International Journal of Industrial Engineering Computations 16 (2025) 1089–1100

Contents lists available at GrowingScience

International Journal of Industrial Engineering Computations

homepage: www.GrowingScience.com/ijiec

Research on collaborative scheduling method for multi-robot tomato picking based on improved particle swarm optimization algorithm

Tao Ding[a], Shichao Wang[a], Guohua Gao[a*] and Xudong Zhao[a]

[a]*College of Mechanical and Energy Engineering, Beijing University of Technology, Beijing, P.R. China*

C H R O N I C L E

Article history:
Received March 6 2025
Received in Revised Format May 10 2025
Accepted June 27 2025
Available online June 27 2025

Keywords:
Multi-robot Collaboration
Agricultural Machinery
Task Allocation
Picking Robot
Transport Robot

A B S T R A C T

Currently, multi-robot cooperative algorithms are widely used in the field of agriculture, which greatly improves the efficiency of agricultural production. However, the multi-robot cooperative operation of agricultural machinery is mostly limited to the efficiency and accuracy of scheduling. To address the mentioned shortcomings, a novel multi-task scheduling method based on improved particle swarm optimization algorithm is proposed, which is applied to the efficient collaborative scheduling problem of tomato picking robots and transfer vehicles in greenhouse cultivation of different scales. Firstly, the scene of collaborative scheduling between tomato automatic picking and transshipment is described, and the mathematical model of multi-machine collaborative scheduling is established with the shortest waiting time of picking robots and the minimum number of transshipment vehicles as the optimization objectives. Secondly, an improved particle swarm optimization algorithm is expounded in detail, which customizes the fitness function and enhances the particle update strategy. Finally, the experimental results show that the improved particle swarm optimization algorithm can not only determine the optimal number and execution order of cooperative robots, but also reduce the task execution time by 47% compared with the unimproved method.

1. Introduction

Currently, multi-robot collaboration of agricultural machinery is one of the main research topics in the field of agricultural machinery. Reasonable arrangement of cooperative operation of multiple agricultural machinery or various types of agricultural machinery can become an important way to reduce the cost and increase the efficiency of agricultural machinery (Chen et al., 2021; Li et al., 2023; Ma et al., 2022; Zhi-qiang et al., 2021). However, due to the complexity and diversity of the agricultural environment, there is a lack of multi-robot collaborative scheduling algorithms for tomato picking and transportation in the greenhouse environment, which greatly reduces the efficiency of tomato harvesting.

In modern agricultural production, greenhouse facilities have become an important place for efficient cultivation of vegetables and fruits due to their unique environmental regulation advantages (Gorjian et al., 2021). However, many operational tasks inside greenhouses, such as picking and transportation of crops such as tomatoes, still rely on manual labor. This not only consumes a large amount of labor but also results in low efficiency, failing to meet the increasing market demands. Therefore, there is an urgent need for automated and intelligent solutions to enhance the efficiency and quality of greenhouse operations. The application of transport robots (Ansari & Karayiannidis, 2021; Kaltsas et al., 2022) can significantly improve efficiency and reduce labor intensity. However, the operational efficiency and application scope of a single agricultural robot (Jin et al., 2021; Rovira-Mas et al., 2021) are limited, making it challenging to meet the demands of large-scale operations. Therefore, it is particularly important to give full play to the advantages of scheduling algorithms for multi-robot (Zhang & Noguchi,

---

* Corresponding author

E-mail gaoguohua@bjut.edu.cn (G. Gao)

ISSN 1923-2934 (Online) - ISSN 1923-2926 (Print)
2025 Growing Science Ltd.
doi: 10.5267/j.ijiec.2025.6.013

2017) cooperative systems. Aiming at the above problem, this study comprehensively analyzes the tomato picking and transport process in the greenhouse environment, combs the cooperation problem of picking robot and transport robot, and constructs the objective function with waiting time and the number of transport robots as the optimization objective. In addition, an improved particle swarm optimization algorithm is proposed to solve the problem of multi-robot cooperation between picking robots (Su et al., 2023; Xiang et al., 2024) and transport robots in tomato automatic picking.

The remaining sections of this paper are structured as follows: Section 2 provides a detailed review of related work. Section 3 introduces assumptions and relevant constraints for the multi-transport robot scheduling model. Section 4 models the design of multi-transport robot scheduling using the improved particle swarm optimization algorithm. Section 5 presents experimental validation and simulations. Finally, Section 6 concludes the main contributions of this work and proposes future research directions.

2. Related Work

In the past few decades, many experts have studied the scheduling algorithm and made it fully developed. Liu (2017) proposed a multi-objective warehouse material transportation system scheduling strategy based on a cloud model to solve the problem of how to determine the priority of tasks. Xiao et al. (2012) proposed a real-time multi-attribute task scheduling strategy for automated guided vehicle systems in manufacturing environments, which assigns tasks by evaluating task urgency combined with distance. Different from the above, some scholars regard the shortest time as the optimization goal while considering the rationality of task allocation. Yaun et al. (2018) studied the "goods-to-person" order picking mode based on AGVs, and established a scheduling model with the goal of minimizing completion time. Teng et al. (2023) transformed the automated port AGV horizontal transportation scheduling problem into a parallel machine scheduling problem and developed an improved artificial bee colony algorithm, so as to minimize transportation time. While considering the rationality of task allocation, some relevant scholars also consider low cost as the optimization goal. Dang et al. (2021) studied the transportation scheduling problem for multi-function robots with battery management, and minimized the delay costs and transportation costs.

