Proceedings of the 39th Chinese Control Conference
July 27-29, 2020, Shenyang, China

Multi-robot Task Allocation and Path Planning System Design

Yunfeng Fan[1], Fang Deng[1,2], Xiang Shi[1]
1. School of Automation, Beijing Institute of Technology, Beijing 100089, P. R. China
E-mail: fan_yun_feng@163.com
2. Beijing Institute of Technology Chongqing Innovation Center, Chongqing, 401120, China

Abstract: This paper studies the task allocation and path planning of multi-robot system in a two-dimensional warehouse logistics environment. A task allocation algorithm based on market auctions is used, and an improved Astar algorithm is proposed to implement multi-robot path planning. During the task assignment, robots make bids in conjunction with the cost of the task itself and the associated cost between tasks, in order to optimize the two performance indicators of the total distance of all robots and the total running time of the robots. Compared with the traditional Astar algorithm, the improved Astar algorithm is combined with traffic rules and reservation tables to obtain the shortest path while avoiding problems such as collisions. The simulation results show that the system can effectively implement the tasks of multi-robot task allocation and path planning in a two-dimensional warehouse logistics environment.

Key Words: Warehousing environment, Multi-robot system, Task allocation, Path planning

1 Introduction

With the rapid increase in the number of online shopping and online payment, China's warehousing industry has developed rapidly in recent years. In recent years, multi-robot systems [1] have played a huge advantage in terms of the types of tasks that can be performed, the efficiency of task execution, and the cost of task execution due to their excellent time and space scheduling flexibility, collaboration, and easy scheduling [2]. Therefore, the multi-robot system can be applied in a warehousing environment to improve the operating efficiency of the system.

There are many research contents of multi-robot systems, such as communication, formation control[3,4], navigation and positioning[5,6], game theory[7]. This paper focuses on task allocation[8,9] and path planning[10,11]. At present, the main methods for solving multi-robot task allocation problems are linear programming, emotional recruitment, and some swarm intelligence algorithms such as ant colony algorithm, fish swarm algorithm[12], particle swarm algorithm [13], etc. The research on motion planning appeared in the 1960s[14], and more and more solutions to path planning problems have been proposed, such as Astar algorithm[15], ant swarm algorithm, neural network, etc[16].

In this paper, auction-based [17,18] method and improved Astar algorithm are used to solve multi-robot task allocation and path planning problems.

2 Multi-robot Task Allocation (MRTA)

2.1 Problem Description

Applying MRTA technology in a warehousing environment, the system can be viewed as two units: drive units, which represent mobile robots, and storage units, which represent shelves for storing goods. In this paper, the task type of the robots is to move the shelf, that is, to move the entire shelf to a specific location, and only perform the operations of entering and leaving the warehouse.

Abstract the warehousing environment into a rasterized map as shown in Fig. 1. The task of the robot is to catch the shelf of the target point (x, y) according to the assignment, and then transport the shelf to the nearby picking station.After the picking is completed, the shelf is transported back to the place (x, y).

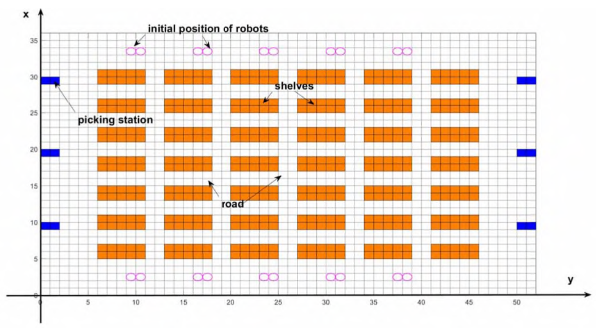

Fig. 1: A rasterized map of warehousing environment

We define the own cost function $S(t_i)$ and associated cost function $R(t_i, t_j)$. Distance is defined by Manhattan distance. Assume that a task is being completed, the coordinate of the picking station to which the robot travels is (x_s, y_s) and the coordinate of the task point is (x_i, y_i) . Since the robot cannot cross obliquely through the grid, its own cost is

$$S(t_i) = 2(|x_i - x_s| + |y_i - y_s|) \quad (1)$$

In the process of completing two tasks (x_i, y_i), (x_j, y_j), define the distance between two task points as the associated cost $R(t_i, t_j)$.

$$R(t_i, t_j) = |x_i - x_j| + |y_i - y_j| \quad (2)$$

Obviously, $R(t_i, t_j) = R(t_j, t_i)$, which means the associated costs of tasks are symmetrical.

