Mathematical Biosciences and Engineering

MBE, 20(2): 3528–3565.
DOI: 10.3934/mbe.2023165
Received: 19 October 2022
Revised: 14 November 2022
Accepted: 21 November 2022
Published: 06 December 2022

http://www.aimspress.com/journal/MBE

Theory article

Conflict-free and energy-efficient path planning for multi-robots based on priority free ant colony optimization

Ping Li[1,2] and Liwei Yang[1,2,*]

[1] Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650093, China
[2] Faculty of Information Engineering, Xinjiang Institute of Technology, Akesu 843100, China

* **Correspondence:** Email: 18916336783ylw@gmail.com.

Abstract: With the background of limited energy storage of robots and considering the high coupling problem of multi-agent path finding (MAPF), we propose a priority-free ant colony optimization (PFACO) to plan conflict-free and energy-efficient paths, reducing multi-robots motion cost in the rough ground environment. First, a dual-resolution grid map considering obstacles and ground friction factors is designed to model the unstructured rough terrain. Second, an energy-constrained ant colony optimization (ECACO) is proposed to achieve energy-optimal path planning for a single robot, in which we improve the heuristic function based on the combined effects of path length, path smoothness, ground friction coefficient and energy consumption, and consider multiple energy consumption metrics during robot motion to improved pheromone update strategy. Finally, considering multiple collision conflict cases among multiple robots, we incorporate a prioritized conflict-free strategy (PCS) and a route conflict-free strategy (RCS) based on ECACO to achieve MAPF with low-energy and conflict-free in a rough environment. Simulation and experimental results show that ECACO can achieve better energy saving for single robot motion under all three common neighborhood search strategies. PFACO achieves both the conflict-free path and energy-saving planning for robots in complex scenarios, and the study has some reference value for solving practical problems.

Keywords: multi-agent path finding; ant colony optimization; conflict-free; energy-efficient; unstructured rough terrain

1. Introduction

In recent years, with the wide application of intelligent robots in intelligent warehouse management and traffic control, multi-agent path finding (MAPF) as a critical technology in multi-intelligence collaboration has received extensive attention from scholars [1,2]. MAPF technique is to plan conflict-free paths for multiple robots with different tasks under obstacles and multiple constraints, and satisfies some production targets [3,4].

The algorithms for single robot path planning can be classified into three according to their characteristics: search-based, sampling-based and intelligent optimization-based. The typical method based on the search is the A* algorithm, but it is difficult to guarantee the optimality of paths in large-scale complex environments [5]. Lu et al. [6] address the large memory consumption and long running time of traditional A* by improving the heuristic function to reduce search efficiency. A typical method based on sampling is the Rapidly-exploring Random Trees (RRT), but there is a large amount of redundant computation during the traversal process [7]. Therefore, Xu et al. [8] and Ruan et al. [9] proposed a regional sampling RRT based on simplified maps. Zhang et al. [10] proposed an improved RRT based on target constraint sampling and target bias expansion. They [8–10] all address the low execution efficiency of the traditional RRT algorithm. Intelligent optimization algorithms include the ant colony algorithm [11], genetic algorithm [12], sparrow algorithm [13], and so on. In recent years, related scholars have made promising achievements in solving common problems such as slow convergence speed and easy falling into local optimality, which are widely faced in intelligent optimization algorithms. Yang et al. [14] introduced optimal and worst solutions in ant colony pheromone updating to expand the influence of high-quality ants and weaken the pheromones on the path of worst ants, which accelerated the convergence of ACO. Xu et al. [15] proposed a two-level ant colony algorithm with mutual collaboration, using external ant colonies for global search and internal ant colonies for local search, improving the algorithm's solution accuracy. Li et al. [16] proposed an improved forward search optimization planning path. This method can shorten the global path length by jumping the node search way and combined with an improved artificial potential field to smooth that path. Experimental results show that this method keeps the path at a safe and comfortable distance from obstacles compared to the traditional strategy, which is more suitable for single robot motion.

MAPF contains the research mentioned earlier on single-robot obstacle avoidance and path-optimal metrics (path length, smoothness, safety, energy consumption, etc.), more importantly, considers more complex motion conflict situations between robots. It has received much attention from scholars as a classical NP-Hard problem [17]. Lin et al. [18] reviewed the relevant research of MAPF over the years and have made a more systematic summary and generalization, classifying the planning algorithms into classical, heuristic, bio-inspired, and artificial intelligence methods and classifying the decision-making methods into centralized and decentralized. Cheng et al. [19] proposed an enhanced heuristic A* algorithm for solving the path-planning problem of large parking lots. Ren et al. [20] proposed an MS* exact algorithm, which can solve the optimal conflict-free path for 20 robots performing 50 tasks. Wen et al. [21] proposed a MAPF-POST algorithm based on the characteristics of robot motion in the actual scene. It reduces the conflict between robot paths and improves search efficiency. Xin et al. [22] proposed a solution to the problems of high computational complexity and difficulty in local planning for collision-free path planning of AGVs based on a Time-space network model. The global planning problem is decomposed into more minor local planning problems to improve computational efficiency, newly defined decision variables and constraints are introduced to improve the planning capability, and extensive case studies verify the feasibility. Murakami [23]

considered scheduling and conflict-free paths for capacity-constrained AGV systems (robot capacity constraint and buffer capacity constraint) in a spatiotemporal network model. The proposed method enables optimal solutions in a wide range of cases.

In order to improve the solution efficiency and planning success rate of MAPF problems, many scholars have applied the priority-based planning method, conflict-based search and traffic rule method in recent years. Cap et al. [24] proposed a class of MAPF algorithms based on Prioritized Planning (PP) and designed several rules for determining the priority. Greshler et al. [25] introduced the cooperative conflict-based search (Co-CBS) to solve task assignments of multiple robots and their path planning. Wang et al. [26] proposed a method combining priority to obtain the path with optimal cost and execution time. Li et al. [27] combined D* Lite and PP algorithms to achieve high search efficiency and optimization of path quality. In order to efficiently achieve conflict-free path planning for swarm robots in more complex environments, Wu et al. [28] embedded priority rules into the improved heuristic algorithm to eliminate robot trajectories with potential conflicts, which has a high success rate and achieves a good balance between maximum completion time and working time objectives. Dewanga et al. [29] proposed a distance-priority-based approach, enabling low-priority robots to avoid collisions by bypassing high-priority robots.

There are promising results regarding the success rate of path solving and efficiency of algorithm execution for MAPF. However, the uncertain natural environmental factors may affect the cost of actual robot motion [30]. Accordingly, an investigation considering energy consumption [31] emerged. Zhang et al. [32] proposed a low energy consumption optimal path planning method based on the improved AD* (Anytime dynamic A*) algorithm, which incorporates distance and energy consumption cost into the evaluation function of the search nodes to search for the energy consumption optimal path. Mei et al. [33] divide the energy-optimal path planning process into two layers. First, a series of paths are generated by traditional path planning methods, such as the heuristic search [34] and RRT algorithms [35]. Then calculating these paths' consumption according to the energy consumption model in the path optimization process, an energy-efficient path can obtain.

In summary, most research on MAPF has focused on path planning success rate, algorithm execution efficiency, and so on as optimization objectives. Therefore, Considering the performance advantages of ACO in multi-objective planning paths have been verified and demonstrated by many scholars [36–38]. With the background of limited stored energy and the rough motion environment of the robot, we conducted a conflict-free and energy-efficient MAPF study based on an improved multi-objective ACO. The main contributions are as follows:

• A dual-resolution raster map that considers obstacle and ground friction factors is designed to model the unstructured warehouse environment.

• The energy-constrained ant colony optimization (ECACO) for a single mobile robot is proposed based on the excellent robustness and iterative evolution capability of the ant colony algorithm in path planning, which contains the heuristic function by energy-constrained improvement and the pheromone update strategy by energy model optimization. The improved heuristic function considers the path length, path smoothness, road friction coefficient, and path energy consumption. The enhanced pheromone update strategy considers the transformed kinetic energy of robot motion, the energy consumption to overcome the loss of traction resistance, and the energy consumption of robot hardware.

• The priority conflict-free strategy (PCS) and route conflict-free strategy (RCS) are incorporated in ECACO. We named it priority-free ant colony optimization (PFACO) and applied it to the multi-warehouse robot systems. The PCS mainly solves the deadlock conflicts of multiple robots by switching the priority between conflicting robots and re-route planning to achieve the purpose of

conflict resolution. The RCS mainly solves node conflict, counterpoint conflict, occupancy conflict, and blockage conflict among multiple robots based on waiting in place or re-route planning.

The rest of this paper is organized as follows: Section 2 classifies MAPF conflicts and proposes a 2.5D environment modeling method; Section 3 presents ECACO for single-robot energy-efficient path planning; Section 4 presents the multi-robot energy-efficient planning PFACO. Section 5 shows the simulation experiments and analysis; Section 6 provides a summary and the future work plan.

2. Problem description

2.1. Multi-agent path finding (MAPF)

The MAPF problem is defined as robots starting from their positions, planning paths to reach their separate target points, and eliminating spatiotemporal conflicts in the robot paths as much as possible. Based on conflict-free, optimizing the robot group path's energy consumption is the optimization goal in this paper.

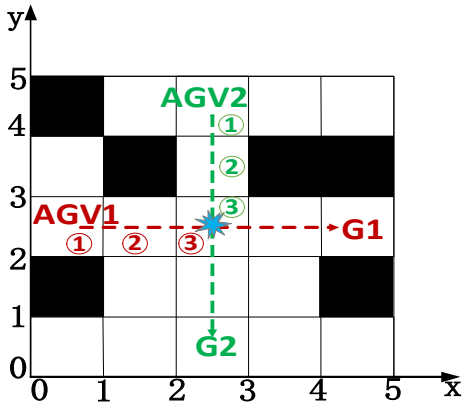


Figure 1. Node conflict.

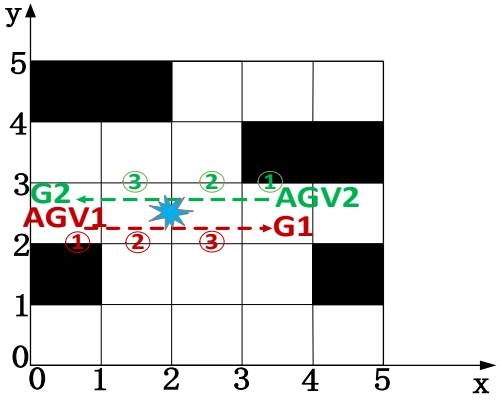


Figure 2. Type of counterpoint conflict.

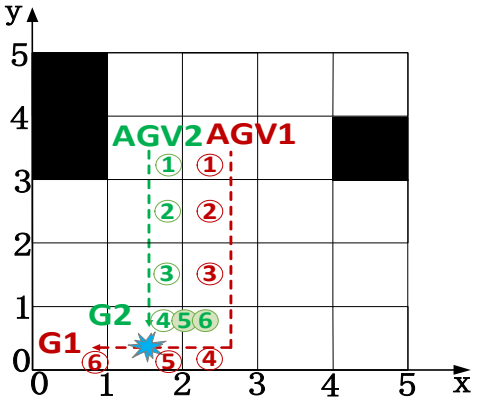
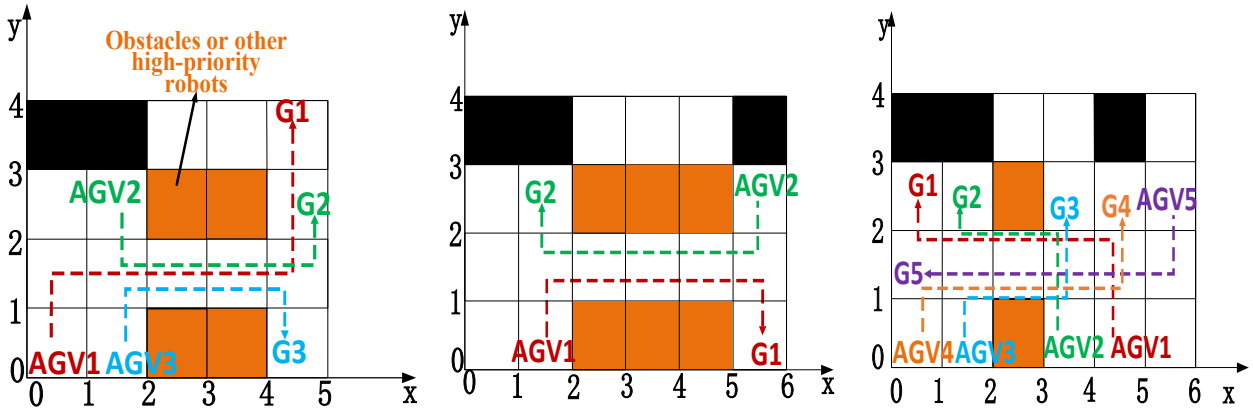


Figure 3. Types of occupancy conflicts.

where, the white grid indicates the free passable area, the black grid indicates the obstacle area, and the yellow grid indicates the free area always occupied by other high-priority robots.



(a) Blockage in the same direction of motion.

(b) Non-co-motion blockage.

(c) Complex blockage.

Figure 4. Narrow terrain obstruction type.

Considering the PCS and map scenarios adopted in this paper, the conflict types are divided into two categories: typical MAPF conflicts (node conflict and counterpoint conflict) and priority MAPF conflicts (occupancy conflict and blockage conflict), which are defined as follows：

1) Node conflict: Two robots will occupy the same node position in the next moment. As shown in Figure 1, AGV1 and AGV2 will collide at moment Step 3 on map (2.5, 2.5).