The existing research on the multi-robot cooperative problem of agricultural machinery mainly addresses two aspects: path planning and task allocation. Cao et al. (2023)used time windows to detect different types of conflicts after planning the path to address the problem of path conflicts in multi-robot cooperative path planning of agricultural machinery. Shi et al. (2023) used unmanned aerial vehicles (UAVs) to collect field information, which was then used to detect field obstacles and assist agricultural machinery in avoiding field obstacles. Zhai et al. (2021) performed path planning for master-slave agricultural machinery, where the paths for slave operations were formulated based on the relative distances of the master and slave and the turning states to avoid collisions. Jiang et al. (2022) proposed an adaptive immune following algorithm based on the immune algorithm and artificial fish swarm algorithm to assign fields with a different amount of work to the agricultural machinery group. Wang et al. (2021) divided the path planning methods between the fields into several categories according to the positional relationship between the fields and assigned the fields to the agricultural machinery group accordingly. Gong et al. (2021) divided a field into multiple areas according to the obstacles in the field and then allocated these areas to the agricultural machinery group. Zhu et al. (2024) studied the collaborative operation of different fertilizer applicators in the field ,coordinated fertilizer delivery for fertilizer trucks and developed the CCFWA algorithm.

In short, although there is quite a lot of research on multi-robot cooperative scheduling algorithms, the research on multi-task allocation of random numbers of devices is relatively scarce. Therefore, this paper proposes an improved particle swarm optimization algorithm, which takes the number of transport robots and task waiting time as the optimization objectives, and realizes the random task allocation for different numbers of transport robots.

3. Establishment of multi-robot scheduling model and the improvement of particle swarm optimization algorithm

3.1 Greenhouse picking robot and transport robot collaborative scheduling scenario

In the greenhouse, picking robots perform partition operations to complete the task of harvesting tomatoes. The transport robot is responsible for transporting the tomatoes picked by the picking robot to the designated unloading point in the greenhouse. Due to the limited capacity of the picking robot collection unit, it is necessary to coordinate the transportation robots among multiple picking robots to prevent the situation that the picking robot is filled but the tomato cannot be transported in time. The greenhouse map is depicted in Fig. 1, where black areas represent tomato cultivation zones. The picking robots pick tomatoes in the picking area. Before the picking robot reaches the limit capacity, the transport robot goes to the tomato receiving point to receive the tomatoes. Upon completion, they return to designated tracks to await subsequent tasks. Once they reach capacity, transport robots proceed to the unloading area for tomato discharge.

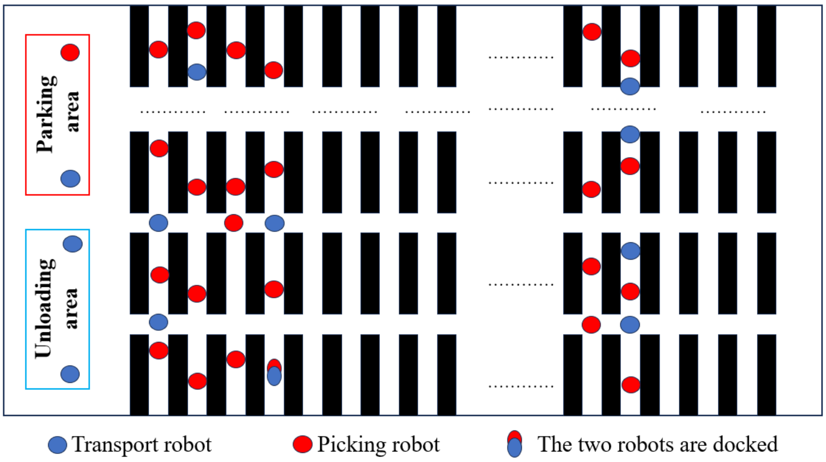

Fig. 1. Greenhouse grid map

3.2 Collaborative Scheduling Modeling

1. Problem analysis and hypothesis

The scheduling problem of tomato transport robots in the greenhouse can be formulated as follows: Initially, all transport robots are stationed in the parking area. Based on greenhouse conditions, the tomato picking robot releases n sets of tasks to be performed $T = \{T_1, T_2, T_3, \dots T_n\}$. Each tomato transport task T_i includes task details such as task number, release time, and pickup location. The greenhouse accommodates a total of m tomato transport robots, denoted as $Z = \{Z_1, Z_2, Z_3, \dots Z_m\}$.Each tomato transport robot contains the following information, including robot number, travel speed, current location, and starting location. Each transport robot departs from its initial point, reaches the designated task execution site, and awaits tomato reception. Afterward, it returns to the central aisle corresponding to its current track and awaits the next task allocation. When the internal capacity of the transport robot reaches the maximum, the transport robot begins to reach the unloading point for unloading. In order to establish the collaborative scheduling model between transport robots and picking robot, this study makes the following assumptions:

(1) The exact locations of all areas within the greenhouse, including parking areas, unloading areas, tomato plant rows, and aisles, are known;
(2) All transport robots have uniform carrying capacities, set at three times the capacity of the picking robots;
(3) Once the transportation robot initiates the task, it cannot be interrupted;
(4) The operation of robots is conducted without consideration of battery levels;
(5) In the greenhouse, all transport robots operate at a fixed speed, and the time taken for both receiving and unloading tomatoes remains constant;
(6) The robots in the greenhouse switch between wheel sets to turn corners, with a fixed time allocated for this maneuver;
(7) After completing the current transport task, the transport robot needs to return to the middle aisle position near the track entrance and wait for the next transport task.