This work is supported by the National Key Research and Development Program of China (No.2018YFB1309300).

2.2 Design of Task Allocation with Auction-based Algorithm

Set performance indicator as follow:

$$minf(c(r_1, L_1), c(r_2, L_2), ..., c(r_n, L_n)) \tag{3}$$

where $\{r_1, r_2, ..., r_n\}$ represent the mobile robots, L_i represents the sequence of tasks assigned to robot r_i, function c represents the total cost of each robot to complete the all assigned task.

$$c(r_i, L_i) = \sum_{p=1}^{k} S(t_{ip}) + \sum_{p=1}^{k-1} R(t_{ip}, t_{i(p+1)}) \tag{4}$$

Performance indicators include two aspects: total distance and total time.

$$f(c(r_1, L_1), c(r_2, L_2), ...c(r_n, L_n)) = \sum_{p=1}^{n} c(r_p, L_p) \tag{5}$$

$$f(c(r_1, L_1), c(r_2, L_2), ...c(r_n, L_n)) = \max_i c(r_i, L_i) \tag{6}$$

In this paper, the auction items are tasks, the bidders are robots, and the robot with the lowest bid price is selected as the successful bidder. Because the number of tasks in the system may change constantly, the mode of task allocation can be divided into one-time allocation and reallocation. Suppose that the assigned task sequence of each robot is $\{L_1, L_2, ..., L_n\}$, and there is an unassigned task set T_{wait}. Robots will conduct round-by-round bidding. In each round, every robot will bid for each unassigned task. Then select the smallest bid of all bids, and assign the corresponding task to the corresponding robot.

Suppose that there are 4 robots to assign 5 tasks. The bids of each robot are shown in Table 1 below. Find the minimum bid for each robot, and then find the minimum bid 10 from these bids. And then the task t_3 is assigned to the robot r_2, then it will be deleted from the T which is unassigned set.

Table 1: Robots' bids

Robot\Task	t_1	t_2	t_3	t_4	t_5
r_1	24	16	29	21	33
r_2	35	25	10	40	28
r_3	19	32	23	46	20
r_4	24	30	27	14	22

The above explains the idea and process of bidding. Next, it is necessary to clarify how the price of each robot is determined.

Suppose T is the current set of tasks to be assigned, the bidding robots are $\{r_1, r_2, ..., r_n\}$, and the current task sequences are $\{L_1, L_2, ..., L_n\}$. Robots bid for task t. Let $\{L_1', L_2', ..., L_n'\}$ be the task sequence formed after the assignment of t. Then the difference of performance indicators before and after the allocation is:

$$\begin{aligned} D = & f(c(r_1, L_1'), c(r_2, L_2'), ..., c(r_n, L_n')) - \\ & f(c(r_1, L_1), c(r_2, L_2), ..., c(r_n, L_n)) \end{aligned} \tag{7}$$

In order to get the best performance indicators, the smaller the difference, the better. Supposed that task t is assigned to robot r_i. Considering formula (5), now

$$\begin{aligned} D &= \sum_{p=1}^{n} c(r_p, L_p') - \sum_{m=1}^{n} c(r_p, L_p) \\ &= c(r_i, L_i \cup t) - c(r_i, L_i) \end{aligned} \tag{8}$$

$c(r_i, L_i \cup t) - c(r_i, L_i)$ will be the bid. And

$$BidAll\_route_i^t = c(r_i, L_i \cup t) - c(r_i, L_i) \tag{9}$$

Similarly, when considering formula (6),

$$D = \max_p c(r_p, L_p') - \max_p c(r_p, L_p) \tag{10}$$

For each robot, their bids have a constant term $\max_p c(r_p, L_p)$, so this term can be ignored when biding. D is updated to

$$BidMax\_time_i^t = \max_p c(r_p, L_p') \tag{11}$$

Robot r_i with the task sequence L_i,which is $t_{i1} \rightarrow t_{i2} \rightarrow ... \rightarrow t_{ik}$, bids for task t. We calculate the bid in the form of $t_{i1} \rightarrow t_{i2} \rightarrow ... \rightarrow t_{ik} \rightarrow t$, which means t is placed at the bottom of the sequence, and the bid is

$$\begin{aligned} BidAll\_route_i^t &= c(r_i, L_i \cup t) - c(r_i, L_i) \\ &= S(t) + R(t_{ik}, t) \end{aligned} \tag{12}$$