2) Counterpoint conflict: Two robots traveling in opposite directions and exchanging positions with each other at the next moment. As shown in Figure 2, at the moment of Step 3, AGV1 will move from the map (1.5, 2.5) to map (2.5, 2.5), AGV2 will move from the map (2.5, 2.5) to map (1.5, 2.5), and AGV1 and AGV2 will collide in the exchange of positions.

3) Occupancy conflict: It means that in the priority MAPF (the principle is that the low-priority robot avoids the high-priority robot during its motion), as shown in Figure 3, the low-priority AGV2 stops more advance than the high-priority AGV1. The target point of AGV2 is located precisely on the

node that AGV1 has not yet passed, and it cannot effectively avoid AGV1 later because the task is over.

4) Blockage conflict: We define it as a highly coupled local dynamic scenario that allows only one robot to pass. As shown in Figure 4 (yellow obstacles indicate positions always occupied by other high-priority robots), the congruent and phased movements between AGVs can significantly exacerbate the environment's coupling and limit the performance of conventional algorithms.

2.2. Dual-resolution grid map modeling scheme

Considering the differentiated energy consumption of robot movement on the road surface with different roughness, this paper proposes a dual-resolution 2.5D map-building scheme, a composite map composed of two layers of grid maps. As shown in Figure 5, Maps A and B store the information on the friction coefficient of non-flat ground and obstacle information in the environment, respectively. Where the friction coefficient value of each node in Map A represents the average value of the friction coefficient in a grid range, the nodes in Map B represent white grids that robots can pass freely or black obstacle grids that are forbidden to pass.

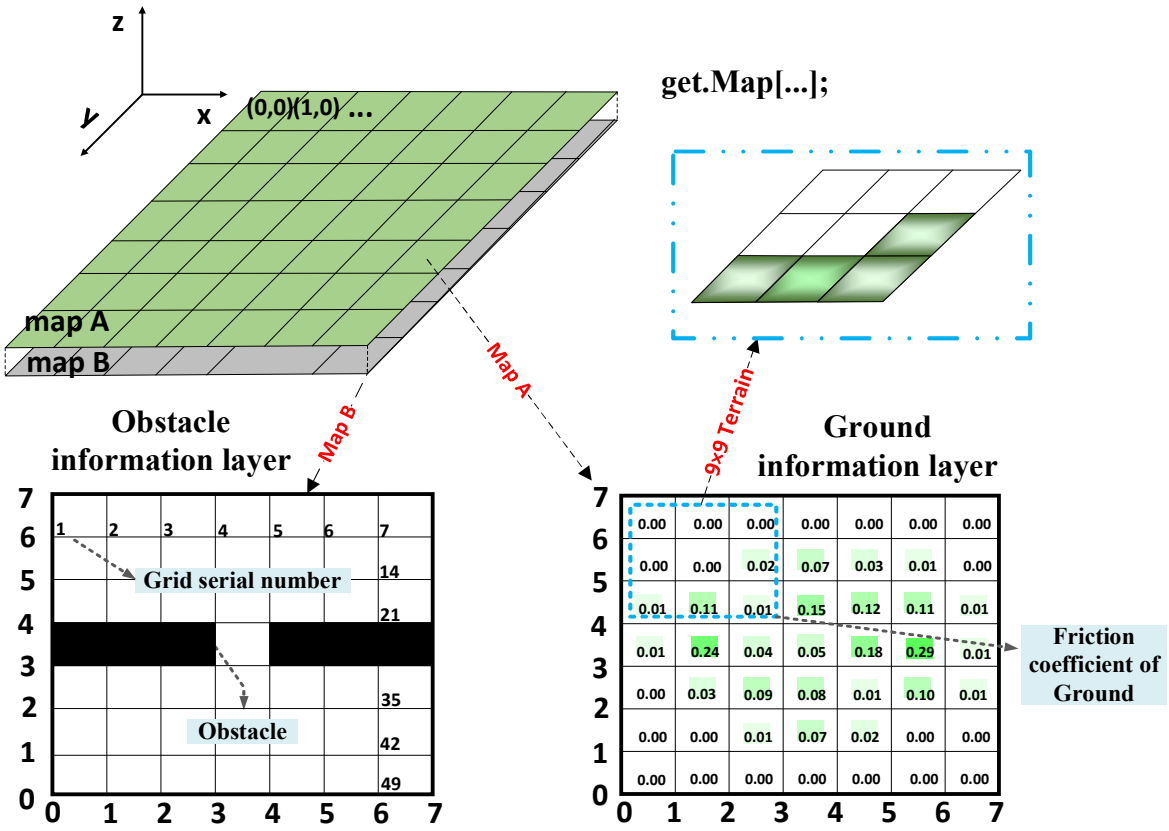


Figure 5. Dual resolution 2.5D map.

In dual-resolution 2.5D, each grid information can be read arbitrarily by any robot. For example, we can use the function '*get. Map (1.5, 1.5)*' to get the ground friction coefficient and obstacle information corresponding to Maps A and B at (1.5, 1.5), respectively. To reduce the computation of the algorithm, we mark the grid order from left to right and from top to bottom, and the relationship

between the grid number i and the coordinates (x, y) is as follows:

$$\begin{cases} x = \begin{cases} mod(i, MM) - 0.5, mod(i, MM) \neq 0 \\ MM + mod(i, MM) - 0.5, \quad otherwise \end{cases} \\ y = MM + 0.5 - ceil(i, MM) \end{cases} \tag{1}$$

where, mod is the remainder operation; $ceil$ is the upward rounding operation; MM is the maximum value of the horizontal axis of the grid map.

3. Ant colony optimization theory

3.1. Traditional ant colony optimization (ACO)

The ant colony algorithm simulates the foraging behavior of a natural ant population, where pheromones and heuristic information determine the probability $p_{i,j}^k$ of how ants choose their next travel direction in grid environments.

$$p_{i,j}^k(t) = \begin{cases} \dfrac{\left[\tau_{i,j}(t)\right]^\alpha \left[\eta_{i,j}^k(t)\right]^\beta}{\sum_{s \subset allowed_i} \left[\tau_{i,j}(t)\right]^\alpha \left[\eta_{i,j}^k(t)\right]^\beta}, j \in allowed_i \\ 0, j \notin allowed_i \end{cases} \tag{2}$$

where, k is the ant number; i and j denote the current position grid and the next transferred grid of ant k, respectively; t denotes the current iterations; α and β denote the expected values of the pheromone concentration $\tau_{i,j}(t)$ and heuristic function $\eta_{i,j}(t)$, respectively; $allowed_i$ denotes the set of selectable lattices of ant k at the grid i. $\eta_{i,j}(t)$ can be expressed as Eq (3).

$$\eta_{i,j}(t) = 1/d \tag{3}$$

where, d is the Euclidean distance from grid i to grid j.

As the iteration proceeds, $\tau_{i,j}(t)$ pheromones will accumulate and volatilize, and the pheromone update model is shown as follows:

$$\tau_{i,j}(t+1) = \rho \tau_{i,j}(t) + \sum_{k=1}^{M} \Delta\tau_{i,j}^k(t) \tag{4}$$

$$\Delta\tau_{i,j}^k(t) = \begin{cases} Q/L_k, \{i, j\} \subset visited_i \\ 0\text{ , }other \end{cases} \tag{5}$$

where, ρ is the pheromone volatility residual; Q is the pheromone constant; L_k is the total length traveled by ant k in this cycle; M is the total number of ants; $visited_i$ is the collection of grids already visited by ant k.

3.2. Energy constrained ant colony optimization (ECACO)

There are three main methods [39–41] to improve the energy utilization of robots: The first is to use high-efficiency drive motors, the second is to improve the electrical energy conversion efficiency of motor drive circuits, and the third is to reduce energy consumption through motion planning. However, available hardware and electrical conditions limit the first two methods. Therefore, we

propose the energy-constrained ant colony optimization (ECACO) to explore an energy-efficient path planning of mobile robots based on the advantages of the iterative evolution of the ACO.

3.2.1. Heuristic function of energy consumption constraint

Based on the unstructured rough environment modeled in 2.5D, the distance of the path, the degree of the path curvature, and the energy consumption caused by the ground roughness factor on the robot motion are the best indicators to evaluate the path. However, the initial undifferentiated distribution of pheromones in ACO is prone to the chaotic nature of ant colony motion. To make the heuristic function with better guidance, we explore a heuristic function combined with an energy consumption constraint to reduce the energy consumption of ants in the search process.

$$\eta_{i,j}(t) = [wL_{i,j}(t) + xT_{i,j}(t) + y\Lambda_{i,j}(t) + zE_{i,j}(t)] * Backcost \tag{6}$$

$$Backcost = \begin{cases} 0.01, if\ \ ant_k\ \ back\ \ up\ \ last\ \ time \\ 1,\quad other \end{cases} \tag{7}$$

where, $L_{i,j}(t)$ is the road evaluation function; $T_{i,j}(t)$ is the smoothing evaluation function; $\Lambda_{i,j}(t)$ is the friction factor evaluation function; $E_{i,j}(t)$ is the energy consumption evaluation function; w, x, y, z are the weight of each function; $Backcost$ is the fallback suppression factor.

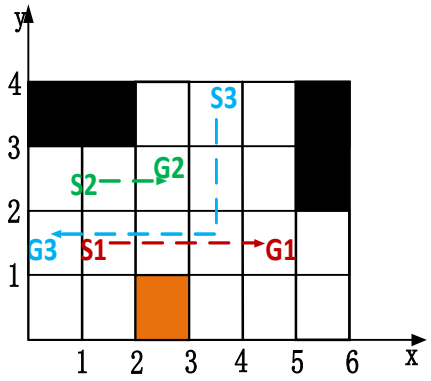


Figure 6. Blockage situation.

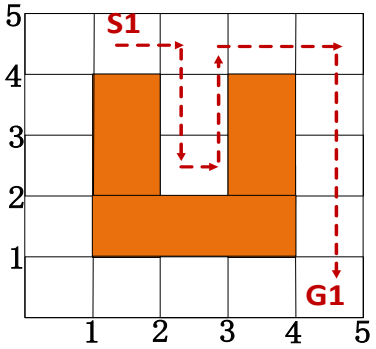


Figure 7. Fallback strategy.

Considering the large-scale multi-robot scenario, ants will be given a large number of node

conflict constraints. Moreover, there is a high possibility of blockage conflict, as shown in Figure 6, under the obstacle environment constraint. To relieve such situations, we introduce a fallback strategy [42] based on the traditional ACO. The effect of this strategy is illustrated in Figure 7, where ants return to the original path when a zero suitable node is available to follow. For the purpose of preventing unnecessary backtracking of ants resulting in extra energy consumption, we use the suppression factor $Backcost$ to reward and penalize the heuristic function.

1) Road evaluation function

The conventional ACO uses the distance factor of Eq (3) to evaluate the Euclidean distance from the ant's currently available grid that can be selected to the target grid. Since the difference in this distance is not apparent, we modified the distance evaluation function with the following equation to reflect the attractiveness of the target grid.

$$L_{i,j}(t) = \frac{D_{max} - d(j,E)}{D_{max} - D_{min} + 0.01} \tag{8}$$

where, D_{max} and D_{min} denote the maximum and minimum Euclidean distance from the neighboring grid to the endpoint E, respectively; $d(j,E)$ is the Euclidean distance from the neighboring grid j to the endpoint E.

2) Smoothing evaluation function

Considering the smoothness of the path, the fewer the number of turns, the better the energy consumption is. The smooth evaluation function of the two-dimensional path is introduced as follow:

$$T_{i,j}(t) = \begin{cases} 1/Lgrid, dr_{g,i} = dr_{i,j} \\ 0.5/Lgrid, other \end{cases} \tag{9}$$

where, $Lgrid$ is the length of cell grid; $dr_{g,i}$ and $dr_{i,j}$ are the movement direction of ant at the previous and current moments, respectively. If the two transfer directions are the same, the ant has a higher probability of transferring in the direction of the previous movement.

3) Friction factor evaluation function

The friction factor in a rough road environment is the main factor affecting the energy-efficient path planning of robots: the larger the friction factor, the greater the energy consumption generated by the work of robots to overcome traction resistance [43]. In order to make the ants tend to choose the grid blocks with smaller friction coefficients during the movement, we design the friction factor evaluation function as follows.

$$\Lambda_{i,j}(t) = \frac{\mu_{max} - |\mu_i - \mu_j|}{\mu_{max} - \mu_{min} + 0.01} \tag{10}$$

where, μ_{max} and μ_{min} denote the maximum and minimum values of the friction coefficients of the current grid i and the neighboring grid j, respectively; μ_i and μ_j denote the friction coefficients of the ant's current position and the neighboring position, respectively.

4) Energy consumption evaluation function

Considering the physical characteristics of the mobile robot to overcome friction, we introduced an energy-related criterion to quantify the accumulated path energy consumption of the ant and the energy consumption to be generated. We used this criterion as an energy consumption evaluation function, expressed as follows:

$$E_{i,j}(t) = \frac{1}{\sum \mu_x \cos\theta \, mgs_x}, x \subset visited_i \tag{11}$$

where, μ_x is the friction coefficient of ants passing through the grid path $visited_i$; θ is the slope of the ground (this paper only considers the two-dimensional plane, so we set θ to 0); m is the robot weight; g is the acceleration of gravity; s_x is the path length of the ant's single-step transfer.