2. Restrictive conditions

Based on the definitions and assumptions provided above, additional constraints are introduced to ensure the feasibility and effectiveness of the model:

(1) Grid occupancy constraint

In the process of carrying out the task of the transport robot, two or more transport robots cannot occupy the same grid. Only when the current transport robot leaves this grid, other transport robots can occupy this grid.

$$\sum_{g=1}^{n} z_{p_k} \le 1, p = 1, 2, \dots g \tag{1}$$

where $z_{pk} = 1$ represents that the p-th grid is occupied by the k-th transport robot; g is the total number of greenhouse map grids.

(2) Transportation task matching constraints

The task matching constraint guarantees that each task is assigned to a transport robot once and can only be performed once. And each transport robot can only perform one task at a time to avoid task execution conflicts.

$$\sum_{k=1}^{n} b_{ik} = 1, i = 1,2...m \tag{2}$$

$$\sum_{i=1}^{m} c_{ijk} = x_{jk}, j = 1,2...m.k = 1,2...n \tag{3}$$

$$\sum_{j=1}^{m} c_{ijk} = x_{ik}, i = 1,2...m.k = 1,2...n \tag{4}$$

where $\mathrm{b}_{ik} = 1$ represents that the k-th transport robot performs the i-th task; $c_{ijk} = 1$ represents that the k-th transport robot performs the j-th task after performing the i-th task; m is the number of transport task,

(3) Number constraint of transport robots

The number of transport robots running simultaneously in any time period must be less than the total number of all transport robots in the greenhouse. Ensure that the task allocation will not exceed the actual available number of transport robots in the current system, and prevent the execution failure caused by over-allocation.

$$n(t) \le n \tag{5}$$

where $n(t)$ is the number of transport robots running in the greenhouse at mid-moment; n is the total number of transport robots in the greenhouse.

(4)Transport robot operation time constraints

In the scheduling system, tomato transport tasks are initiated by picking robots. Each transport task is assigned a specific time. To ensure smooth tomato picking and transport operations, the transport robot must start a new transport task at a time greater than the completion time of the previous task plus the travel time from its current position to the new task point.

$$RT_{k,i-1} + Z_{k,i-1,i} \le T_{ki} \tag{6}$$

where $RT_{k,i-1}$ is the actual transport time of the *k*-th transport robot for the *(i-1)*-th transport task; $Z_{k,i-1,i}$ is the travel time of the *k-th* transport robot from the *(i-1)*-th transport task to the *i*-th transport task; T_{ki} is the release time of the *k*-th transport robot for the *i*-th transport task.

(5) Start and end point constraints of the transport robot

In the scheduling system, once a transport task is assigned, the transport robot departs from a centralized parking area. After completing all assigned tasks, it returns to this parking area. Therefore, the starting point and end point of the transport robot to perform the transportation task are the parking area.

$$F_{k_1} = P \tag{7}$$

$$E_{k_1} = P \tag{8}$$

where F_{k_1} is the first task of the k_1-th transport robot; E_{k_1} is the last task of the k_1-th transport robot; P is the location of the centralized parking area.

3. Establishment of multi-transport robot scheduling model

The total task waiting time and the number of transport robots are taken as the optimization objectives. The optimization of the total task waiting time can improve the overall efficiency of the greenhouse operation. The optimization of the number of transport robots can reduce the collision and congestion caused by too many robots. The following is the multi-objective modeling process.

(1) The shortest total waiting time

All transport tasks issued by picking robots need to be promptly executed, as shorter task waiting times enhance the operational efficiency of the picking robots.

$$F_1 = min\left\{\sum_{i=1}^{n}\left(RT_{t_n} - T_{t_n}\right)\right\} \tag{9}$$

where T_{tn} is the release time of the n-th transport task; RT_{t_n} is the actual transport time of the n-th transport task; n is the number of transport robots.

(2) The shortest number of transport robots

$$F_2 = \min(n) \tag{10}$$

Using the weighted sum method, assign weights to each objective to transform the multi-objective problem into a single-objective problem. Construct an objective function based on minimizing total task waiting time and minimizing the number of transport robots:

$$F = \varphi_1 \min\left\{\sum_{i=1}^{n}\left(RT_{t_n} - T_{t_n}\right)\right\} + \varphi_2 \min(n) \tag{11}$$

where φ_1 is the shortest waiting time weight coefficient; φ_2 is the shortest number of transport robots weight coefficient.