Since the own cost does not affect the subsequent task auction in the entering and leaving warehouse tasks, so

$$\begin{aligned} BidAll\_route_i^t &= c(r_i, L_i \cup t) - c(r_i, L_i) \\ &= R(t_{ik}, t) \end{aligned} \tag{13}$$

Similarly,

$$\begin{aligned} BidMax\_time_i^t &= \max_p c(r_p, L_p') \\ &= \max(c_{else}, c(r_i, L_i \cup t)) \end{aligned} \tag{14}$$

c_{else} represents the costs of robots except r_i. These robots don¡¯t receive task this time, so c of these robots will not change. And $c(r_i, L_i \cup t)$ will be

$$\begin{aligned} c(r_i, L_i \cup t) = & \sum_{p=1}^{k} S(t_{ip}) + \sum_{p=1}^{k-1} R(t_{ip}, t_{i(p+1)}) + \\ & S(t) + R(t_{ik}, t) \end{aligned} \tag{15}$$

Finally, the bid of robot r_i to task t is

$$\begin{aligned} & BidMax\_time_i^t = \max_p c(r_p, L_p') \\ & = \max(c_{else}, \sum_{p=1}^{k} S(t_{ip}) + \sum_{p=1}^{k-1} R(t_{ip}, t_{i(p+1)}) + \\ & S(t) + R(t_{ik}, t)) \end{aligned} \tag{16}$$

This paper considers the above two performance indicators, so the bid could integrate these two aspects.

$$\begin{aligned} & Bid\Pr ice = \alpha BidAll\_route \\ & +(1-\alpha) BidMax\_time \end{aligned} \tag{17}$$

α is weighting factor, $\alpha \in [0, 1]$.

The flowchart of auction -based algorithm is as Fig. 2.

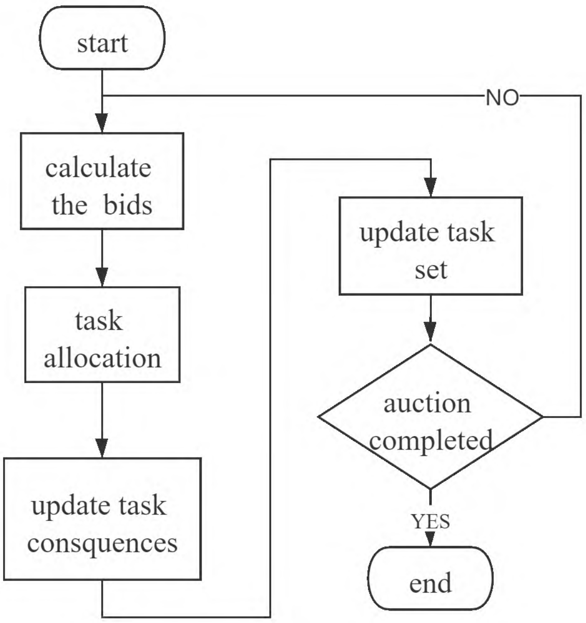

Fig. 2: Flowchart of auction-based algorithm

3 Multi-robot Path Planning (MRPP)

3.1 Problem Description

The purpose of multi-robot path planning is to reduce the running distance of the robot during the task execution, and to avoid collision and blockage of the robots. The path planning of the robot to complete the task can be divided into three stages: from the starting point to the task point, from the task point to the picking station, and from the picking station back to the task point. Because robots may have more than one task, path planning is performed cyclically, and path planning is performed for each stage of each task.

Compared with single robot path planning, multi-robot path planning focuses on collision avoidance between robots. Because the robot is constantly moving, other robots can be viewed as dynamic obstacles for the target robot. For static obstacles, we can effectively avoid them based on the global map information, while mobile robots are dynamic obstacles, and the information on the map is updated at all times.

We made the following simplified assumptions for the warehousing environment in the modeling:

(1) Use a rectangular coordinate system for grid modeling;

(2) The allowed travel directions of the mobile robots are up, down, left, and right;

(3) Each grid point in the map records its current information: 0 means not occupied, 1 means shelf position, 2 means occupied by robot, 3 means picking station;

(4) The mobile robot runs at a constant speed in the warehouse with the speed of 1.