3.2.2. Pheromone update considering robot energy consumption model

1) Energy consumption model

The robot's load size, mode of operation, road conditions, distance traveled, sensor composition, and electronics all affect the system's energy consumption. This is especially true for warehouse robots that continuously transport materials in large-scale and rough ground environments, requiring higher energy reserves. In this paper, the energy consumption is based on the work done by robots to overcome ground friction, and the influence of the travel distance and ground environment factors is discussed. The following assumptions are made in advance:

Assumption 1: The robot has absolute power and can follow the planned path strictly without any failure.

Assumption 2: To ensure the safety and controllability of the multi-warehouse robot system, the robot's acceleration and deceleration motion control is achieved within half a grid area and works at rated power.

The primary energy consumption [44–46] of the warehouse robots is: the energy converted into kinetic energy of the robot E_k, the energy to overcome traction resistance to do work E_f, and the loss from the conversion of the robot's hardware energy E_p, as shown in the following:

$$E = E_k + E_f + E_p \tag{12}$$

$$\begin{cases} E_k = \frac{1}{2} m|v^2(t) - v^2(t-1)| + \frac{1}{2} I\omega^2 \\ v(t) = v(t-1) + at \\ I = m * R^2 \end{cases} \tag{13}$$

$$E_f = \int \mu mg\cos\theta v(t)dt \tag{14}$$

$$E_p = \int (1-\eta)Pdt \tag{15}$$

where, m is the weight of the mobile robot itself and the supplies; $v(t)$ and $v(t-1)$ denote the robot's velocities at moments t and $t-1$, respectively; I is the rotational inertia; ω is the angular velocity of the robot during the turn; R is the radius of the tire; μ is the ground friction coefficient; g is the acceleration of gravity. η is the robot energy conversion efficiency; P is the rated power of onboard sensors and electronics.

2) Pheromone update under energy consumption constraint

Several feasible paths will be obtained after the completion of the current iteration, which will be combined with the above energy consumption model to replace the traditional ant-period model.

$$\tau_{i,j}(t+1) = \rho\tau_{i,j}(t) + \sum_{k=1}^{M} \Delta\tau_{i,j}^{k}(t) \tag{16}$$

$$\Delta\tau_{i,j}^{k}(t) = \begin{cases} \frac{Q}{E}, & \{i,j\} \subset visited_{t}^{k} \\ 0, & other \end{cases} \tag{17}$$

We know from the above equation that the evaluated energy consumption value of a path always affects the update of the pheromone, which means that the lower the energy consumption of a path, the greater the accumulation of the pheromone. The probability $p_{i,j}^{k}(t)$ is consequently doubly influenced by the energy consumption heuristic function of Eq (6) and the pheromone of Eq (16), which makes the ant colony always tend to choose the path with a low energy consumption value in the subsequent search process. A path with the lowest energy consumption is eventually evolved by our ECACO algorithm.

Taking into account the possible local optimum problem caused by the search mechanism of the ant colony algorithm, we set the pheromone in the range as shown in Eq (18) by maximum τ_{max} and minimum τ_{min}.

$$\begin{cases} \tau_{i,j}(t) \le \tau_{max} \\ \tau_{i,j}(t) \ge \tau_{min} \end{cases} \tag{18}$$

The pseudo code of ECACO above is shown in Algorithm 1.

Algorithm 1 Energy constrained ant colony optimization

```
procedure ECACO
Constructing a known environment for robots based on the grid method;
Initialize IACO parameters;
for t = 1 to T do      // Iterate from the first generation
  for k = 1 to m do      // Ant number starts from 1
    Put all ants into start grid;
    while ant k is not in endpoint do
      τij(t) and ηij(t) are obtained through Eqs (6) and (16), respectively; and choose the next
      grid by Eq (2);
    end while
    if all ants have arrived endpoint then
      E ← Energy consumption of antk's path;
      Update the pheromone by Eqs (16)–(18);
    End if
  End for
End for
Output the path with optimal energy consumption for the wheeled robot;
End procedure
```

4. Priority free ant colony algorithm for multi robot

A straightforward approach to solve trajectory coordination problems is to see all robots in the

system as one composite robot with many degrees of freedom and use some path planning algorithm to find a joint path for all the robots. However, the size of the joint configuration space that has to be searched during the planning is exponential in the number of robots. Thus, this approach quickly becomes impractical if one wants to plan for multi-robots. A pragmatic approach often useful for large multi-robot teams is prioritized planning. The idea of prioritized planning was first articulated by Erdman et al. [47].

Priority-free ant colony optimization (PFACO) is a heuristic algorithm that considers multi-robot path conflict and energy consumption cost, which is proposed for the first time in this paper. Based on our ECACO algorithm, the PFACO algorithm integrates the prioritization-free conflict strategy (PCS) and route-free conflict strategy (RCS). The execution process is as follows: First, the optimal energy consumption path of a single robot is planned by ECACO (the conflicts between robots are ignored), and the four-neighborhood search mechanism is adopted in this step. Then the robots are given the corresponding priority according to the energy consumption value. Then the low-priority robot searches for Spatio-temporal conflicts with the high-priority robot through the binary conflict-tree (CT). When a conflict is recognized, the path of the higher-priority robots holds while lower-priority robots follow wait or replan. We employ a dynamic priority adjustment method to improve conflict-free success, as the solution space of lower-priority robots is limited when there are too many robots.

PFACO consists of two crucial parts, a scheme for prioritizing robots in rough ground environments and a free strategy for resolving low-priority robot conflict. We will discuss how to fulfill them in this section.

4.1. Prioritized conflict-free strategy

It was proposed by Erdmann et al. [47] in 1987 for the priority scheme. This works as follows: First, each robot is assigned a unique priority; Second, the trajectories of the individual robots are planned in decreasing order of priority; Third, one conflict-free trajectory is obtained for each robot.

4.1.1. Prioritization order

One crucial question is how to assign different priorities to the robots. While there certainly can be some natural reasons, such as based on the task's importance or the robots' different times, this is not something we assume here. In addition, as there are N! priority plans if we have N robots, it would not be an appropriate solution to sacrifice valuable CPU time to try all plans to choose the best one [48].

We propose a simple method to assign priority to robots. In our ECACO algorithm, the optimal energy consumption path for each robot in a non-smoothed environment from the start point to the goal point is determined, so we define a robot's energy consumption query set as $Energy = \{E_1, E_2, \ldots E_m\}$. In our approach, the corresponding priority of a robot is its query position in the ascending energy consumption set $Energy^* = \{E_i, E_j, \ldots, E_z\}^*$. The rationale behind this heuristic can be illustrated in Figure 8: Robots with more significant query energy consumption traverse a larger number of regions and have a more extensive solution space than low-energy consumption robots. It should be able to complete the planning task relatively unhindered by giving it a lower priority, and this should be beneficial for reducing our energy consumption goal.

Since the path energy consumption of each robot is differentiated, it means that the priority of each robot can be specified. For example, this is done by the ECACO algorithm for the robots numbered {①, ②, ③, ④, ⑤}; Then we get $\text{Energy} = \{200, 345, 514, 490, 145\}$, after ascending

order as $\text{Energy}^* = \{145, 200, 345, 490, 514\}$, the robots correspond to priority $\{1, 2, 3, 4, 5\}$ in order, and the robots numbered {⑤, ①, ②, ④, ③}. We know that the query energy consumption of robots ⑤ and ③ are 145 J and 514 J, respectively, and the corresponding priorities are 1 and 5, respectively, so robot ③ needs to execute PCS for robot {⑤, ①, ②, ④} which has higher priority than it.

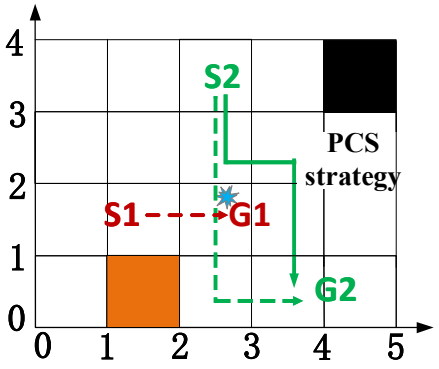


Figure 8. PCS mechanism.

4.1.2. Secondary-free mechanism

For example, as shown in Figure 9, PCS determines the numbering of AGVs as ①–④, with decreasing priority. There is a risk of collision between AGV4 and AGV3 at Step 3 when the robots start moving simultaneously, and the low-priority AGV4 cannot search for a valid path. To solve this invalid pre-planned path caused by the priority mechanism, we perform the reward and punishment operation shown in Eq (20) on the heuristic function of ECACO. It prevents the ants from selecting those nodes with spatio-temporal conflicts during the path search process as much as possible by quadratic solution. Eventually, it obtains the ideal path in Figure 10.

$$New_{\eta_{i,j}(t)} = R\ \ \eta_{i,j}(t) \tag{19}$$

$$R = \begin{cases} 0.01, If\ \ ants\ \ selecte\ \ the\ \ same\ \ node\ \ as\ \ \ loser\_Rounte\ \ at\ \ same\ \ time \\ 1, other \end{cases} \tag{20}$$

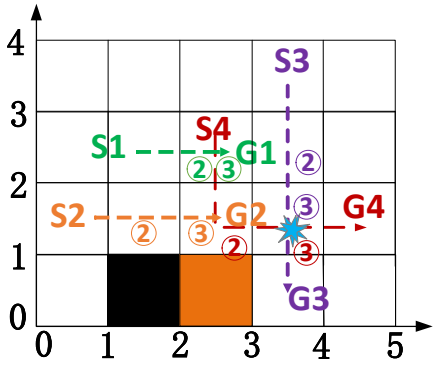


Figure 9. Unsolved path.

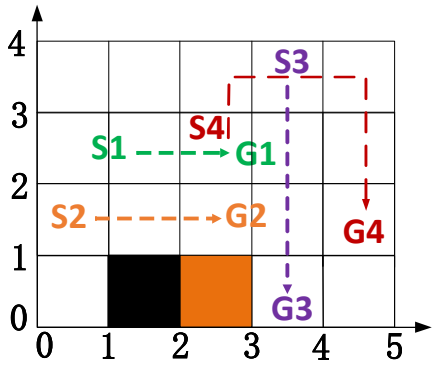


Figure 10. PCS quadratic solution path.

4.1.3 Dynamic prioritization strategy

The hypothesis is that other factors still hinder the secondary planning path of the ECACO algorithm, which will face an awkward situation, as shown in Figure 11. Considering the sensitivity of the priority mechanism to the robot scheduling order [49], we propose a priority dynamic adjustment strategy to respond. The principle is to find the highest priority robot which has a conflict with AGV4 by CT, that is AGV2, and adjust the original priority AGV1 > AGV2 > AGV3 > AGV4 to AGV1 > AGV4 > AGV2 > AGV3 by the priority insertion mechanism, and replan the paths of subsequent robots from AGV4. The coordinated conflict-free path is shown in Figure 12, and only AGV3 increases the waiting time at Step 3 moment. The dynamic priority adjustment strategy can possibly reduce the collision risk and energy consumption compared to the replanning path.

In this subsection, we discuss the PCS for resolving the overall conflict, which contains how to determine the task priority of the robots, the secondary-free operation, and the priority dynamic adjustment mechanism. In the following subsection, we will discuss the path conflict-free strategy when two robots are situation in detail.

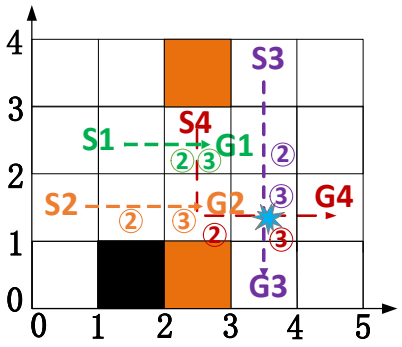


Figure 11. Failed planning.

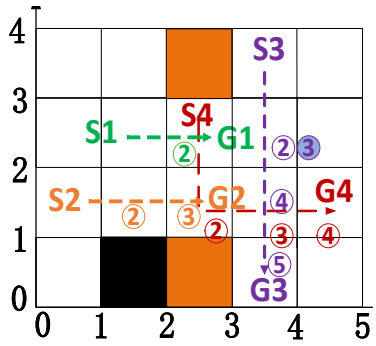


Figure 12. PCS -Dynamic prioritization strategy.

4.2. Route conflict-free strategy (RCS)

4.2.1. Four-neighborhood search mechanism

The diagonal directional transfer shown in Figure 13 may occur in a single robot's traditional eight-neighborhood search path. It is difficult to effectively judge the motion conflict between two robots, especially when there are large-scale robots. In this paper, we restrict the selection of the grid by ECACO to the four-neighborhood shown in Figure 14 through Eq (21), which makes the single-step jump distance of 1m for the ant. This is not only guaranteed robot not to collide with obstacles but also more conducive to achieving the synchronization of swarm robot movements.

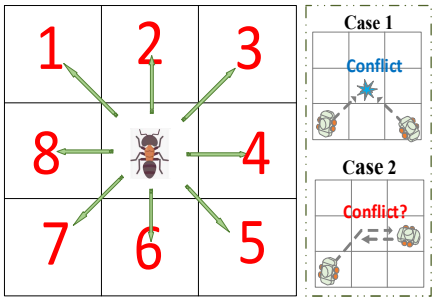


Figure 13. Eight-neighborhood search.