3.3 Overview of improved particle swarm optimization algorithm

Traditional particle swarm optimization (PSO) algorithms are commonly applied in fields such as engineering design optimization, data mining, neural network training, image processing, and control engineering. Research in the area of scheduling and task allocation is relatively limited. Moreover, the particle update strategy of the traditional particle swarm optimization algorithm is too direct, which makes it easy to fall into the local optimal solution in the process of iteration. To solve this problem, a roulette wheel selection mechanism is proposed to update particles.

Firstly, initialize a random population of particles with assigned task execution sequences. Execute tasks according to the initialized particle sequences and compute the fitness values for each particle. Determine the best fitness value *pbest*, among all particles and compute the global best fitness value *gbest*, for the particle swarm. Utilize the distance between transport robots and the next task point to convert into selection probabilities. Select appropriate transport robots based on these probabilities. Evolve particle positions using this method and adjust random learning parameters dynamically based on iteration parameters to accelerate algorithm convergence. When reaching the preset maximum iteration count, terminate the particle swarm algorithm's updates to complete the scheduling problem solution. The algorithm flow is illustrated in Fig. 2.

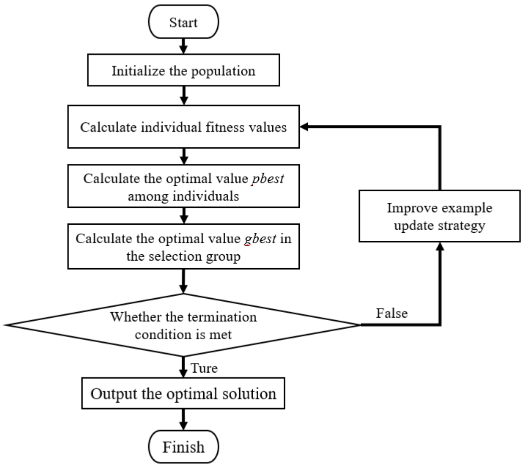

Fig. 2. Improved particle swarm optimization algorithm diagram

3.4 Particle swarm optimization update method

1. Initialize the population

According to the characteristics of the transport robot task, a particle population with M particles is initialized. Each particle is a one-dimensional array, in which the tasks are sorted by time, and each position in the array corresponds to a task number. The value at each position indicates the transport robot assigned to execute that task. For example, in the array [1,1,1,2,3,1],the first position indicates that task 1 is executed by transport robot 1, the second position indicates that task 2 is also executed by transport robot 1, and so on. This configuration allows us to determine the set of tasks performed by each transport robot. For instance, in a greenhouse with three transport robots numbered 1, 2, and 3, and a total of 10 tasks, the initial population could be represented as [1,1,1,2,3,1,2,1,3,2].

2. Evaluating individual fitness

The particles defined in the algorithm, such as [1,1,1,2,3,1,2,1,3,2] ,do not directly provide the number of transport robots. Therefore, it is necessary to include a process in each particle to determine the number of transport robots, as shown in Fig. 3.

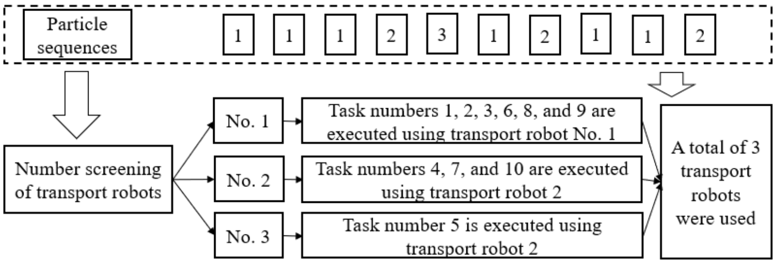

Fig. 3. Flow chart for solving the number of transport robots

Based on the solving process described above, we can obtain the number of transport robots used by each particle and the total waiting time for all tasks to complete. By introducing appropriate penalty coefficients, we derive the fitness value for each particle. Comparing fitness values allows us to find the optimal system solution. The fitness function is expressed by the following equation:

$$Fitness = 10000 \times Robot\_numbers + Waiting\_time \tag{12}$$

3. Improved particle update strategy

The update strategy optimizes the allocation of transport robots to minimize the total waiting time for multiple transport tasks and the number of transport robots. The improved particle update strategy employs a distance-based selection mechanism to dynamically adjust the probability of assigning specific tasks to transport robots.

(1) Initialization Phase: Each position of the particle represents the assignment state of a task. The length of each particle is equal to the total number of tasks, and the value range of its elements is the number of the transport robots.

(2) Distance Calculation: After determining the specific positions of tasks and robots, the distance d_{iv} is obtained by calculating the Euclidean distance between the two position coordinates. Assuming task i is located at the two−dimensional coordinates $\boldsymbol{x}_i = (x_i, y_i)$ and robot v is located at coordinates $\boldsymbol{p}_v = (p_{vx}, p_{vy})$the Euclidean distance d_{iv} can be calculated as follows:

$$d_{iv} = \sqrt{(x_i - p_{vx})^2 + (y_i - p_{vy})^2} \tag{13}$$

(3) Distance Conversion to Selection Probability: According to the calculated distance, a selection probability is defined so that the robot with a shorter distance has a higher probability of being selected. The exponential function is used to convert the distance into probability to enhance numerical stability. The calculation formula is as follows:

$$P_v = \frac{\exp(-\alpha d_{iv})}{\sum_{v=1}^{V} \exp(-\alpha d_{iv})} \tag{14}$$

where α is a positive proportionality coefficient used to adjust the sensitivity of distance to probability; V is the total number of transport robots.