As shown in Fig. 1, it is a storage environment model established by using the grid method. The entire map size is 36x52.

3.2 Improved Astar Algorithm Design

The traditional Astar algorithm is a heuristic search optimization algorithm based on Dijkstra[19]. It is a very effective algorithm for finding the optimal collision-free path from the initial point to the target point in a static graph, but it also exists some problems. The traditional Astar algorithm does not consider the time dimension. When performing path planning, it directly searches for a path from the starting point to the target point. When path planning is required for multiple robots, it may happen that the robots travel to a grid point at the same time and collide. Aiming at the shortcomings of the traditional Astar algorithm, an improved Astar algorithm combined with traffic rules and reservation table mechanism is proposed.

Collisions may occur during the operation of the robots. We simplify collisions into two categories: head-on collisions and cross collisions. Head-on collision is the collision between two robots facing each other, as shown in Fig. 3. Cross collisions can be divided into two types: one is a collision that requires a robot to enter a grid that is already occupied by another, and the other is that when two robots need to enter a same grid at the same time, as shown in Fig. 4, 5.

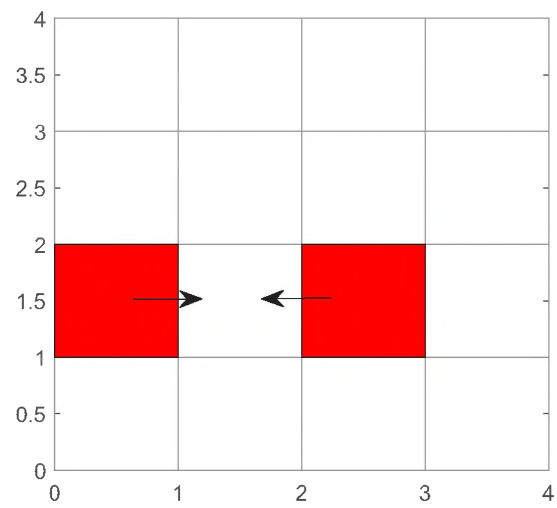

Fig. 3: Head-on collision

In order to solve the problem of head-on collision, this paper adopts the keep-right driving rule that mimics the traffic rules to let the robot drive close to the right of the road to avoid head-on collision, as shown in Fig. 6.

For type 1 cross collision, a waiting rule can be set, that is, the robot which is blocked waiting for the blocking robot to run first. This waiting rule can not only avoid the occurrence of cross collisions, but also avoid "botching" of the robots.

For type 2 cross collision, we propose the reservation table method. The idea of the reservation table is to mark the reservation information for the places where type 2 cross collisions may occur when the robot is planning the path. The robot indicates that it will walk through this grid during the travel, and when other

