heuristic weight approach in complex confined environments.

#### 4.3. Hierarchical scheduling algorithm verification

To validate the effectiveness of the scheduling method, simulations were conducted with 4 and 8 robots performing multiple tasks in a predefined environment. We compared five approaches: no-task assignment (NTA), PSO, GA, PSGA, and our APSGA, all integrated with the proposed path planning algorithm. To reduce the influence of algorithmic randomness, each method was tested 20 times, with one solution randomly selected for path planning. The simulation results are presented in Table 2.

As shown in Table 2, a well-designed task assignment strategy significantly enhances various performance metrics compared to random task assignment. In the complex scenario we designed, neither PSO nor GA achieved the optimal solution within 20 trials. However, by combining PSO's fast convergence with GA's ability to escape local optima, PSGA successfully identified the optimal solution within 20 experiments. Moreover, APSGA, incorporating adaptive parameter tuning, further improves solution quality. Compared to PSO, GA, and PSGA,

APSGA reduces the average fitness value by 9.6%, 12.8%, and 1.0%, respectively, across 20 trials. This indicates that APSGA is the most effective at obtaining optimal solutions, ensuring a more efficient and high-quality task assignment strategy. Based on the assigned tasks, the proposed hierarchical scheduling algorithm demonstrates substantial advantages in travel distance and total cost during route execution. Notably, it achieves significant cost reductions, lowering the average total cost by 47.7%, 19.2%, 23.4%, and 11.0% compared to NTA, PSO, GA, and PSGA, respectively.

#### 4.4. Impact of robot and task quantity on operational performance

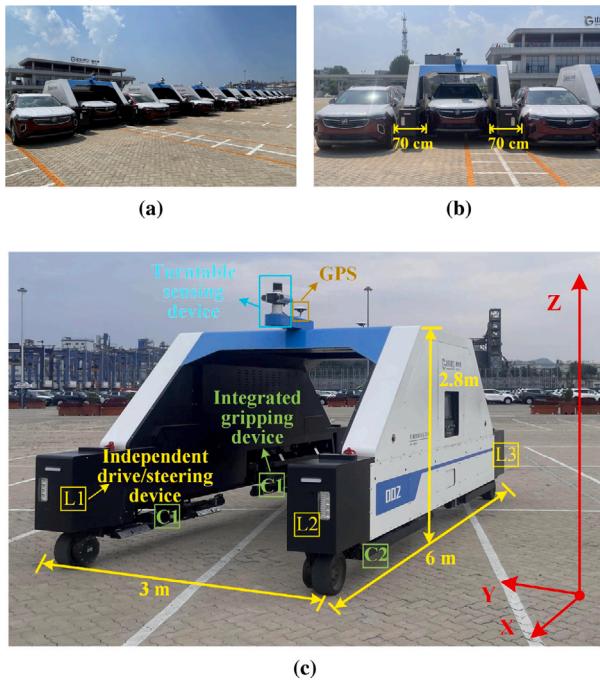
To further validate the advancement of our algorithm in managing varying numbers of robots and tasks, we designed scenarios in which robots executed different task quantities and distributed the same number of tasks among different robots. Relevant data were recorded and shown in Figs. 7 and 8.

Performance indicators for 4 robots executing tasks are shown in Fig. 7. As the number of tasks increases, the average travel distance and total cost also rise. Beyond evaluating task assignment schemes, these metrics reflect mode switching, detours, and waiting times due to path conflicts. When executing 8 tasks, the PSO-based scheme achieves the lowest total cost, while for 12 tasks, the GA-based scheme shows an average travel distance comparable to APSGA. All approaches significantly outperform scenarios without task assignment. Notably, APSGA almost consistently delivers the best results. When the number of tasks increases from 4 to 16, the total cost savings of the proposed method over random assignment rise from 36.3% to 47.1%, demonstrating that APSGA's advantage becomes more significant as task complexity increases.

With 8 and 16 tasks assigned to varying numbers of robots, the simulation results are shown in Fig. 8. In simpler scenarios, increasing the number of robots reduces total cost. However, as task complexity grows, the total cost for 4 robots handling 16 tasks becomes comparable to 8 robots performing the same tasks. This is due to congestion in confined spaces, which efficient task assignments can mitigate but not fully eliminate. Therefore, while a moderate increase in robots enhances efficiency and resource utilization, potential conflicts and congestion must be carefully managed.