(4) Roulette Wheel Selection Mechanism: The cumulative probability of each robot is calculated, and a random number r is generated. The first robot with cumulative probability greater than r is selected. If all the cumulative probabilities do not exceed r (rare cases), the nearest robot is directly selected.

(5) Update Particle Position: Based on the above process, the position vector of the particle is updated.

The above improved method can effectively reflect the distance relationship between the transport robot and the task, so as to optimize the overall task execution efficiency.

4.Verification and analysis of multi-transport robots scheduling algorithm

4.1 Experimental environment construction

Due to the environmental characteristics in the greenhouse and the capacity limitation of the tomato collection device of the picking robot, the picking robot will reach the limit capacity when picking to the midpoint of each picking area, so the middle position of each picking area in the greenhouse is set as the receiving point of the transportation task in advance. Fig. 4 shows the positions of transport points when four picking robots operate simultaneously. The robots start from the left side, using the aisles as starting points, and pick upwards and downwards on both sides. Two robots adjacent on one side operate simultaneously. After a robot completes picking a column of tomatoes, it exits the current track and switches to an adjacent idle position on the same side (spaced by one column). Different colors (red, blue, black, green) in the figure denote different picking tasks, with markers of the same color indicating positions where the same robot operates on tomato transport points. Fig. 5 shows the electric grid map generated by MATLAB, in which the task points are numbered from left to right, and the starting point of the transport robot and the tomato unloading point are specifically marked on their respective positions on the map.

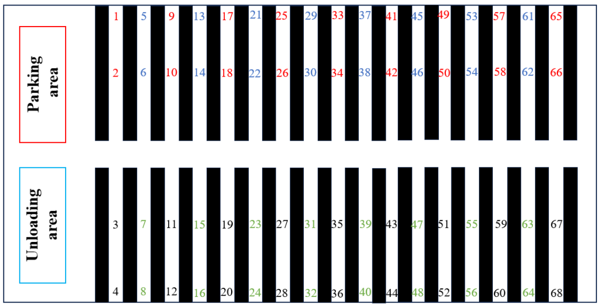

Fig. 4. Location map of tomato transportation receiving point

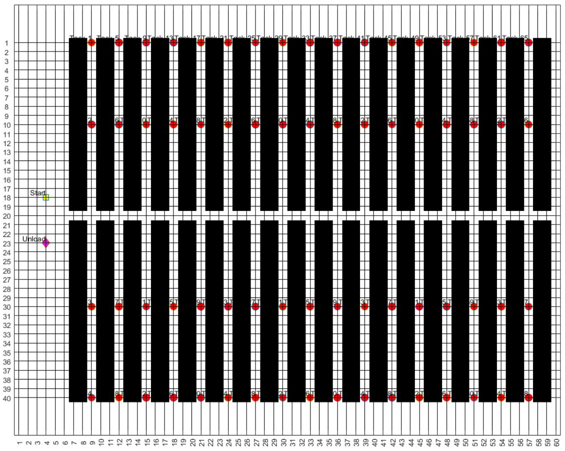

Fig. 5. Number diagram of tomato transportation receiving point

The tomato picking robot takes an average of 10 seconds to pick one tomato, and its collection unit can hold up to 80 tomatoes. Therefore, it takes approximately 14 minutes to fill the tomato box. Due to the irregular distribution of tomatoes, the arrival time at tomato transport points fluctuates within a range of 12 to 16 minutes. Considering the time consumed by the robot for forward movement, the time spent at tomato transport points ranges from 15 to 20 minutes. Additionally, after completing the two tasks, the picking robot needs to move to the next track for picking, and the orbit change time of the picking robot is defined as 300 seconds. When the scheduling algorithm is initialized, neither the start time nor the end time is defined, and all tasks are marked as unfinished. The preliminary results are summarized in Table 1.

Table 1
Table of tomato transportation task information

Task Number	Release Time	Pick-up Point	Execution Time	Waiting Time	Completed
1	00:37:20	(9,1)	/	/	False
2	00:17:00	(9,10)	/	/	False
3	00:18:20	(9,30)	/	/	False
4	00:40:30	(9,40)	/	/	False
5	00:38:10	(12,1)	/	/	False
6	00:16:10	(12,1)	/	/	False
7	00:17:30	(12,3)	/	/	False
8	00:40:00	(12,4)	/	/	False
9	01:14:00	(15,1)	/	/	False
10	00:53:30	(15,1)	/	/	False
…	…	…	…	…	…
61	05:11:50	(54,1)	/	/	False
62	04:48:10	(54,1)	/	/	False
63	05:06:00	(54,3)	/	/	False
64	05:27:30	(54,4)	/	/	False
65	05:52:00	(57,1)	/	/	False
66	05:30:00	(57,1)	/	/	False
67	05:38:30	(57,3)	/	/	False
68	06:00:00	(57,4)	/	/	False

4.2 Algorithm modeling and parameter setting

Before inputting all transportation task information into the scheduling algorithm, it is necessary to set various parameters of the particle swarm algorithm, such as the population size of particles, maximum number of iterations, transport robot speed, and other relevant parameters. The table detailing algorithm parameters is presented as Table 2.In the improved particle swarm optimization algorithm, the update strategy for particles is outlined in Table 3.