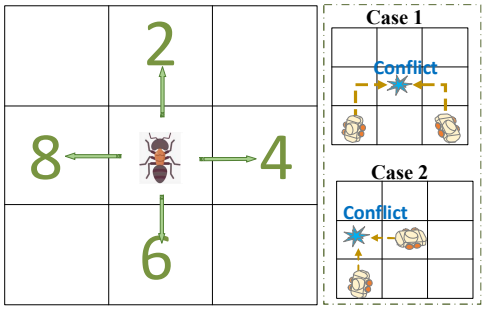


Figure 14. Four-neighborhood search.

$$D(i,j) = \begin{cases} 1, mod(Direction(j),2) = 0 \\ inf, \quad other \end{cases} \tag{21}$$

where, $D(i,j)$ is the distance from the grid i to grid j in eight-neighbourhood range; $Direction()$ is the direction which the ant move from grid i to grid j.

Algorithm 2 RCS for typical conflict

```
Input: J, Robots, visited, MAP; //adjacency matrix, path of high priority robots, visited nodes, Map
MM = size(MAPA, 1); //Map size
Dir = [-1, MM,1, -MM]; //Motion direction
temp_conflict = 0 //conflict indicator
For k = 1: 4
  k1 = Dir(k) + visited(end); // k1 is the raster number visited indicates the visited path
  position = Find(k1, Node_robots (t-1) ); //Find the location of the conflicting node
  If  position  //there is a node conflict
    Location = Find(k1, Node_robots (t-1) );
    If |(visited(end)-k1| = MM & |Robots(t) - Robots(t-1)| = 1
      temp_conflict = 1;  //Cross node conflict
    Elseif |visited(end) - k1) = 1| & |Robots(t-1) - Robots(t) = MM|
      temp_conflict = 1;  //Cross node conflict
    Else  //Phase node conflict
      Continue
    End
  End
  If k1 = Robots (t-1) & visited (end) = Robots(t) // There is an alignment conflict
    Continue
  End
  J(Jc) = k1;  // The set of raster to be visited
  N(Jc) = k;  // set of directions to be selected
  Jc = Jc + 1;
End
If visited(end) ~ = Robots(t) & temp_conflict = 1 // Waiting mechanism for node conflict trigger
  J = []; N = [];
  J = visited(end);
  N = last_direct; // movement direction is the same as last time
End
Output J and N //Node number and direction of movement to be transferred
```

4.2.2. RCS for typical path conflicts

Node and counterpoint conflicts are the most typical conflict in MAPF, where two robots meet in motion. We determine the movement nodes of the low-priority robots according to the node expansion

approach in Section 4.2.1 to stop-wait or detour. The stop-wait mechanism is shown in Figure 15, which means expanding the current node as the next state point and resuming the motion after the high-priority robot leaves. The detour mechanism applies when the stop-wait mechanism fails, and the low-priority robot can only choose a conflict-free path with the high-priority one. The RCS pseudo-code for the typical conflict is shown in Algorithm 2.

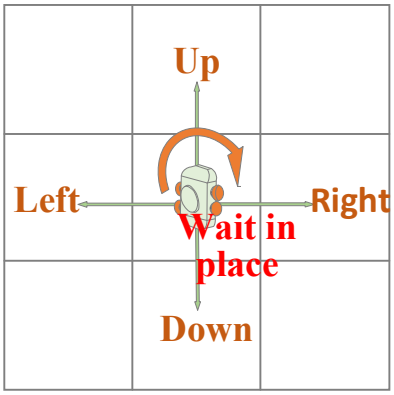


Figure 15. Waiting in place.

1) Cross-shaped node conflict: as shown in Figure 16, the priority of conflicting robots is judged by ascending energy consumption priority strategy. The low-priority robot AGV2 implements wait-in-place, i.e., the extended wait point is the next state point until the motion path of the high-priority robot AGV1 no longer affects its movement and then continues to travel along the original path.

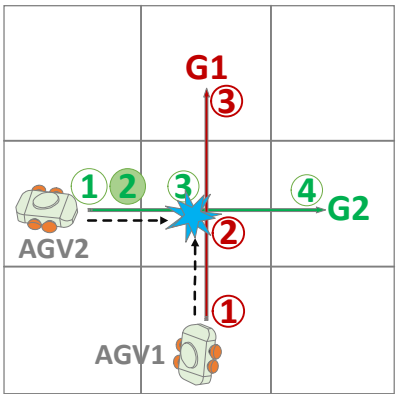


Figure 16. Cross node conflict RCS.

2) Phase direction type node conflict: As shown in Figure 17, the high-priority robot AGV1 continues to move according to the planned path, and the low-priority robot AGV2 performs a local re-planning path movement to the target point.

3) Counterpoint conflict: As shown in Figure 18, a collision between robots must exist, and the low-priority wait-in-place strategy fails. At this time, the high-priority robot AGV1's path remains unchanged, and the low-priority robot AGV2 gets a new path.

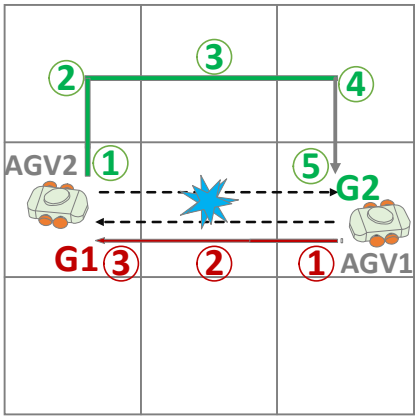


Figure 17. RCS of phase wise node conflict.

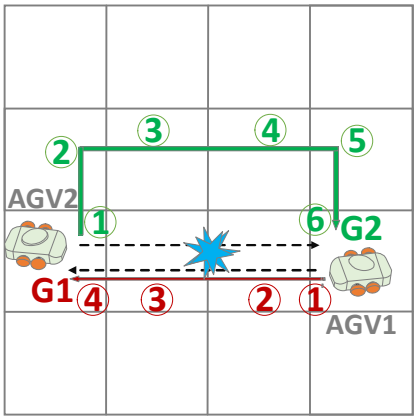


Figure 18. Counterpoint conflict RCS.

4.2.3. RCS for priority-derived conflicts

Node and counterpoint conflicts are typical motion conflicts, and fewer scholars have focused on the risk of occupancy conflict and large-scale robot blockage coordination problems derived from priority MAPF. We additionally concentrate on the green path planning problem, with the following RCS for resolving related conflicts.

1) Placeholder Conflict

Occupancy conflict define: the target point of the low-priority robot is located exactly on the path not passed by the high-priority robot.

RCS (The RCS pseudo code is shown in Algorithm 3):

❖ **Strategy 1**- As shown in Figure 19, it is predicted that there is an occupancy conflict between the low-priority AGV2 and the high-priority AGV1. We then allow AGV2 to activate the waiting mechanism at the nearest Step3 position (conflict-free node) from the target point and resume its movement after the conflict release.

❖ **Strategy 2-** As shown in Figure 20, the path is replanned if the conditions of AGV3 would not enable a waiting mechanism of strategy 1 to be triggered. In this process, the waiting mechanism will be triggered cyclically so that we can also save energy resources while taking into account the obstacle avoidance process as much as possible.

Algorithm 3 RCS for occupancy conflict

```
Input: J, Goal, Robots, visited
//adjacency matrix Goal point Path of high priority robots Visited nodes
If find(J = Goal) //Robots may next choose a goal point with conflict
  For i = 1: nub - 1
    position = Find(Goal, Robot(i)); // Find the location of the conflicting node
    If position > length(visited) + 1 & visited(end)~ = Robots(t + 1) //Robots have conflicting
    positions and there is no temporal conflict node
      J = visited(end);
      N = last_direct;
      Break
    Else //discard the target point from the node to be selected
      Position* = Find(J,Goal);
      J(Position*) = [];
    End
  End
End
Output J and N
```

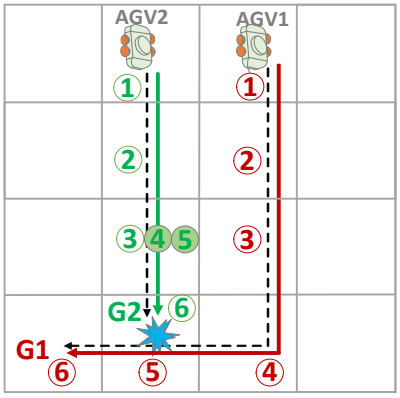


Figure 19. Occupancy conflict strategy 1.

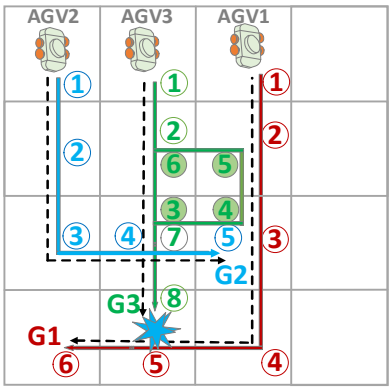


Figure 20. Occupancy conflict strategy 2.

2) Blockage Conflict

Blockage conflict define: low-priority robot meets high-priority robot in a narrow environment.

RCS (The RCS pseudo-code is shown in Algorithm 4):

Algorithm 4 RCS for Blockage conflict

```
Input: J, temp_conflict ,visit    //adjacency matrix Cross node conflict indicator Visited nodes
Back_number = 0; Back_conflict = 0
If isempty(J) & temp_conflict == 1 //the set of accessible nodes J is empty and there is a conflict == >> back strategy
   Back_conflict = 1; //back-off    has occurred
   If length(visited)-2 * (Back_number + 1) + 1>0 //Not back to the start
      J = visited(end-2 * (Back_number + 1) + 1); // contains the matrix of raster numbers to be visited
      N = last_direct; //retains the direction of the last movement
      Back_number = Back_number + 1; //Record the number of backtracking
   Else //has backed up to the starting point, then wait for
      J = visited;
      N = last_direct;
   End
Elseif J = [] & temp_conflict == 0 //If it can't backtrack, the ant is dead
   break
End
If Back_conflict == 0 //No backtracking occurred this time, then the backtracking number is cleared to 0
   Back_number = 0
End
Output J and N
```

❖ **Strategy 1**- As shown in Figure 21, a failed path is searched by low-priority AGV3 at Step3 time under massive conflict. We allow AGV3 to retreat to the previous path node if nothing is disturbed by other robots at the next moment. If this conflict holds, it will stop the movement at the original safe position to save energy. The movement on the original path is continued as soon as the conflict is solved. Otherwise, the path of AGV3 is re-planned.

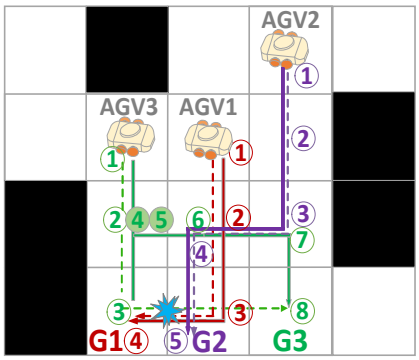


Figure 21. Blockage conflict strategy 1.

❖ **Strategy 2-** In the particular case shown in Figure 22, there is no wait command satisfied by the low-priority AGV2. It will trigger a multiple fallback method without affecting the movement of high-priority AGV1 until the conflict is entirely released before resuming the original motion.

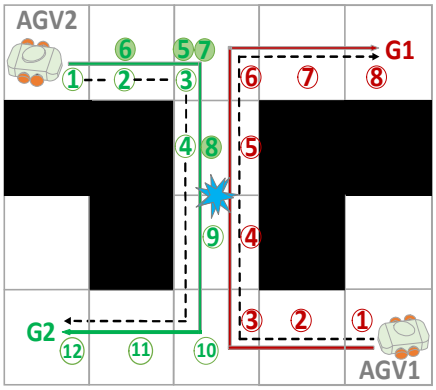


Figure 22. Blockage conflict strategy 2.

Remember: We did introduce the secondary freedom mechanism in Section 4.1.2 and the dynamic priority policy in Section 4.1.3 for resolution in the planning process when RCS failed to resolve the above four conflicts effectively.

4.3. Algorithm flow

The flow of our multi-robot energy-efficient planning algorithm is shown in Figure 23.

5. Simulation experiments

This paper is a simulation experiment on Matlab 2020a with the computer performance of Lenovo R7000, i5-11400H@2.70GHz. The relevant parameters of the robot and the ant colony algorithm are shown in Table 1.

Table 1. Main parameters of our robot and algorithm.