#### 5. Experimental

To validate our multi-robot scheduling framework, we performed an engineering implementation of the proposed algorithm and conducted real-world tests at the Ro-Ro terminal of Yantai Port. The high-density parking configuration (30 cm front-rear and 70 cm lateral spacing) severely restricts mobility. The 4WIDS robots, designed for high-speed and heavy-load operations, enable flexible movement in confined spaces, making them well-suited for this environment.



**Fig. 9.** Experimental conditions. (a) Port actual scene. (b) The robot straddles vehicles. (c) Transfer robot.

**Table 3**  
Motion parameters of transfer robot.

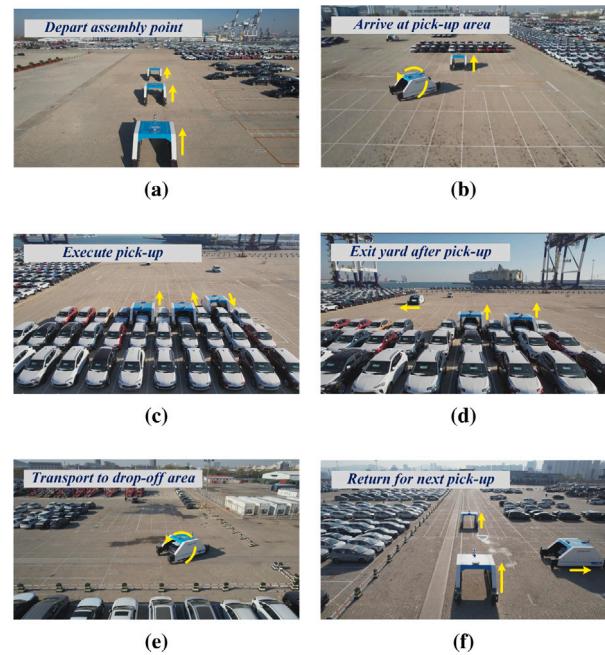
Description	Values
Robot weight	6 t
Robot length × width × height	6 m × 3 m × 2.8 m
Robot wheelbase	5.43 m
Maximum traveling speed	25 km/h
Maximum turning angular velocity	0.57 rad/s
Maximum acceleration	2.31 m/s <sup>2</sup>
Maximum deceleration	1.16 m/s <sup>2</sup>
Robot steering angle	360°

### 5.1. Experimental condition

The autonomous transfer system for Ro-Ro terminal commercial vehicles comprises a port production task control system, a cloud scheduling system, and a robot execution system. The port production task control system oversees port operations and formulates transfer tasks, which the cloud scheduling system processes to allocate tasks and generate paths for multiple robots. The robot execution system then follows these paths to transfer vehicles efficiently.

Our method was deployed on the cloud scheduling system running on two high-performance servers (32-core CPUs, 128 GB RAM, 10 TB storage), with a modular design ensuring seamless integration with the robot execution layer. Due to ongoing production tasks at the site, testing faced constraints such as manual scheduling, safety protocols, and equipment coordination, where even minor errors could cause costly collisions. Therefore, we conducted only the engineering implementation and on-site testing of the proposed algorithm. To ensure stable system operation and safety in real-world conditions, the deployed version adopts a more conservative obstacle avoidance strategy, restricts vehicle motion to a predefined safe road network, and takes into account the actual operational characteristics of the robots.

The robot execution system includes a high-load electric straddle transfer robot with 4WIDS and clamping capabilities. It also features



**Fig. 10.** Experimental process of the transport robot. (a) Departure from the assembly point. (b) Sequential arrival at the yard pick-up area. (c) Execution of the pick-up operation. (d) Exit from the yard after vehicle pick-up. (e) Transport to the drop-off area for vehicle delivery. (f) Return to the pick-up area for the next operation.