Table 2
Table of algorithm parameter information

Parameter Name	Parameters
Particles	68、80、92
Maximum Number of Iterations	1000
Transport robot Speed	0.1m/s
Loading Time	10s
Unloading Time	30s

Table 3
Improved particle update method of particle swarm optimization algorithm

```
Algorithm  Update Particle
Require: particle, locations, numRobots
Ensure: Updated particle with assigned robot positions
numTasks ← length(particle.positions)
for i ← 1 to numTasks do
    taskLocation ← locations[i]
    distances ← array of size numRobots
    for v ← 1 to numRobots do
        robotLocation ← particle.robots[v].currentPosition
        distances[v] ← calculateDistance(taskLocation, robotLocation)
    end for
    probabilities ← array of size numRobots
    for v ← 1 to numRobots do
        probabilities[v] ← exp(-distances[v] / max(distances))
    end for
    normalize probabilities to sum to 1
    cumulativeProbabilities ← compute cumulative probabilities from probabilities
    r ← random number between 0 and 1
    selectedRobot ← find vehicle index where cumulativeProbabilities > r
    if no selectedRobot then
        selectedRobot ← find robot index with minimum distance
    end if
    particle.positions[i] ← selectedRobot
end for
```

4.3 Solving result

Based on Fig. 5-1, the greenhouse dimensions and task quantities are expanded to evaluate the algorithm's performance across three different sizes: 68 tasks in a 42m × 120m greenhouse, 80 tasks in a 42m × 140m greenhouse, and 92 tasks in a 42m × 160m greenhouse. Using 68 tasks as an example, the task information is inputted into the scheduling algorithm, and the results are depicted by the blue line in Fig. 5-3 to Fig. 5-6. After four comparative analyses, although the initialization of each particle is different, a consistent scheduling result is still obtained. The optimal task sequence was identified as[2,4,3,1,2,3,1,2,1,3,2,4,4,3,1,3,2,3,1,3,4,4,3,3,3,1,1,1,1,2,3,2,1,2,2,1,1,4,3,3,2,3,2,3,2,1,1,4,2,1,1,1,2,4,4,3,3,1,3,4,2,3,4,1,3,1,2,3].