Parameter	Value	Parameter	Value
Robot weight m (kg)	10	Velocity v at uniform motion (m/s)	1
Gravitational acceleration g (m/s^2)	9.8	Angular velocity ω (rad/s)	10
Tire radius R(m)	0.15	Motor efficiency η	0.78
Acceleration a at start/stop (m/s^2)	1	Motor power P (w)	200
Number of ants M	30	τ_{max}	10
Number of iterations Nc	50	τ_{min}	40
Pheromone Heuristic Factor α	1	w	20
Expectation heuristic factor β	3	x	1
Pheromone constant Q	100	y	10
Pheromone volatilityρ	0.8	z	1
Lgrid	1		

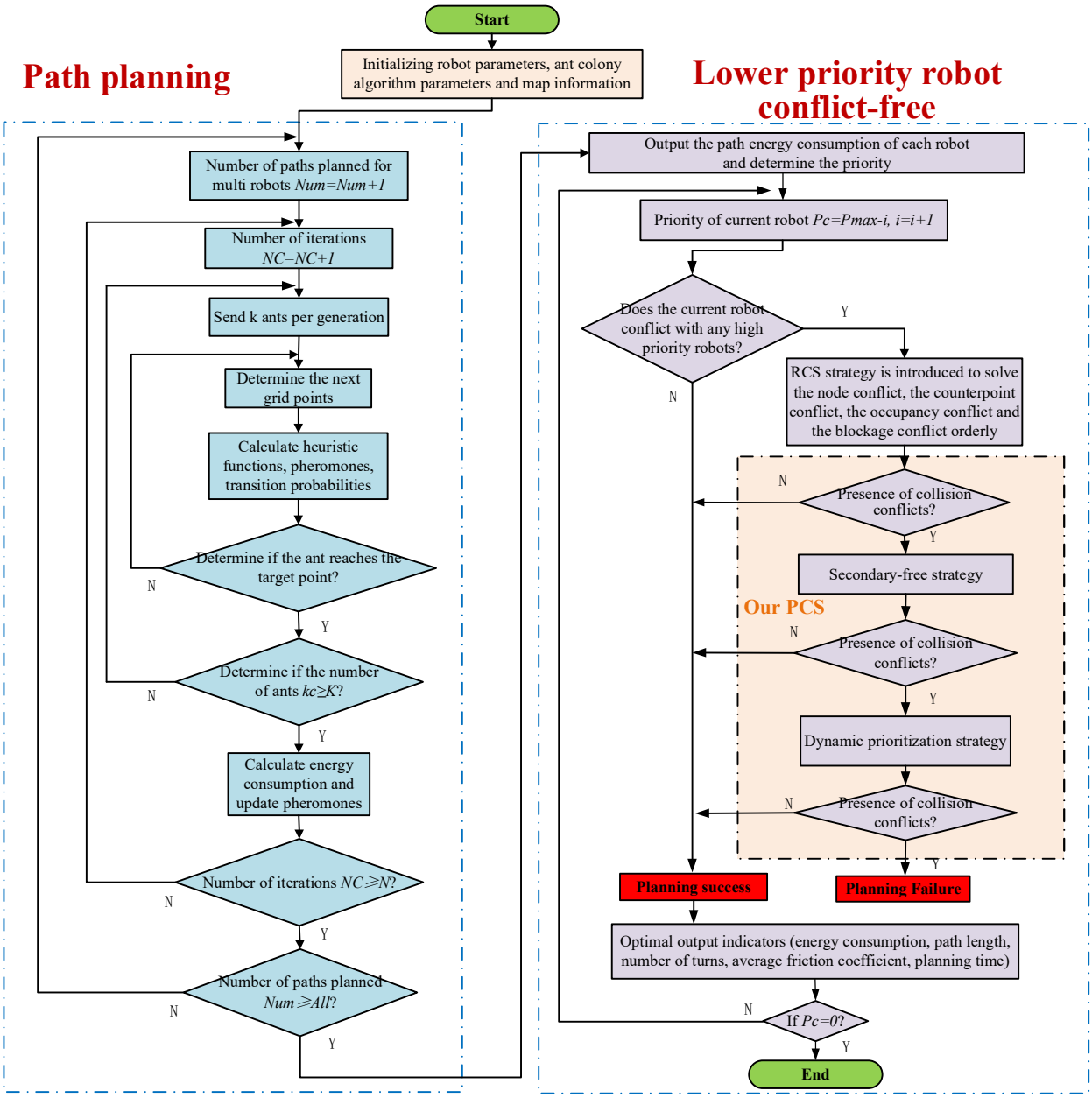


Figure 23. Flow chart of PFACO.

5.1. Single-robot path planning comparison experiment

In order to verify the energy-saving effect of the ECACO algorithm proposed in this paper on robot motion, a dual-resolution map is used to simulate an unstructured scenario. The map model uses a 20 m × 20 m scale, the friction coefficient of the non-flat road surface is randomly generated between 0 and 0.45, and the robot's weight is 10 kg. To reflect the applicable scenarios of our algorithm, we consider three standard neighborhood search methods for path planning: traditional eight-neighborhood, secure eight-neighborhood [50], and optimized four-neighborhood [51]. In addition, we verify the effectiveness of ECACO in energy saving by ablation experiments under each type of approach. The compared algorithms include Improved heuristic function-ACO (IHF-ACO), Improved pheromone update (IPU-ACO), traditional ACO, traditional A* and IA*[52].

5.1.1. Traditional eight-neighborhood search method

We conducted comparative experiments in an environment filled with diagonal obstacles, and the results of the simulation experiments are shown in Table 2 and Figures 24 and 25. It can be noticed that the energy cost of the path planned by ECACO in this paper is 9973 J, which is optimized by 5.04, 5.62, 2.70, 16.09 and 17.08% compared to IHF-ACO, IPU-ACO, traditional ACO, A* and IA*, respectively. A* and IA* outperforms ECACO slightly in terms of path length and solution time. Regarding path length and the number of energy-stable iterations, our ECACO algorithm is significantly better than the other algorithms with both 6 iterations. In terms of path smoothing, IHF-ACO and IA* perform the worst, ECACO has the least number of turns. In general, the algorithms in this paper outperform the comparison methods in comprehensive indexes. However, the paths planned by all algorithms in the traditional eight-neighborhood search method pass through the diagonal obstacle at (15, 4) and collide with the obstacle's edge many times on the map. Sections 5.1.2 and 5.1.3 will explore the safe ground path planning problem under the two safe neighborhood search mechanisms. The IA* algorithm improves the traditional heuristic function of A*, so it has its advantages in node expansion as well as program efficiency but may have the awkward situation of non-optimal solutions.

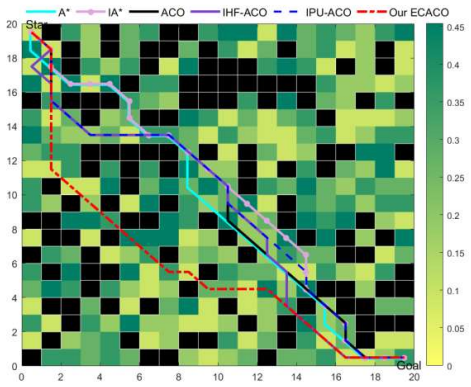


Figure 24. Path planning.

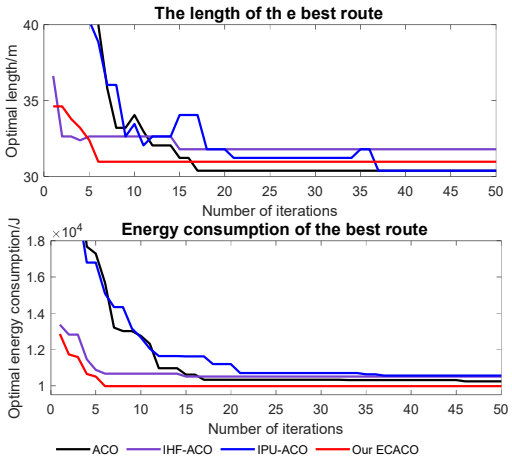


Figure 25. Iterative diagram of path length and energy consumption.

Table 2. Experimental results.

Optimal path indicators	Our ECACO	IHF-ACO	IPU-ACO	ACO	A*	IA*
Energy consumption (J)	9973	10502	10567	10250	11885	12027
Path length (m)	30.9706	31.7990	30.3848	30.3848	29.7990	29.7990
Number of turns (time)	7	13	11	9	12	13
Friction coefficient mean square difference	0.1448	0.1128	0.1148	0.1200	0.1356	0.1321
Number of iterations of path length stabilization convergence	6	15	37	17	---	---
Number of energy-stable convergence iterations	6	15	38	41	---	---
Algorithm solving time (s)	1.2179	3.5864	6.9591	1.0853	0.5102	0.3171

5.1.2. Security eight-neighborhood search method

We remained in the more complex diagonal obstacle environment and conducted safety eight-neighborhood experiments, as shown in Figures 26 and 27 and Table 3. The results show that the paths planned by all algorithms after improving the neighborhood search maintain a certain safety distance from the obstacles, which improves the safety of robot passage compared with Section 5.1.1. In terms of energy consumption solution, the path energy consumption of our ECACO and traditional A* are 11,089 J, which is optimized by 1.67, 2.11, 0.40 and 2.26% compared to IHF-ACO, IIU-ACO, traditional ACO and IA*, respectively. Regarding the path length, IPU-ACO plans the longest path, ACO is the second, and the shortest path length of the robot in this paper is 30.3848 m. The stable iterations of path length and energy consumption in our ECACO are 6, which is better than other algorithms. In the smoothness of the path, ECACO, IHF-ACO, A* and IA*are the same, and both are better than IPU-ACO and ACO. As for the solution time, IHF-ACO has the worst solution time, IA* has the best path solution time.

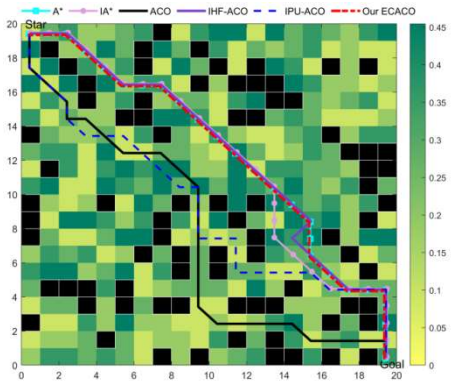


Figure 26. Path planning.

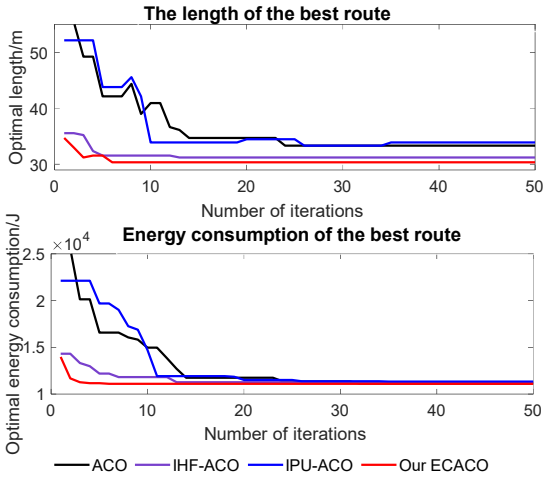


Figure 27. Iterative diagram of path length and energy consumption.

Table 3. Experimental results.

Optimal path indicators	Our ECACO	IHF-ACO	IPU-ACO	ACO	A*	IA*
Energy consumption (*J*)	11089	11277	11328	11133	11089	11345
Path length (*m*)	30.3848	31.2132	33.8995	33.3137	30.3848	30.9706
Number of turns (*time*)	7	7	13	12	7	7
Friction coefficient mean square difference	0.1101	0.1058	0.1242	0.1282	0.1101	0.1165
Number of iterations of path length stabilization convergence	6	13	35	35	---	---
Number of energy-stable convergence iterations	6	13	35	32	---	---
Algorithm solving time (*s*)	1.2887	4.3201	1.6929	1.451	0.7868	0.5700

It can be seen from the comprehensive analyses that the IPU-ACO using energy consumption model optimization has poor algorithm-seeking ability due to the lack of a reasonable heuristic guide. While the IHF-ACO, considering the energy consumption model heuristic function, can achieve low-energy path planning, it also lacks a reasonable pheromone strategy to enhance the iterative evolution of the algorithm. In this paper, ECACO integrates the advantages of IPU-ACO and IHF-ACO under the security eight-neighborhood mechanism to achieve better energy-efficient path planning. Both IA* and A* are heuristic algorithms with path length as the goal of finding the optimal path. IA* improves program execution efficiency but presents the awkward situation of non-optimal paths, as described in Section 5.1.1. The distribution of obstacles is relatively simple in this case, resulting in the path with the shortest path length also being the lowest energy one. Thus, the A* achieves the same effect as ECACO in this paper under the safe eight-neighborhood search mechanism, but there is a significant difference under the four-domain search in Section 5.1.3.

5.1.3. Optimizing the four-neighborhood search method

The more popular current node search strategy for multi-robot systems is the four-neighborhood approach (described in Section 4.2.1). To facilitate the experimental verification of the multi-warehouse robots in subsequent Sections 5.3.1 and 5.3.2, the energy-efficient planning of ECACO is verified in a single-robot environment in advance. The simulation and experimental results are shown in Table 4 and Figures 28 and 29. In terms of energy consumption solution, the energy consumption of the ECACO in this paper is 11,219 J, which is 11.32, 8.39, 13.16, 19.91 and 26.30% optimized compared with IHF-ACO, IIU-ACO, traditional ACO, A* and IA*, respectively. As for the solution time, ECACO has the worst solution time, and IA* has the best path solution time. Since the optimized four-neighborhood approach is adopted, the path lengths of all six algorithms are the same. However, our ECACO significantly reduces the energy cost due to its clear superiority over other algorithms in path smoothing and node selection for rough pavement. In addition, ECACO outperforms the IHF-ACO, IIU-ACO and traditional ACO regarding path and energy consumption iteration stability. Taken together, it can be seen that the ECACO algorithm proposed in this paper is suitable for multi-robot environments and can reduce energy costs.

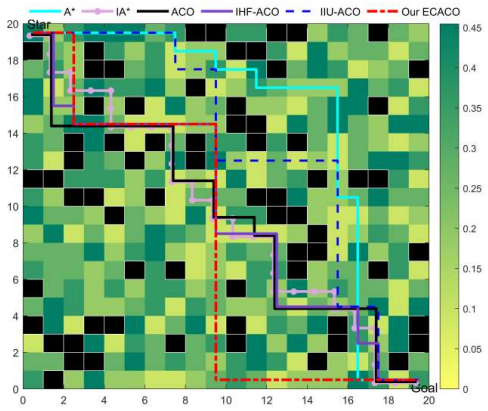


Figure 28. Path planning.