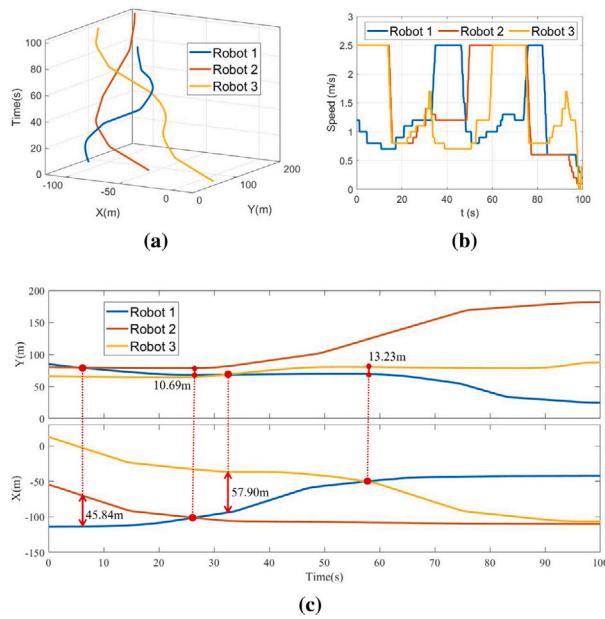
a turntable sensing system (LiDAR and a visible-light camera) and a global positioning system. The experimental environment is shown in Fig. 9, with its robot's basic motion parameters listed in Table 3.

### 5.2. Experimental scenario and result analysis

We used a pre-established road network map to verify the hierarchical scheduling algorithm's effectiveness. The cloud scheduling system assigned tasks planned paths and coordinated three transfer robots to transfer commercial vehicles between two yards (200 m east–west, 150 m north–south). The experimental process is shown in Fig. 10.

As shown in Fig. 10, after receiving tasks from the port production task control system, the three transfer robots depart from the assembly point pick-up area, retrieve vehicles, and proceed to the drop-off area before returning for the next cycle. The maximum transit speed is set at 2.5 m/s. No conflicts, collisions, or deadlocks occur throughout the transportation process, ensuring smooth execution as planned by the cloud scheduling system. To further evaluate coordination, we analyze a scenario where all three robots converge at an intersection, considering factors such as priority and speed adjustments, with trajectories and speeds depicted in Fig. 11.

Figs. 11(a) and 11(c) illustrate the spatiotemporal trajectories of three robots, with four instances of alignment along the X or Y axis. At these points, the minimum distances along the orthogonal axis were 45.84 m, 10.69 m, 57.90 m, and 13.23 m, respectively, ensuring safe separation and no collisions. As shown in Fig. 11(b), velocity profiles indicate that at 20 s and 80 s, two robots briefly slowed to 0.8 m/s when converging at a narrow junction, before accelerating toward their targets after clearing the zone. These results confirm that the cloud-based scheduling system successfully allocated tasks from the port's production control system and generated safe, collision-free trajectories well-suited to real-world constraints. All robots accurately followed their planned paths and completed tasks, validating the proposed framework's practical effectiveness.



**Fig. 11.** Response curves of three robots at the same intersection under hierarchical scheduling. (a) Trajectories. (b) Speed curves. (c) X- and Y-coordinate trajectories.

## 6. Conclusion

This paper presents a hierarchical scheduling method for multi-robot systems in complex, confined environments utilizing 4WIDS robots. At the task assignment layer, the adaptive update strategy is combined with GA optimization to improve the classical PSO algorithm and solve the task assignment problem more effectively. At the path planning layer, an improved conflict-aware path planning algorithm is proposed, combining motion mode switching and adaptive heuristic weight factors to accelerate the search process. The proposed method is validated through high-fidelity simulations, analyzing the impact of robot and task quantity on operational performance. Simulation results demonstrate that, compared to NTA, PSO, GA, and PSGA, the proposed approach reduces the average total cost by 47.7%, 19.2%, 23.4%, and 11.0%, respectively, highlighting its effectiveness and superiority. Furthermore, the framework has been tested at the Ro-Ro terminal at Yantai Port, verifying its feasibility in real-world multi-robot systems.

Future work will optimize task assignment by assigning more tasks to specific robots based on task distribution and location, further reducing costs and improving efficiency. Additionally, we plan to integrate distributed multi-robot navigation methods to enable real-time trajectory replanning, allowing robots to dynamically adapt to unknown environmental obstacles.

## CRediT authorship contribution statement

**Lin Zhang:** Writing – original draft, Visualization, Validation, Methodology, Formal analysis. **Yichen An:** Writing – original draft, Software, Investigation. **Tianwei Niu:** Writing – review & editing, Visualization. **Runjiao Bao:** Visualization, Validation, Conceptualization. **Shoukun Wang:** Writing – review & editing, Funding acquisition. **Junzheng Wang:** Writing – review & editing, Conceptualization.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 62473044) and the BIT Research and Innovation Promoting Project (No. 2024YCXZ007).

## Data availability

Data will be made available on request.