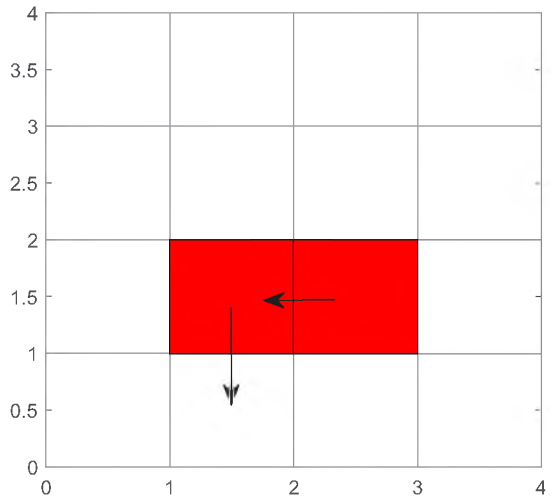

Fig. 4: Cross collision: type 1

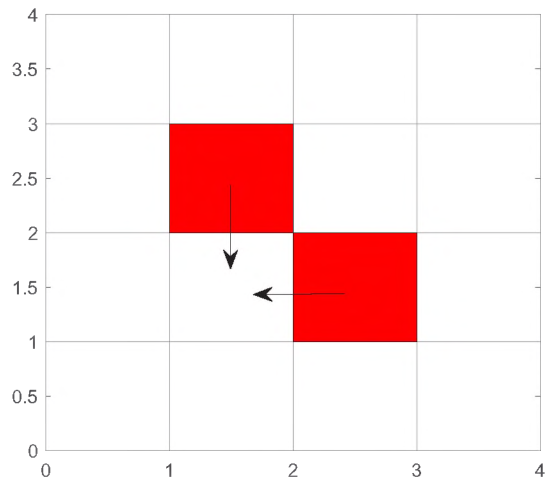

Fig. 5: Cross collision: type 2

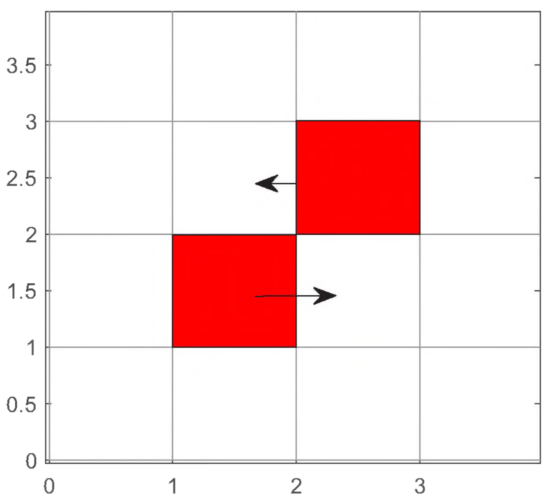

Fig. 6: Keep-right driving rule

robots are going to pass by, they will query the reservation status of this grid. If it is found that another robot has made an appointment and will also enter this grid, then the low priority(according to the ID of robot, small ID with higher priority) robot will wait in place according to the priority order to avoid collisions.

4 Simulation Experiments

4.1 Simulation Environment

This section simulates and implements a multi-robot system. MATLAB is used for simulation, and the GUI interface is designed to display the operation process of the robot in the warehousing environment. The GUI simulation interface is shown below:

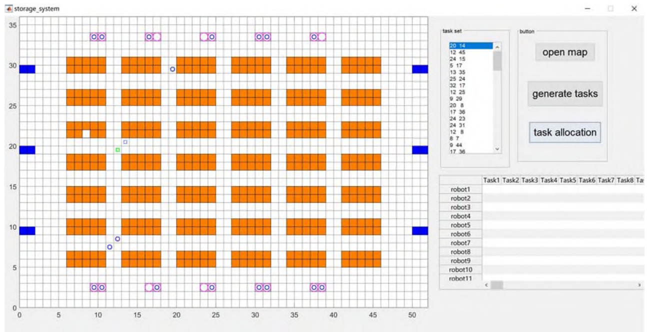

Fig. 7: GUI simulation interface

The size of the map is 36x52, with a total of 20 robots. The tasks are randomly generated with 20 tasks in a batch, and 6 picking stations are distributed on both sides of the warehousing environment.

4.2 Task Allocation Results

This experiment generates 5 batches of tasks, with a total of 100 tasks and 20 robots, which are allocated using three calculation methods of bidding prices. Table 2 records the results of the total distance and running time of the robot at different bid prices.

Table 2: Simulation results of different bid prices

Bid price	Total distance	Total time
$BidAll\_route$	5117	1731
$BidMax\_time$	5348	270
$BidPrice$	5230	294

It can be seen from Table 2 that when $BidAll\_route$ is used as the bid price, the total distance traveled by the robots is the smallest, but the total time for the robots to complete the tasks at this time far exceeds the other two cases. When $BidMax\_time$ and $BidPrice$ are used as the bid price, the total distance and time are similar with each other. $BidPrice$ combines the advantages of $BidAll\_route$ and $BidMax\_time$, and has some optimizations on the two indicators of total distance and total time.

When using $BidPrice$ as the bid price for task allocation, the weight α is an important parameter, indicating the proportion of the two performance indicators of the total distance and time of the robots. Fig. 8, 9 reflect the impact of α on the task allocation results. Obviously, the larger α, the better performance on total distance, but when α reaches a value above 0.9, the total time grows rapidly. In order to balance the performance of total distance and total time, we should choose the α according to the curves to acheive a better allocation.