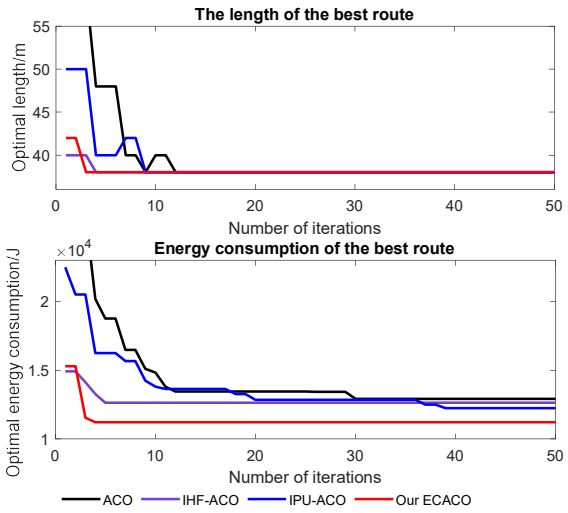


Figure 29. Iterative diagram of path length and energy consumption.

Table 4. Experimental results.

Optimal path indicators	Our ECACO	IHF-ACO	IPU-ACO	ACO	A*	IA*
Energy consumption (*J*)	11219	12651	12247	12919	14008	15222
Path length (*m*)	38	38	38	38	38	38
Number of turns (*time*)	4	12	8	12	10	22
Friction coefficient mean square difference	0.1225	0.1248	0.1322	0.1277	0.1325	0.1219
Number of iterations of path length stabilization convergence	3	4	9	12	---	---
Number of energy-stable convergence iterations	4	5	39	30	---	---
Algorithm solving time (*s*)	1.8781	1.5603	1.4352	1.4343	0.2507	0.1631

5.2. Multi-robot priority scheduling sensitivity verification

The PFACO algorithm in this paper incorporates three strategies: single-robot ECACO, PCS for determining multi-robot priorities, and RCS for coordinating multi-robot conflicts. Section 5.1 verifies the energy-saving effectiveness of ECACO. This section will test the sensitivity of robot priority scheduling on MAPF problem through a 10 m × 10 m scale blank rough environment and a 10 m × 10 m scale obstacle rough environment energy consumption. Within the framework of ECACO-PCS-RCS in this paper, three types of PCS are selected for comparison: energy consumption ascending PCS (the smaller the energy consumption value, the higher the robot priority), energy consumption descending PCS (the more extensive the energy consumption value, the higher the robot priority), and regular PCS (the robot priority is not specially treated). We set up 10 robots in each of the environments for 10 labs, and the starting and endpoints of each robot were randomly generated. The algorithm performance was quantified by averaging the energy consumption, path length, the number of turns, mean squared difference of friction coefficients, and time spent on algorithm solving.

Table 5. Experimental results.

Experimental index	Our PCS	Descending PCS	Conventional PCS
Average energy consumption *(KJ)*	73.858	85.408	80.113
Average path length *(m)*	61.8	73.5	67.3
Average number of turns *(time)*	21	26	22
Average friction coefficient mean square difference	0.2702	0.2682	0.2679
Algorithm average time *(s)*	11.9934	13.9810	12.2915

5.2.1. Blank rough environment

The experimental results of the three PCS strategies in the blank rough environment are shown in Figures 30–32 and Table 5. What can be known is that the average energy consumption value of PCS

in this paper is 73.858 KJ, which is 13.5 and 7.8% more optimized than descending PCS and conventional PCS, respectively. Regarding path length solving, the results of this paper and the descending PCS are optimal and worst, respectively. As for path smoothing, Our PCS and conventional PCS solving effects are close, and the descending PCS performs the worst. The three strategies are close in effect in terms of the average friction coefficient mean square deviation and algorithm-solving time. In addition, we can obviously discover from Figure 30 that the descending PCS performs the worst in each experiment, and the conventional PCS outperforms our PCS only in the seventh experiment. Overall, the strategy of this paper to query the energy consumption of robots in blank rough environments based on the ECACO algorithm and assign higher priority to robots with lower energy consumption values is feasible and effective.

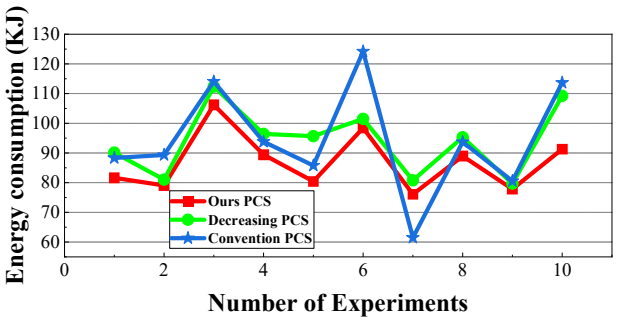
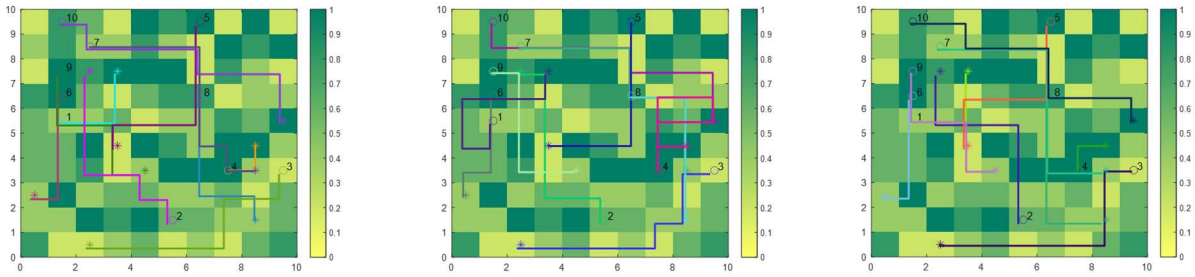
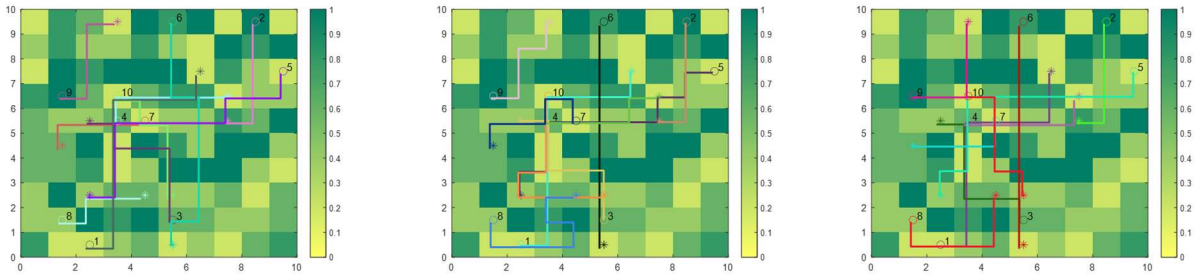


Figure 30. Multi-robot energy consumption distribution.



(a) This paper PCS (b) Descending PCS (c) Regular PCS

Figure 31. Path diagram of the 9th experiment.



(a) This paper PCS (b) Descending PCS (c) Regular PCS

Figure 32. Path diagram of the 10th experiment.

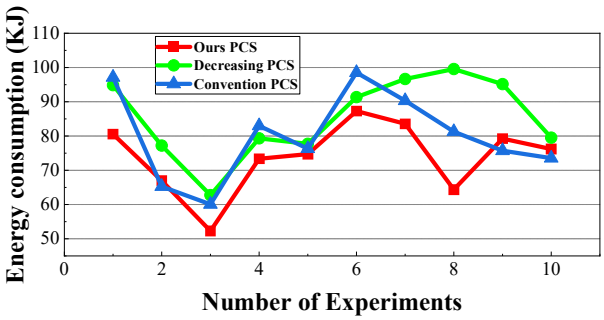
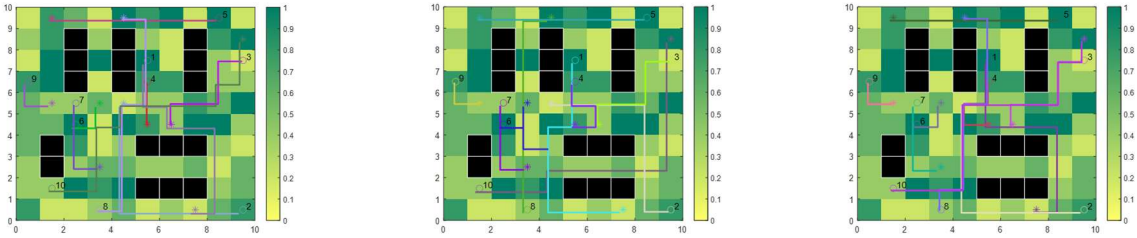
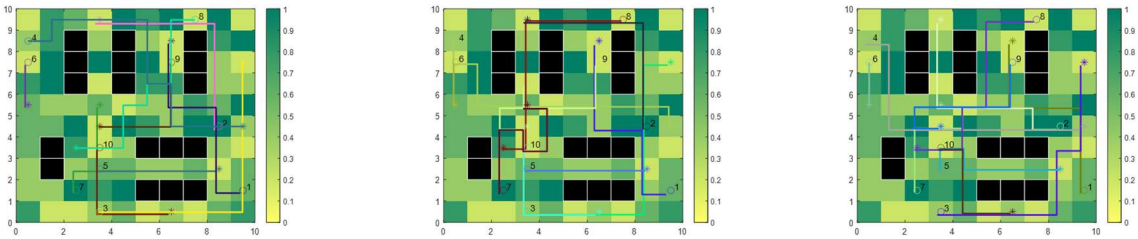


Figure 33. Multi-robot energy consumption distribution.



(a) This paper PCS **(b)** Descending PCS **(c)** Regular PCS

Figure 34. Path diagram of the 9th experiment.



(a) This paper PCS (b) Descending PCS (c) Regular PCS

Figure 35. Path diagram of the 10th experiment.

5.2.2. Obstacle rough environment

This section adds the obstacle environment to the above rough terrain experiments to further expand the test scenario of the algorithm, and the experimental results are shown in Figures 33–35 and Table 6. What can be known is that the energy consumption values of our PCS, descending PCS, and conventional PCS are 86.9405, 94.2088 and 94.4898KJ, respectively. Our PCS is optimized by 4 and 8% compared to the two strategies. Our strategy performs optimally regarding path length, path smoothing and algorithm time-solving. We compare three different PCS strategies in the framework of the ECACO-PCS-RCS algorithm through this example experiment and verify the feasibility of the ascending PCS to determine the robot priority in this paper. In the following subsection, we will expand

the number of robots and the scale of the environments, then compare and analyze multiple performance metrics in both horizontal and vertical directions.

Table 6. Experimental results.

Experimental index	Our PCS	Descending PCS	Conventional PCS
Average energy consumption (KJ)	86.9405	94.2088	94.4898
Average path length (m)	72.9	81.5	80.1
Average number of turns (time)	25	27	29
Average friction coefficient mean square difference	0.27712	0.27206	0.27474
Algorithm average time (s)	15.17284	18.82398	17.85909

5.3. Multi-robot path planning comparison experiment

In order to evaluate the practical advantages of the multi-robot energy-efficient path planning algorithm in this paper, two different scenarios are selected for comparison experiments, which are 20 m × 20 m scale blank rough and 40 m × 40 m scale rough warehouse environments. Based on the algorithmic framework of ECACO-PCS-RCS in this paper, our PFACO algorithm is compared with three methods, ACO-TPCS-RCS, ACO-PCS-RCS and ECACO-TPCS-RCS, in each scenario. Where ACO is the ant colony algorithm of the traditional ant perimeter model; ECACO is the ant colony algorithm of the energy consumption model in this paper; TPCS is the traditional priority strategy; PCS is the priority strategy in this paper; RCS is the free strategy for multi-robot path conflicts (to ensure that there is no collision conflict among robots) in this paper.

Regarding experimental design and comparison metrics: the number of robots increases from 10 until the total number is over 100, with 10 more each round. For each experiment, the starting and target points are randomly generated to match the number of robots, and they are guaranteed to be different from each other to avoid invalid conflict. In order to avoid the chance of single simulation experiment results, 20 non-repeated experiments are conducted for the same number of robots, and the average of these experiments reflects the algorithm's accuracy. The performance indexes of the algorithm mainly include path planning success rate (We only count those cases where the paths are free of any conflicts), average energy consumption, average path length, the average number of turns, average variance of ground friction coefficient, and average program running time value.

5.3.1. Blank rough environment

As shown in Figure 36, when the number of robots is small (less than 20), the success rate of all two algorithms (PFACO, ACO-PCS-RCS) in planning paths is 100%. The success rate of solving ACO-TPCS-RCS and ECACO-TPCS-RCS starts to decrease obviously when the number of robots has more than 20, and the conflict among robots increases, so the robots with low priority under the TPCS strategy may not be able to find feasible paths. In contrast, based on our PCS strategy, PFACO and ACO-PCS-RCS success rates only start to decrease when the number of robots is 60 and 50, respectively. However, our PFACO maintains a high success rate, which is still 15% higher than ACO-PCS-RCS at a scale of 100 robots. As shown in Figures 37–40 for each comparison metric, the comparative analysis of PFACO and ACO-PCS-RCS is focused due to the premature death of the ACO-TPCS-RCS and ECACO-TPCS-RCS algorithms. When the number of robots is 10, the energy

consumption solution result of our PFACO is 4.0 KJ better than ACO-PCS-RCS. When the number of robots is increased to 100, the difference achieves 30.4 KJ. The difference between the two algorithms increases as the number of robots increases, indicating that our PFACO algorithm framework is not only better at solving robot conflict but also more energy efficient. This is mainly reflected in the fact that PFACO significantly outperforms the other algorithms in terms of path length, the number of turns, and friction coefficient in Figures 38–40. Regarding the solution time shown in Figure 41, the difference in solution time between the four algorithms is small when the number of robots is small. As the number of robots increases and the congestion of robots becomes higher, the wait-in-place and detour behaviors performed by the RCS strategy rise, and the solution times of PFACO and ACO-PCS-RCS increase significantly. The maximum solution time of ACO-PCS-RCS is 13.4 s larger than our method at first, but the difference between the two is 255.3 s when the number of robots reaches 100. The path schematic of the maximum number of robots that the algorithm can achieve is shown in Figure 42. Our PFACO execution success rate is higher and the quality of the planned paths is better, as shown by comparing the various metrics of the four algorithms in the blank rough environment.