## References

- Allgor, Russell, Cezik, Tolga, & Chen, Daniel (2023). Algorithm for robotic picking in amazon fulfillment centers enables humans and robots to work together effectively. *INFORMS Journal on Applied Analytics*, 53(4), 266–282.
- Bhargava, Ankur, Suhail, Mohd, & Singholi, Ajay S. (2024). A review of recent advances, techniques, and control algorithms for automated guided vehicle systems. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 46(7), 419.
- Bolognini, Michele, Fagiano, Lorenzo, & Limongelli, Maria Pina (2023). A fault-tolerant automatic mission planner for a fleet of aerial vehicles. *Control Engineering Practice*, 135, Article 105501.
- Chen, Xuchao, He, Shiwei, Zhang, Yongxiang, Tong, Lu Carol, Shang, Pan, & Zhou, Xuesong (2020). Yard crane and AGV scheduling in automated container terminal: A multi-robot task allocation framework. *Transportation Research Part C: Emerging Technologies*, 114, 241–271.
- Chen, Rui, Zhu, Xinyu, Yuan, Zean, Pu, Huayan, Luo, Jun, & Sun, Yu (2024). A bioinspired single actuator-driven soft robot capable of multi-strategy locomotion. *IEEE Transactions on Robotics*.
- Gao, Guohua, Li, Dongjian, Liu, Kai, Ge, Yuxin, & Song, Chunxu (2024). A study on path-planning algorithm for a multi-section continuum robot in confined multi-obstacle environments. *Robotica*, 42(10), 3324–3347.
- Gao, Shan, Zuo, Lei, Lu, Xiaofei, & Tang, Bo (2025). Cooperative target allocation for heterogeneous agent models using a matrix-encoding genetic algorithm. *Journal of Information and Intelligence*, 3(2), 154–172.
- Geng, Rongmei, Ji, Renxin, & Zi, Shuanjin (2022). Research on task allocation of UAV cluster based on particle swarm quantization algorithm. *Mathematical Biosciences and Engineering*, 20, 18–33.
- Hiraoka, Naoki, Ishida, Hirokazu, Hiraoka, Takuma, Kojima, Kunio, Okada, Kei, & Inaba, Masayuki (2024). Sampling-based global path planning using convex polytope approximation for narrow collision-free space of humanoid. *International Journal of Humanoid Robotics*, Article 2450005.
- Huo, Lin, Mao, Jianlin, San, Hongjun, Li, Ruiqi, & Zhang, Shufan (2025). Deep reinforcement learning of group consciousness for multi-robot pathfinding. *Engineering Applications of Artificial Intelligence*, 155, Article 110978.
- Li, Han, Chen, Peng, Yu, Guizhen, Zhou, Bin, Han, Zhixuan, & Liao, Yaping (2023). Collaborative trajectory planning for autonomous mining trucks: a grouping and prioritized optimization based approach. *IEEE Transactions on Vehicular Technology*, 73(5), 6283–6300.
- Li, Jiaoyang, Tinka, Andrew, Kiesel, Scott, Durham, Joseph W, Kumar, TK Satish, & Koenig, Sven (2021). Lifelong multi-agent path finding in large-scale warehouses. Vol. 35, In *Proceedings of the AAAI conference on artificial intelligence* (pp. 11272–11281).
- Ma, Changjia, Han, Zhichao, Zhang, Tingrui, Wang, Jingping, Xu, Long, Li, Chengyang, et al. (2023). Decentralized planning for car-like robotic swarm in cluttered environments. In *2023 IEEE/RSJ international conference on intelligent robots and systems* (pp. 9293–9300). IEEE.
- Mao, Xiao, Wu, Guohua, Fan, Mingfeng, Cao, Zhiguang, & Pedrycz, Witold (2024). DL-DRL: A double-level deep reinforcement learning approach for large-scale task scheduling of multi-UAV. *IEEE Transactions on Automation Science and Engineering*, 22, 1028–1044.
- McBeth, Courtney, Motes, James, Uwacu, Diane, Morales, Marco, & Amato, Nancy M (2023). Scalable multi-robot motion planning for congested environments with topological guidance. *IEEE Robotics and Automation Letters*.
- Moldagaleeva, Akmral, Ortiz-Haro, Joaquin, Toussaint, Marc, & Höning, Wolfgang (2024). Db-cbs: Discontinuity-bounded conflict-based search for multi-robot kinodynamic motion planning. In *2024 IEEE international conference on robotics and automation* (pp. 14569–14575). IEEE.
- Peng, Jih-Sien, Ngo, Van-Tam, & Liu, Yen-Chen (2023). Toward distributed control of networked omnidirectional automated ground vehicles in complex environment with delay communication. *Control Engineering Practice*, 134, Article 105464.
- Ruan, Sipu, Poblete, Karen L, Wu, Hongtao, Ma, Qianli, & Chirikjian, Gregory S (2022). Efficient path planning in narrow passages for robots with ellipsoidal components. *IEEE Transactions on Robotics*, 39(1), 110–127.
- Su, Jian-li, & Wang, Hua (2021). An improved adaptive differential evolution algorithm for single unmanned aerial vehicle multitasking. *Defence Technology*, 17(6), 1967–1975.