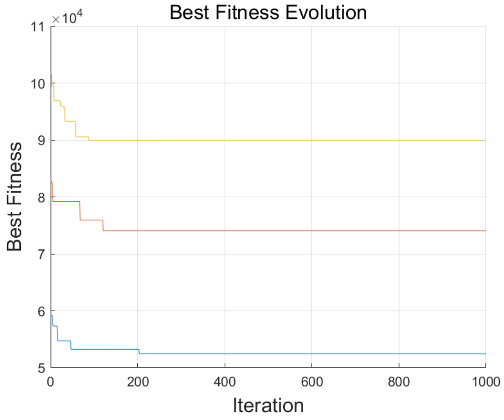

Fig. 6. Graph of first iteration

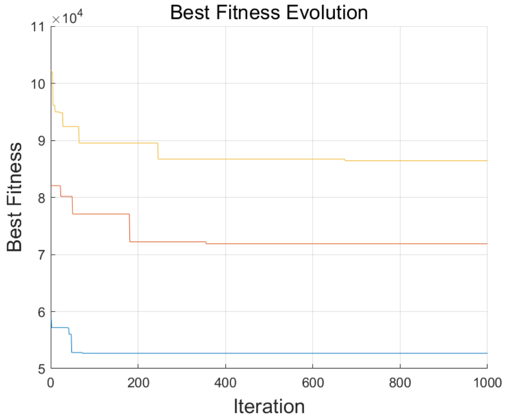

Fig.7. Graph of second iteration

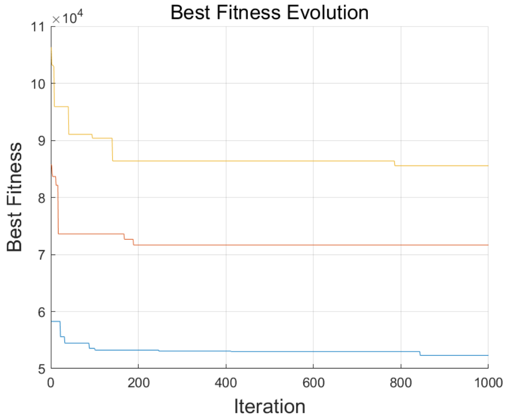

Fig. 8. Graph of third iteration

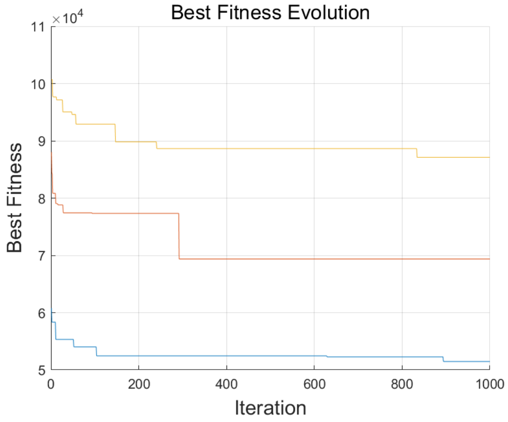

Fig. 9. Graph of fourth iteration

The optimal number of transport robots used is 4, with a total waiting time for all tasks of 11440 seconds and an average waiting time of 168 seconds. The convergence curves for fitness values under greenhouse environments with 80 and 92 tasks are represented by the red and yellow lines, respectively, in the figures. The experimental results are shown in Fig. 6 to Fig. 9. Since the task allocation of transport robots in the greenhouse constitutes a discrete problem, the iterative experimental solution of the above scheduling algorithms reveals that the fitness convergence curve is not smooth and continuous, but a ladder diagram. The three solutions converged within the predefined maximum number of iterations, and the optimal fitness values consistently remained within a certain range, indicating that the improved particle swarm optimization algorithm exhibits high stability.

4.4 Result analysis

In order to verify the superiority of the improved particle swarm optimization algorithm, we compare it with other algorithms. The comparison results are shown in Table 4.

Table 4
Table of algorithm comparison results

Algorithm name	**Total waiting time**	**Number of transport robots**	**Maximum Waiting Time for a single time**	**Average Waiting Time**
Improved particle swarm optimization algorithm	11440s	4	2220s	168s
Particle swarm algorithm considering only the number of transport robots	19400s	3	1810s	285s
Particle swarm optimization algorithm considering only waiting time	9670s	8	1450s	142s

From the comparative analysis of the operational results of the three algorithms in Table 4, it can be observed that the improved particle swarm optimization algorithm, compared to the particle swarm algorithm that only considers the number of transport robots, reduces waiting time by approximately 70%. Furthermore, compared to the algorithm that only considers task consumption waiting time, it saves 4 transport robots. This demonstrates that the improved particle swarm optimization algorithm not only considers the number of transport robots but also takes into account the overall operational task consumption waiting time, which not only improves the efficiency of task execution but also reduces the cost of task execution. Next, a comparison is conducted on the total task consumption time, maximum single waiting time, and average waiting time for different numbers of transport robots. This comparison enables us to assess the impact of the number of transport robots and verify the best results. The comparative results are presented in Table 5.

Table 5
Comparison of time for different number of transport robots

Number of transport robots	Total consumption waiting time	Maximum waiting time for a single time	Average waiting time
3	19400s	1810s	285s
4	11440s	2220s	168s
5	8200s	1670s	120s
6	4830s	2450s	71s
7	3050s	1360s	44s
8	9670s	1450s	142s

Upon observation, it is noted that when there are 7 transport robots, both the total consumption waiting time, maximum single waiting time, and average waiting time are minimized. However, considering the number of transport robots and the total waiting time, and observing the change of total waiting time when the number of transport robots increases, 4 transport robots are better than 7 transport robots.

Finally, based on the current improved particle swarm optimization algorithm, the particle update strategy is modified to select transport robots based on distance, specifically choosing the nearest transport robot when assigning tasks. Through the comparison of two particle update strategies, as shown in Table 6, applying the improved particle swarm optimization algorithm for multi-robot collaboration of transport robots significantly reduces waiting time.

Table 6
Particle Update Strategy Comparison

Particle update strategy	Number of transport robots	Total consumption waiting time	Average waiting time
Roulette Mechanics	4	11440	168
Distance of transport robot	4	21840	321

5. Conclusion

This study establishes a mathematical model for cooperative scheduling of multiple transport robots in greenhouse environments and proposes an improved particle swarm optimization algorithm for solving it. The main contributions include:

(1) Establishing a collaborative scheduling model for multiple tomato picking robots and transport robots in greenhouses to enhance multi-robot collaboration efficiency.

(2) Introducing a scheduling design method based on the improved particle swarm optimization algorithm, which enhances particle update strategies and addresses the issue of the algorithm's susceptibility to local optima by proposing a roulette wheel selection mechanism for particle updates.

(3) Validating the effectiveness of the improved particle swarm optimization algorithm through experimental simulations under selected operational modes. The experimental results demonstrate the rationality and effectiveness of the improved particle swarm optimization algorithm.

Future research can focus on the real greenhouse picking environment, explore the collaborative picking operation of different vegetables, and fully consider the coordination between picking robots, transportation robots, and transportation robots themselves.

Acknowledgements

This research was supported by the National Key R&D Program of China (2023YFD2000013). The authors would like to thank the editor and the anonymous reviewers for their constructive and helpful comments, which helped to improve the presentation of the paper.

References

Ansari, R. J., & Karayiannidis, Y. (2021). Task-Based Role Adaptation for Human-Robot Cooperative Object Handling [Article]. *Ieee Robotics and Automation Letters, 6*(2), 3592-3598.

Cao, R., Guo, Y., Zhang, Z., Li, S., Zhang, M., Li, H., & Li, M. (2023). Global path conflict detection algorithm of multiple agricultural machinery cooperation based on topographic map and time window [Article]. *Computers and Electronics in Agriculture, 208*, Article 107773.

Chen, C., Hu, J., Zhang, Q., Zhang, M., Li, Y., Nan, F., & Cao, G. (2021). Research on the Scheduling of Tractors in the Major Epidemic to Ensure Spring Ploughing [Article]. *Mathematical Problems in Engineering, 2021*, Article 3534210.