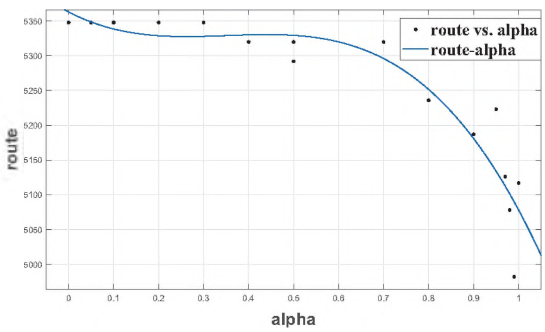

Fig. 8: $\alpha's$ impact on total distance

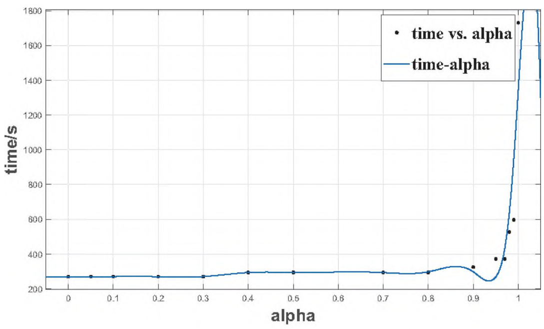

Fig. 9: $\alpha's$ impact on total time

4.3 Path Planning Results

In this experiment, the robots¡¯ path planning method is the improved Astar algorithm. Fig. 10 shows the path planning of one robot to a task point:

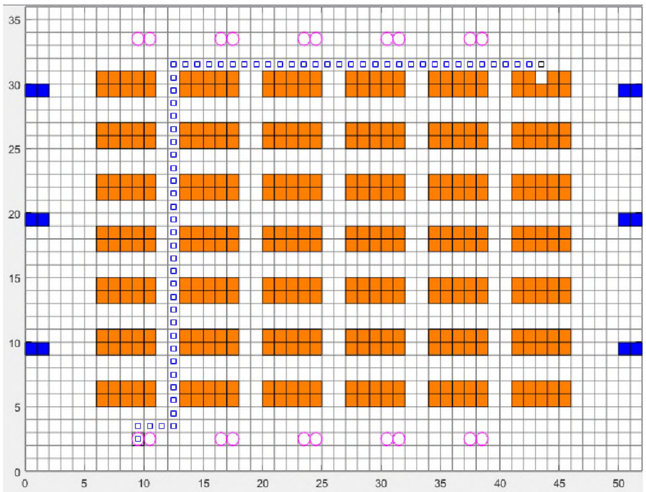

Fig. 10: The path of one robot to task point

The starting position of the robot is (3,10), the coordinate of the task point is (32,44), and the path length is 63. It can be seen that this path is the optimal path under the set rules.

In order to verify the effectiveness of the collision avoidance algorithm, I compared the improved Astar algorithm with the traditional Astar algorithm. With different numbers of tasks, the number of the improved Astar algorithm effectively avoids collisions compared with the traditional Astar algorithm is shown as Fig. 11.

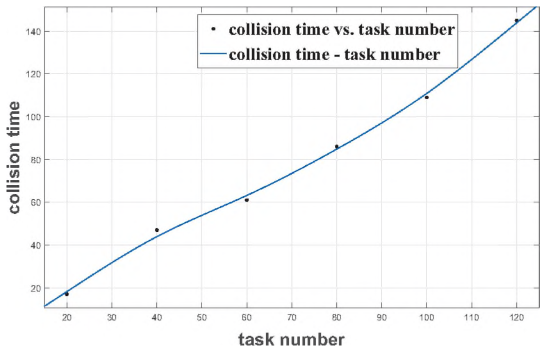

Fig. 11: Number of collisions that can be effectively avoided by improved Astar algorithm compared to traditional Astar algorithm

5 Summary

This paper studies the multi-robot task allocation and path planning in a warehouse environment. The purpose is to respond to the rapid development of the current e-commerce logistics industry, design a multi-robot system applied in a warehouse environment, and use intelligent mobile robots to implement warehouse logistics automation. To improve the efficiency of the entire warehousing system, the methods of task allocation and path planning are designed respectively and simulations verify the effectiveness of the algorithm.

The application of multi-robot systems in the warehousing environment is in a development stage. Many domestic and foreign companies and researchers have invested in this area, but most of them are in the theoretical stage and have few practical applications. This paper has solved some of task allocation and path planning problems of multi robots, but there are still some difficulties. When the density of robots is too high, the congestion that may occur is not resolved in this article.