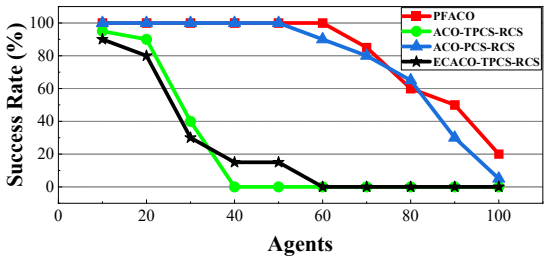


Figure 36. Algorithm success rate.

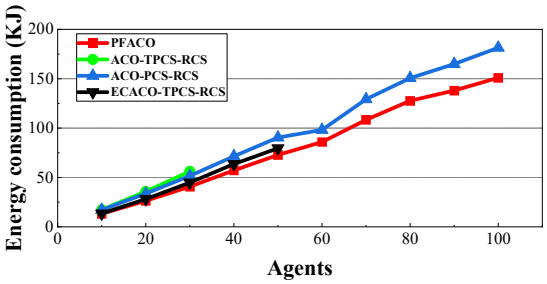


Figure 37. Average energy consumption.

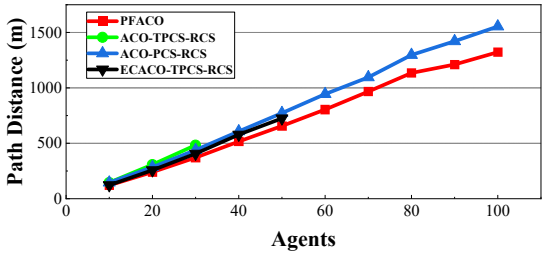


Figure 38. Average path length.

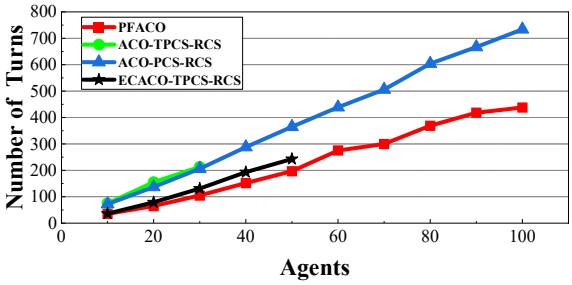


Figure 39. Average total number of turn points.

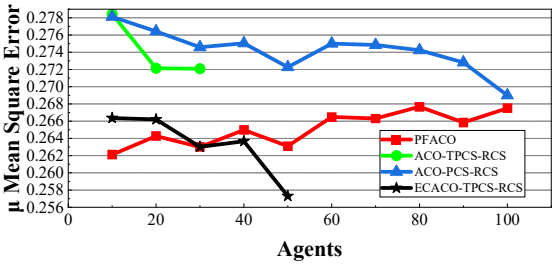


Figure 40. Average friction coefficient mean square deviation.

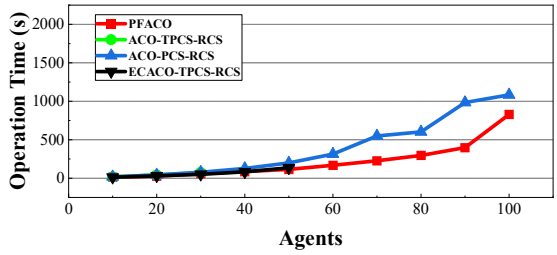
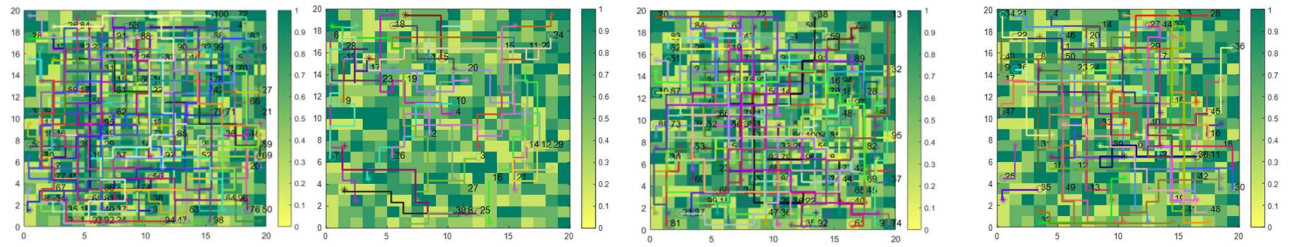


Figure 41. Average running time.



(a) PFACO (Agents = 100) | **(b)** ACO-TPCS-RCS (Agents = 30) | (c) ACO-PCS-RCS (Agents = 100) | (d) ECACO-TPCS-RCS (Agents = 50)

Figure 42. Multi robot path planning.

5.3.2. Rough warehouse environment

To further verify the effectiveness and stability of the algorithm, the maximum load m = 500kg of the unmanned warehouse robot of Jingdong Company was used as the background. The relevant parameters of ECACO were modified as follows: the pheromone constant Q was 10,000, the number of ants M was 20, and the maximum number of iterations NC was 10. The size of the unstructured warehouse scene is 40 m × 40 m, and 48 shelves are placed. We conducted more comprehensive testing and analysis of the algorithms by the different levels of robots and random tasks, and the experimental results are shown in Figures 43–49.

The solution success rates of all algorithms are shown in Figure 43, where the success rates of ACO-TPCS-RCS and ECACO-TPCS-RCS still decrease to 0 prematurely, and the following analysis is focused on PFACO and ACO-PCS-RCS. Both PFACO and ACO-PCS-RCS success rates are at 100% when the number of robots is less than 60, and the success rate gradually decreases as the number increases. Although our ECACO outperforms ACO significantly for individual robots (verified in Section 5.1.3), this also means that more path intersections and more conflicts are generated when applied to multiple robots. These experiments add obstacle constraints compared to the obstacle-free environment in Section 5.3.1, making the success rate of our ECACO slightly worse than that of ACO-PCS-RCS when the number of robots is 90–120. However, it is clear from the energy consumption shown in Figure 44 that the energy consumption advantage of our PFACO algorithm is still apparent, especially when the number of robots is 120, the average energy consumption results of PFACO and ACO-PCS-RCS algorithms are 10207.1 KJ and 14675.2 KJ, respectively, and ours reduces 30% consumption compared with ACO-PCS-RCS. In Figures 45–48, with the number of robots at 120, our PFACO algorithm outperforms the ACO-PCS-RCS in terms of path length, the number of turns, friction coefficient, and algorithm running time by 24, 53, 3 and 51%, respectively.

Through the comparison experiments under two different environments in Section 5.3, we thoroughly verify the advantages of the PFACO algorithm in this paper regarding the success rate and cost of multi-robot path planning, and the innovative and industrial reference value of the theory is also reflected.

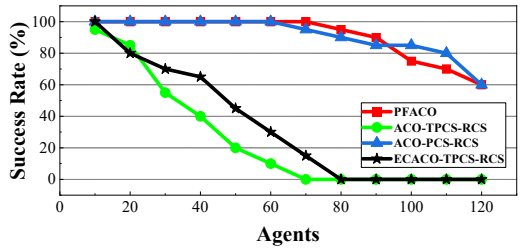


Figure 43. Algorithm success rate.

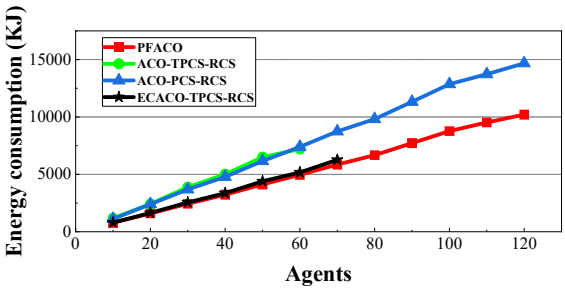


Figure 44. Average energy consumption.

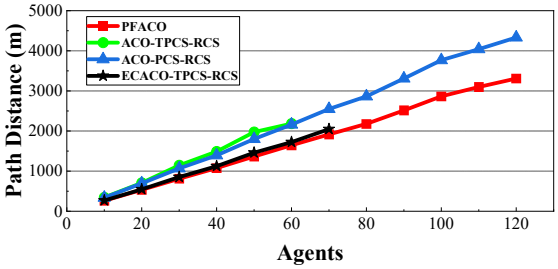


Figure 45. Average path length.

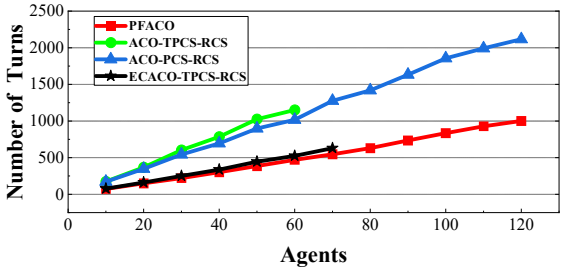


Figure 46. Average total number of turn points.

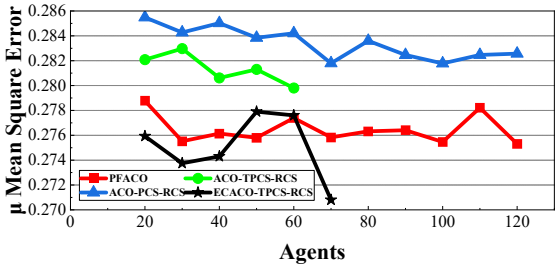


Figure 47. Average friction coefficient mean square deviation.

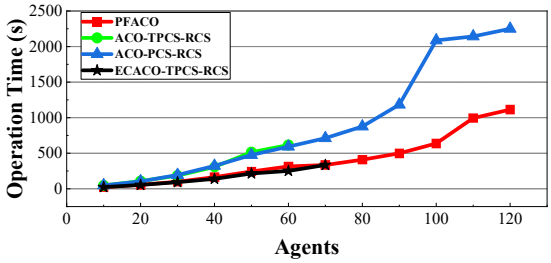
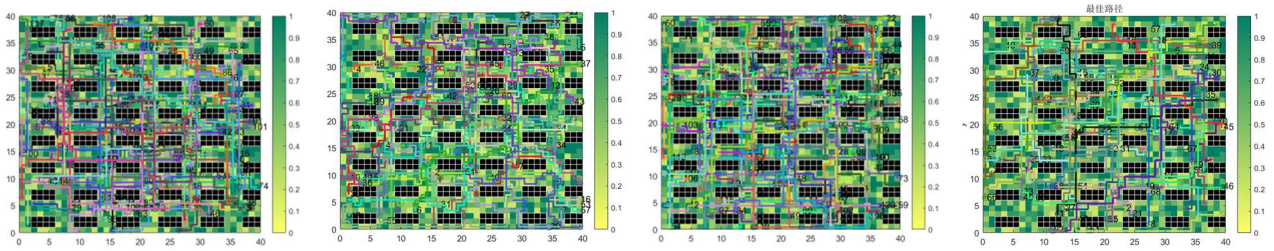


Figure 48. Average running time.



(a) PFACO (Agents = 120) (b) ACO-TPCS-RCS (Agents = 60) (c) ACO-PCS-RCS (Agents = 120) (d) ECACO-TPCS-RCS (Agents = 70)

Figure 49. Multi robot path planning.

6. Conclusions and prospect

Multi-robot coordination and cooperation in completing work will be the trend of robotics research. This paper proposes a multi-robot energy-efficient path planning algorithm based on the

priority scheduling concept with limited energy storage and robots' energy consumption as the MAPF background. First, we design an unstructured storage environment modeling method that considers both obstacle and rough ground friction factors. Second, we propose an ECACO energy-efficient planning algorithm for a single mobile robot, which contains a heuristic function for energy consumption constraint improvement and a pheromone update strategy for energy consumption model optimization. The improved heuristic function considers the path length, path smoothness, road friction coefficient, and path energy consumption. The improved pheromone update considers the transformed kinetic energy of robot motion, energy loss from overcoming traction resistance, and robot hardware energy consumption. Then, we propose a PFACO energy-efficient planning algorithm for multi-warehouse robots, which incorporates Prioritized Conflict-free Strategies (PCS) and Route Conflict-free Strategies (RCS) based on ECACO. PCS mainly solves the deadlock conflicts of multiple robots by swapping the priority between conflicting robots and re-route planning to free conflict. RCS mainly resolves node conflict, counterpoint conflict, occupancy conflict and blockage conflict among multiple robots based on the methods of waiting in place or re-route planning. The results show that our ECACO algorithm and the extended PFACO algorithm have more significant improvements in energy savings and solution success rate than other methods in complex rough terrain, which are of reference value when applied to practical warehouse robots.

The performance of our PFACO algorithm was fully validated in this study, but further improvement of the algorithm success rate, path performance metrics, and program execution efficiency in the large-scale robot scenario will continue during future work. In addition, more complex factors such as MAPF multitask planning (TSP problem), intermittent communication, and dynamic obstacle problems will also be focused on in future research.