Dang, Q.-V., Singh, N., Adan, I., Martagan, T., & van de Sande, D. (2021). Scheduling heterogeneous multi-load AGVs with battery constraints [Article]. *Computers & Operations Research, 136*, Article 105517.

Gong, J., Wang, W., Zhang, Y., & Lan, Y. (2021). Cooperative working strategy for agricultural robot groups based on farmland environment. *Transactions of the Chinese Society of Agricultural Engineering (Transactions of the CSAE), 37*(2), 11-19.

Gorjian, S., Calise, F., Kant, K., Ahamed, M. S., Copertaro, B., Najafi, G., Zhang, X., Aghaei, M., & Shamshiri, R. R. (2021). A review on opportunities for implementation of solar energy technologies in agricultural greenhouses [Review]. *Journal of Cleaner Production, 285,* Article 124807.

Jiang, Y., Li, X., Yang, Z., Zhang, X., & Wu, H. (2022). An adaptive immune-following algorithm for intelligent optimal schedule of multiregional agricultural machinery [Article]. *International Journal of Intelligent Systems, 37*(11), 9404-9423.

Jin, Y., Liu, J., Xu, Z., Yuan, S., Li, P., & Wang, J. (2021). Development status and trend of agricultural robot technology [Article]. *International Journal of Agricultural and Biological Engineering, 14*(4), 1-19.

Kaltsas, P. I., Koustoumpardis, P. N., & Nikolakopoulos, P. G. (2022). A Review of Sensors Used on Fabric-Handling Robots [Article]. *Machines, 10*(2), Article 101.

Liu, S. (2017). Multi-dispatching method for warehouse material handling system. *Computer Engineering and Applications, 53*(18), 224-229.

Li, S., Zhang, M., Wang, N., Cao, R., Zhang, Z., Ji, Y., Li, H., & Wang, H. (2023). Intelligent scheduling method for multi-machine cooperative operation based on NSGA-III and improved ant colony algorithm [Article]. *Computers and Electronics in Agriculture, 204,* Article 107532.

Ma, L., Xin, M., Wang, Y.-J., & Zhang, Y. (2022). Dynamic Scheduling Strategy for Shared Agricultural Machinery for On-Demand Farming Services [Article]. *Mathematics, 10*(21), Article 3933.

Rovira-Mas, F., Saiz-Rubio, V., & Cuenca-Cuenca, A. (2021). Augmented Perception for Agricultural Robots Navigation [Article]. *Ieee Sensors Journal, 21*(10), 11712-11727.

Shi, M., Feng, X., Pan, S., Song, X., & Jiang, L. (2023). A Collaborative Path Planning Method for Intelligent Agricultural Machinery Based on Unmanned Aerial Vehicles [Article]. *Electronics, 12*(15), Article 3232.

Su, L., Liu, R., Liu, K., Li, K., Liu, L., & Shi, Y. (2023). Greenhouse Tomato Picking Robot Chassis [Article]. *Agriculture-Basel, 13*(3), Article 532.

Teng, H., Zhuang, Z., Huang, Z.,Qing, W.,Qing, T., & Zou, Y. (2023). Improved Artificial Bee Colony Algorithm for Horizontal Transportation Scheduling in Automated Terminals. *Computer Integrated Manufacturing Systems, 29*(07), 2449-60.

Wang, M., Zhao, B., Liu, Y., Wang, F., Wei, L., & Fang, X. (2021). Dynamic task allocation method for the same type agricultural machinery group. *Transactions of the Chinese Society of Agricultural Machinery, 52*, 19–28.

Xiao, H., Lou, P., Man, Z., & Qian, X. (2012). Real-time multi-attribute task scheduling method for automated guided vehicle system. *Computer Integrated Manufacturing Systems, 18*(10), 2224-30.

Xiang, J., Wang, L., Li, L., Lai, K.-H., & Cai, W. (2024). Classification-design-optimization integrated picking robots: a review [Review]. *Journal of Intelligent Manufacturing, 35*(7), 2979-3002.

Yuan, R., Wang, H., Sun, Li-rui, Li, J., & Li, J. (2018). Research on the Task Scheduling of "Goods to Picker"Order Picking System Based on Logistics AGV. *Operations Research and Management Science, 27*(10), 133-138.

Zhang, C., & Noguchi, N. (2017). Development of a multi-robot tractor system for agriculture field work [Article]. *Computers and Electronics in Agriculture, 142*, 79-90.

Zhi-qiang, Z., Xiu-qian, W., Liang, W., Zhong-xiang, Z. H. U., Yue-feng, D. U., & En-rong, M. A. O. (2021). Collaborative Path Planning for Autonomous Agricultural Machinery of Master-Slave Cooperation. *Transactions of the Chinese Society for Agricultural Machinery, 52*(S1), 542-547.

Zhu, S., Wang, B., Pan, S., Ye, Y., Wang, E., & Mao, H. (2024). Task Allocation of Multi-Machine Collaborative Operation for Agricultural Machinery Based on the Improved Fireworks Algorithm [Article]. *Agronomy-Basel, 14*(4), Article 710.