- Sun, Haowen, Chen, Ming, Pan, Yijin, Cang, Yihan, Zhao, Jiahui, & Sun, Yuanzhi (2024). Deep reinforcement learning empowered trajectory and resource allocation optimization for UAV-assisted MEC systems. *IEEE Wireless Communications Letters*.
- Sun, Qiankun, Liu, Weifeng, & Cai, Lei (2025). Multi-dynamic target coverage tracking control strategy based on multi-UAV collaboration. *Control Engineering Practice*, 155, Article 106170.
- Tan, Yifang, Zhou, Chao, & Qian, Feng (2024). Cooperative task allocation method for multi-unmanned aerial vehicles based on the modified genetic algorithm. *IET Intelligent Transport Systems*, 18(6), 1164–1173.
- Wang, Ming, Zhang, Peng, Zhang, Guoqing, Sun, Kexin, Zhang, Jie, & Jin, Mengyu (2025). A resilient scheduling framework for multi-robot multi-station welding flow shop scheduling against robot failures. *Robotics and Computer-Integrated Manufacturing*, 91, Article 102835.
- Wu, Quanwei, Wang, Xiangyu, & Qiu, Xuechao (2024). Embedded technique-based formation control of multiple wheeled mobile robots with application to cooperative transportation. *Control Engineering Practice*, 150, Article 106002.
- Xie, Wei, Yu, Gan, Cabecinhas, David, Silvestre, Carlos, Zhang, Weidong, & He, Wei (2024). Robust collision-free formation control of quadrotor fleets: Trajectory generation and tracking with experimental validation. *Control Engineering Practice*, 145, Article 105842.
- Xu, Gang, Kang, Xiao, Yang, Helei, Wu, Yuchen, Liu, Weiwei, Cao, Junjie, et al. (2023). Distributed multi-vehicle task assignment and motion planning in dense environments. *IEEE Transactions on Automation Science and Engineering*.
- Xu, Yongkang, Zhang, Lin, Liu, Zhi, Wang, Shoukun, & Wang, Junzheng (2025). Design and development of a new autonomous transportation robot for finished vehicles docking transportation in RO/RO logistics terminal. *Advanced Engineering Informatics*, 66, Article 103391.
- Yan, Fei, Chu, Jing, Hu, Jinwen, & Zhu, Xiaoping (2024). Cooperative task allocation with simultaneous arrival and resource constraint for multi-UAV using a genetic algorithm. *Expert Systems with Applications*, 245, Article 123023.
- Yang, Yongsheng, He, Sha, & Sun, Shu (2023). Research on the cooperative scheduling of arangs and agvs in a sea-rail automated container terminal under the rail-in-port model. *Journal of Marine Science and Engineering*, 11(3), 557.
- Zang, Zheng, Song, Jiarui, Lu, Yaomin, Zhang, Xi, Tan, Yingqi, Ju, Zhiyang, et al. (2023). A unified framework integrating trajectory planning and motion optimization based on spatio-temporal safety corridor for multiple AGVs. *IEEE Transactions on Intelligent Vehicles*.
- Zhai, Shaobo, Li, Guangwen, Wu, Guo, Hou, Mingshan, & Jia, Qiuling (2023). Cooperative task allocation for multi heterogeneous aerial vehicles using particle swarm optimization algorithm and entropy weight method. *Applied Soft Computing*, 148, Article 110918.
- Zhang, Lin, Xu, Yongkang, Si, Jing, Bao, Runjiao, An, Yichen, Wang, Shoukun, et al. (2025). Autonomous transfer robot system for commercial vehicles at ro-ro terminals. *Expert Systems with Applications*, 289, Article 128347.