Data available

Detailed multi-robot experimental data can be obtained from the link: https://github.com/1354109872/PFACO.

Conflict of interest

The authors declare that there are no conflicts of interests.

References

1. B. Ai, J. C. Jiang, Yu. S, S. S. Yu, Y. C. Jiang, Multi-agent path finding with heterogeneous edges and round trips, *Knowl.-Based Syst.*, **234** (2021), 107554. https://doi.org/10.1016/j.knosys.2021.107554
2. E. T. S. Alotaibi, A. Hisham, Multi-robot path-planning problem for a heavy traffic control application: a survey, *Int. J. Adv. Comput. Sci. Appl.*, **7** (2016), 179–188. https://doi.org/10.14569/IJACSA.2016.070623
3. G. Sharon, R. Stern, M. Goldenberg, A. Felner, The increasing cost tree search for optimal multi-agent path finding, *Artif. Intell.*, **195** (2013), 470–495. https://doi.org/10.1016/j.artint.2012.11.006
4. J. J. Yu, S. M. LaValle, Optimal multirobot path planning on graphs: complete algorithms and effective heuristics, *IEEE Trans. Rob.*, **32** (2016), 1163–1177. https://doi.org/10.1109/TRO.2016.2593448

5. L. Tian, Z. Zhang, C. Zheng, Y. Tian, Y. Zhao, Z. Wang, et al., An improved rapidly-exploring random trees algorithm combining parent point priority determination strategy and real-time optimization strategy for path planning, *Sensors*, **21** (2021). https://doi.org/10.3390/s21206907
6. F. Lu, C. Han, G. X. Wu, M. R. Lu, J. K. Yang, B. R. Miao, et al., Dynamic adaptive security path planning based on A* algorithm, in *Journal of Physics: Conference Series*, **2234** (2022), 012002. https://doi.org/10.1088/1742-6596/2234/1/012002
7. M. B. Du, J. J. Chen, P. Zhao, H. W. Liang, Y. Xin, T. Mei, An improved RRT-based motion planner for autonomous vehicle in cluttered environments, in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, (2014), 4674–4679. https://doi.org/10.1109/ICRA.2014.6907542
8. W. Xu, Y. Yang, L. Yu, L. Zhu, A global path planning algorithm based on improved RRT, *Control Decis.*, **37** (2022), 829–838. https://doi.org/10.13195/j.kzyjc.2020.1354
9. X. G. Ruan, J. Zhou, J. J. Zhang, X. Q. Zhu, Robot goal guide RRT path planning based on sub-target search, *Control Decis.*, **35** (2020), 2543–2548. https://doi.org/10.13195/j.kzyjc.2019.0043
10. W. M. Zhang, S. X. Fu, Mobile robot path planning based on improved RRT* algorithm, *J. Huazhong Univ. Sci. Technol. (Nat. Sci. Ed.)*, **49** (2021), 31–36. https://doi.org/10.13245/j.hust.210101
11. X. Y. Zhong, J. Tian, H. S. Hu, X. F. Peng, Hybrid path planning based on safe a* algorithm and adaptive window approach for mobile robot in large-scale dynamic environment, *J. Intell. Rob. Syst.*, **99** (2020), 65–77. https://doi.org/10.1007/s10846-019-01112-z
12. X. Xu, X. Yu, Y. Zhao, C. Liu, X. Wu, Global path planning of mobile robot based on improved genetic algorithm, *Comput. Integr. Manuf. Syst.*, **28** (2022), 1659–1672. https://doi.org/10.13196/j.cims.2022.06.006
13. Z. Zhang, R. He, K. Yang, A bioinspired path planning approach for mobile robots based on improved sparrow search algorithm, *Adv. Manuf.*, **1** (2022), 114–130. https://doi.org/10.1007/s40436-021-00366-x
14. L. W. Yang, L. X. Fu, P. Li, J. L. Mao, N. Guo, An effective dynamic path planning approach for mobile robots based on ant colony fusion dynamic windows, *Machines*, **10** (2022), 2075–1702. https://doi.org/10.3390/machines10010050
15. K. Xu, H. Lu, Y. Huang, S. Hu, Robot path planning based on double-layer ant colony optimization algorithm and dynamic environment, *Acta Electron. Sin. (Chin.)*, **47** (2019), 2166–2176. https://doi.org/10.3969/j.issn.0372-2112.2019.10.019
16. H. D. Li, T, Zhao, S. Y. Dian, Forward search optimization and subgoal-based hybrid path planning to shorten and smooth global path for mobile robots, *Knowl.-Based Syst.*, **258** (2022), 110034. https://doi.org/10.1016/j.knosys.2022.110034
17. D. Ratner, M. Warmuth, Finding a shortest solution for the N × N extension of the 15-puzzle is intractable, *AAAI*, (1986), 168–172.
18. S. W. Lin, A. Liu, J. G. Wang, X. Y. Kong, A review of path-planning approaches for multiple mobile robots. *Machines*, **10** (2022), 773. https://doi.org/10.3390/machines10090773
19. L. P. Cheng, C. X. Liu, B. Yan, Improved hierarchical A-star algorithm for optimal parking path planning of the largeparking lot, in *2014 IEEE International Conference on Information and Automation(ICIA)*, (2014), 695–698. https://doi.org/10.1109/ICInfA.2014.6932742
20. Z. Q. Ren, S. Rathinam, H. Choset, Ms*: A new exact algorithm for multi-agent simultaneous multi-goalsequencing and path finding, in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, (2021), 11560–11565. https://doi.org/10.1109/ICRA48506.2021.9561779

21. L. C. Wen, Y. Liu, H. L. Li, CL-MAPF: Multi-Agent PathFinding for Car-Like robots with kinematic andspatiotemporal constraints, *Rob. Auton. Syst.*, **150** (2020), 103997. https://doi.org/10.1016/j.robot.2021.103997
22. J. B. Xin, L. Q. Wei, D. S. Wang, H. Xuan, Receding horizon path planning of automated guided vehicles using a time-space network model, *Optim. Control. Appl. Methods*, **41** (2020), 1889–1903. https://doi.org/10.1002/oca.2654
23. K. Murakami, Time-space network model and MILP formulation of the conflict-free routing problem of a capacitated AGV system, *Comput. Ind. Eng.*, **141** (2020). https://doi.org/10.1016/j.cie.2020.106270
24. M. Cap, P. Novak, A. Kleiner, M. Selecky, Prioritized planning algorithms for trajectory coordination of multiple mobile robots, *IEEE Trans. Autom. Sci. Eng.*, **12** (2015), 835–849. https://doi.org/10.1109/TASE.2015.2445780
25. N. Greshler, O. Gordon, O. Salzman, N. Shimkin, Cooperative multi-agent path finding: beyond path planning and collision avoidance, in *2021 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, (2021), 20–28. https://doi.org/10.1109/MRS50823.2021.9620590
26. H. I. Wang, Y. F. Li, W. J. Jiang, P. F. Wang, Q. X. Cao, Combined priority and path planning through a double-layer structure for multiple robots, in *2020 5th International Conference on Advanced Robotics and Mechatronics (ICARM)*, 2020. https://doi.org/10.1109/ICARM49381.2020.9195297
27. H. D. Li, T. Zhao, S. Y. Dian, Prioritized planning algorithm for multi-robot collision avoidance based on artificial untraversable vertex, *Appl. Intell.*, **52** (2022), 429–451. https://doi.org/10.1007/s10489-021-02397-0
28. W. Y. Wu, S. Bhattacharya, A. Prorok, Multi-robot path deconfliction through prioritization by path prospects, in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020. https://doi.org/10.1109/ICRA40945.2020.9196813
29. R. K. Dewangan, A. Shukla, W. W. Godfrey, A solution for priority-based multi-robot path planning problem with obstacles using ant lion optimization, *Mod. Phys. Lett. B*, **34** (2020), 2050137. https://doi.org/10.1142/S0217984920501377
30. H. J. Zhang, Y. D. Zhang, T. T. Yang, A survey of energy-efficient motion planning for wheeled mobile robots, *Ind. Rob.*, **47** (2020), 607–621. https://doi.org/10.1108/IR-03-2020-0063
31. S. Liu, D. Sun, Optimal motion planning of a mobile robot with minimum energy consumption, in *2011 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM)*, (2011), 43–48. https://doi.org/10.1109/AIM.2011.6027010
32. H. Zhang, Z. Su, D. E. Hernandez, B. Su, Energy optimal path planning for mobile robots based on improved AD* algorithm, *Trans. Chin. Soc. Agric. Mach.*, **49** (2018), 19–26. https://doi.org/10.6041/j.issn.1000-1298.2018.09.002
33. Y. Mei, Y. H. Lu, Y. C. Hu, C. S. G. Lee, Energy-efficient motion planning for mobile robots, in *IEEE International Conference on Robotics and Automation*, **5** (2004), 4344–4349. https://doi.org/10.1109/ROBOT.2004.1302401
34. V. Singh, R. K. Barai, P. Mandal, Real-time heuristic search based minimum energy path planning of wheeled mobile robot, in *Proceedings of the 2015 Conference on Advances in Robotics*, 2015. https://doi.org/10.1145/2783449.2783489
35. I. Noreen, A. Khan, Z. Habib, Optimal path planning using RRT* based approaches: a survey and future directions, *Int. J. Adv. Comput. Sci. Appl.*, **7** (2016), 97–107. https://doi.org/10.14569/IJACSA.2016.071114

36. C. W. Miao, G. Z. Chen, C. L. Yan, Y. Y. Wu, Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm, *Comput. Ind. Eng.,* **156** (2021). https://doi.org/10.1016/j.cie.2021.107230
37. G. H. Yi, Z. I. Feng, T. C. Mei, P. S. Li, W. Jin, S. Y. Chen, Multi-AGVs path planning based on improved ant colony algorithm, *J. Supercomput.*, **75** (2019), 5898–5913. https://doi.org/10.1007/s11227-019-02884-9
38. C. Ntakolia, D. Lyridis, A comparative study on ant colony optimization algorithm approaches for solving multi-objective path planning problems in case of unmanned surface vehicles, *Ocean Eng.*, **255** (2022), 111418. https://doi.org/10.1016/j.oceaneng.2022.111418
39. J. Faiz, F. Parvin, Trends and technical advancements on high-efficiency electric motors: a review, *Effic. Complex Syst.*, 2022, 81–95. https://doi.org/10.1007/978-3-030-69288-9_5
40. S. B. Santra, A. Chatterjee, D. Chatterjee, S. Padmanaban, K. Bhattacharya, High efficiency operation of brushless dc motor drive using optimized harmonic minimization based switching technique, *IEEE Trans. Ind. Appl.*, **58** (2022), 2122–2133. https://doi.org/10.1109/TIA.2022.3146212
41. G. J. Liu, P. Liu, W. I. Mu, S. J. Wang, A path optimization algorithm for AUV using an improved ant colony algorithm with optimal energy consumption, *J. Xian Jiaotong Univ.*, **50** (2016), 93–98. https://doi.org/10.7652/xjtuxb201610014
42. Q. Luo, H. B. Wang, Y. Zheng, J. C. He, Research on path planning of mobile robot based on improved ant colony algorithm, *Neural Comput. Appl.*, **32** (2020), 1555–1566. https://doi.org/10.1007/s00521-019-04172-2
43. J. Liu, L. P. He, Y. Y. Wang, H. F. Liang, Generalized path planning method for mobile medical robots integrating energy efficiency with safety factors, *Comput. Integr. Manuf. Syst.*, (2021), 1–15. https://doi.org/10.1109/PHM-Shanghai49105.2020.9280948
44. S. Liu, D. Sun, Minimizing energy consumption of wheeled mobile robots via optimal motionplanning, *IEEE/ASME Trans. Mechatron.*, **19** (2014), 401–411. https://doi.org/10.1109/TMECH.2013.2241777
45. Y. J. Wang, C. D. Chen, C. K. Sung, System design of a weighted-pendulum-type electromagnetic generator for harvesting energy from a rotating wheel, *IEEE/ASME Trans. Mechatron.*, **18** (2012), 754–763. https://doi.org/10.1109/TMECH.2012.2183640
46. Y. Mei, Y. Lu, Deployment of mobile robots with energy and timing constraints, *IEEE Trans. Rob.*, **22** (2006), 507–522. https://doi.org/10.1109/TRO.2006.875494
47. M. Erdmann, T. Lozano-Perez, On multiple moving objects, *Algorithmica*, **2** (1987), 477–521. https://doi.org/10.1109/ROBOT.1986.1087401
48. J. P. V. D. Berg, M. H. Overmars, Prioritized motion planning for multiple robots, in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (2005), 430–435. https://doi.org/10.1109/IROS.2005.1545306
49. K. X. Zhang, J. L. Mao, Z. W. Xuan, F. H. Xiang, L. X. Fu, Hierarchical scheduling based multi-agent path finding for pass Terrain, *Comput. Integr. Manuf. Syst. (Chin.)*, (2022), 1–16.
50. L. W. Yang, L. X. Fu, N. Guo, Z. Yang, H.Q. Guo, Path planning with multi-factor improved ant colony algorithm, *Comput. Integr. Manuf. Syst. (Chin.)*, (2021), 1–18.
51. G. Wagner, H. Choset, Subdimensional expansion for multirobot path planning, *Artif. Intell.*, **219** (2015), 1–24. https://doi.org/10.1016/j.artint.2014.11.001

52. Y. C. Sun, X. I. Zhao, Y. Z. Yu, Research on a random route-planning method based on the fusion of the A* algorithm and dynamic window method, *Electronics*, **11** (2022), 2683. https://doi.org/10.3390/electronics11172683