References

[1] Amasyali, Mehmet Fatih, Marangoz, Salih, Cakmak, Furkan, et al. More scalable solution for multi-robot-multi-target assignment problem. Robotics and Autonomous Systems, 2019, 113: 174-185.

[2] Ronghua Li. Research on task assignment and path planning for multi-robots for intelligent warehouse. Harbin: Harbin Institute of Technology, 2017.

[3] Ya Zhang, Yaoyao Wen, Feifei Li and Yangyang Chen. Distributed Observer-Based Formation Tracking Control of Multi-Agent Systems with Multiple Targets of Unknown Periodic Inputs. Unmanned Systems, 2019, 7(01): 15¨C23.

[4] Jianan Wang, Zhengyang Zhou, Chunyan Wang and Jiayuan Shan. Multiple Quadrotors Formation Flying Control Design and Experimental Verification. Unmanned Systems, 2019, 7(01): 47¨C54.

[5] Lele Zhang, Fang Deng, Jie Chen, Yingcai Bi, Swee King Phang, Xudong Chen, Ben M. Chen, Vision-Based Target Three-Dimensional Geolocation Using Unmanned Aerial Vehicles, IEEE Transactions on Industrial Electronics, 2018, 65(10): 8052-8061.

[6] Fang Deng, Shengpan Guan, Xianghu Yue, Xiaodan Gu, Jie Chen, Jianyao Lv, Jiahong Li, Energy-Based Sound Source Localization with Low Power Consumption in Wireless Sensor Networks,IEEE Transactions on Industrial Electronics, 2017, 64(6): 4894-4902.

[7] Li Liang, Fang Deng, Zhihong Peng, Xinxing Li, Wenzhong Zha, A differential game for cooperative target defense, Automatica, 2019, 102:58-71.

[8] Khamis, A.M., Elmogy, A.M., Karray, F.O. Complex task allocation in mobile surveillance systems. Journal of Intelligent & Robotic Systems, 2011, 64(1): 33-55.

[9] Alaa Khamis, Ahmed Hussein, Ahmed Elmogy. Multi-robot Task Allocation: A Review of the State-of-the-Art. Studies in Computational Intelligence, 2015, 604: 31-51.

[10] Mohammadreza Radmanesh, Manish Kumar, Paul H. Guentert, Mohammad Sarim. Overview of Path-Planning and Obstacle Avoidance Algorithms for UAVs: A Comparative Study. Unmanned Systems, 2018, 06(02): 95-118.

[11] Yiqun Dong, Youmin Zhang, Jianliang Ai. Experimental Test of Unmanned Ground Vehicle Delivering Goods Using RRT Path Planning Algorithm. Unmanned Systems, 2017, 05(01): 45-57.

[12] Shangjun Yang, Yongwei Sun, Yu Pang. Research on Multi-UAV Cooperative Task Allocation Based on Improved Fish Swarm Algorithm. Computer Simulation, 2015, 32(1): 69-72.

[13] YU Lingli, CAI Zixing. Multi-robot mission assignment based on current learning discrete particle swarm optimization algorithm. APPLICATION RESEARCH OF COMPUTERS, 2009, 26(5): 1691-1694.

[14] Nilsson N J. Shakey the robot. Technical Report TR223, SRI International, 1984, 5(3): 123-129.

[15] Michael G.H. Bell. Hyperstar: A Multi-path Astar Algorithm For Risk Averse Vehicle Navigation. Transportation research, Part B. Methodological, 2009, 43B(1): 97-107.

[16] Xiaochuan Zhao, Qingsheng Luo, Baoling Han. Survey on Robot Multi-Sensor Information Fusion Technology. In: 2008 7th World Congress on Intelligent Control and Automation. Chongqing, 2008, 5019 -5023.

[17] Zlot, R.M. An auction-based approach to complex task allocation for multirobot teams. Ph.D. Thesis, Carnegie Mellon University, 2006.

[18] Dias, M., Zlot, R., Kalra, N., etal. Market-based multi-robot coordination a survey andanalysis. In Proceedings of the IEEE, 2006, 1257¨C1270.

[19] ZHANG Yuan, CHEN Yuxuan, WEI Lulu. AGV Intelligent Parking Algorithm Based on Improved A* Algorithm. Computer Systems & Applications, 2019, 28(1): 216-221.