Heat-Map Algorithm Based Multi-robots Path Planning Method

Shuhui Bi[1], Zhihao Li[1], Lei Wang[2], and Yuan Xu[1](✉)

[1] School of Electrical Engineering, University of Jinan, Jinan 250022, China
xy_abric@126.com

[2] HRG (Shandong) Intelligent Equipment Research Institute, No. 1268 Gongye 4 Road, Jinan 250000, Shandong, China

Abstract. Robot conflicts elimination and efficiency improvement from a global perspective are important issues in intelligent storage systems. In order to improve the operation efficiency of intelligent storage system more effectively, a heat-map Algorithm by combing the reservation table in this paper. Firstly, a small storage grid model applicable to multiple storage modes is established. Secondly, considering the frontal collision problem of multiple robots, an improved reservation table is established, which greatly reduces the storage space occupied and improves the operation efficiency at the same time. In addition, the heat map algorithm is added to reasonably allocate the tasks, avoid the congested area and realize the dynamic assignment of tasks. Finally, the effectiveness of the proposed design scheme is demonstrated by simulation.

Keywords: Improved A* algorithm · Improved reservation form · Heat map algorithm

1 Introduction

In recent years, intelligent storage has developed rapidly, and the sales of multi-robot intelligent storage systems have been growing. Conflict-free scheduling of multiple robots is the core of intelligent storage systems. The existing warehouse is expanding, the number of robots is gradually increasing, the system operation process is complex, and the problems are closely connected, and the best route for a single robot and the optimal strategy for a single problem are not optimal when the system is running globally. Since there is more than one robot in the warehouse, conflicts are caused between multiple robots, and the shortest path may have other robots passing through frequently, creating serious conflict problems, and conflicts are strongly dynamic and interleaved, easily causing serial conflicts and congestion, and even deadlock phenomena.

The conflicts between robots can be grouped into the following categories: catch-up conflicts, intersection conflicts, and phase conflicts. For these conflicts, there are various heuristic rules that can be employed, such as improved particle

S.-H. Wang and Y.-D. Zhang (Eds.): ICMTEL 2022, LNICST 446, pp. 655–664, 2022.
https://doi.org/10.1007/978-3-031-18123-8_51

swarm algorithm [1], or adopting various approaches such as leaving, detouring and waiting before starting to avoid different conflicts [1]. The literature [1] proposed a Firefly-based Approach (FA) for robot cluster path planning, where firefly social behavior is used to optimize group behavior. In the literature [1], a Combined Roadmaps and Potentials for Swarms (CRoPS) path planning algorithm is proposed by combining probabilistic roadmaps and potential field methods to enable clusters to move efficiently to desired destinations while avoiding collisions with each other and with static obstacles. The literature [1] investigates an improved bidirectional A* algorithm to reduce the path length and the time required to plan the path for search and rescue UAVs. The literature [1] uses an improved Q-Learning algorithm to plan the shortest path for each robot to complete the mission goal and form a reservation table to reduce the standby state of robots without a task and balance the workload among robots. The literature [1] proposed a particle swarm optimization-based path planning algorithm for UAV clusters in dynamic environments. In route planning, it is easy to fall into the local optimum problem by using a particular algorithm alone, and the literature [1] proposed a hierarchical path planning method based on a hybrid genetic particle swarm optimization algorithm, which can find the optimal path quickly and efficiently by avoiding obstacles in a complex environment.

In the intelligent storage system, the existing methods mostly minimize the conflict range from the path planning, but can not completely optimize the conflict from the global perspective, and can only qualitatively reduce the conflict but not quantitatively eliminate it.

To address this problem, the multi-robots vertex and edge collision problem is considered by using an improved A* algorithm for path finding for each robot based on the establishment of a warehouse raster map with an improved reservation table. Moreover, for the purpose of achieving an efficient multi-robots conflict-free scheduling scheme, a heat map is added by reflecting the congestion level of the lanes and determining the matching rules between robots and order and picking tables.

2 Algorithm Design

2.1 Establishment of the Environment

Raster Map Building. This paper proposes a more standard small storage model applicable to a variety of storage modes, which can cope with the storage requirements of a variety of scenarios. A reasonable environment representation facilitates the establishment of planning methods and the selection of suitable search algorithms to finally achieve a more satisfactory path with less time overhead. There are many ways to build environment maps, such as visual map method, free space method, topology method, raster method, etc. This thesis intends to use the raster method to model the warehouse environment.

A small warehouse model with a length of 25 m and a width of 26 m was constructed, and the map was partitioned using the raster method. Each raster was 1 m long and 1 m wide, and the map was partitioned into 650 rasters, as shown in Fig. 1. The blue area is the picking table location. The picking tables were arranged in the leftmost two columns of the warehouse, and each picking table occupied two grids. The black area is the placement of warehouse shelves. The warehouse shelves are arranged in the form of shelf groups, each group occupies 8 grids, arranged in the form of 2 rows and 4 columns, and the shelf groups are kept at an interval of one grid wide as a transportation path for the robot, the location of the white area in Fig. 1 is the transport aisle. As shown in the figure, the robot represented by the green icon delivers the goods to the designated shelf through the transportation lane. In this paper, we ignore the problem of different sizes of pallets caused by different goods, and set each pallet to be a standard module of 1 m in length and 1 m in width, and the robot transports only one pallet per task.

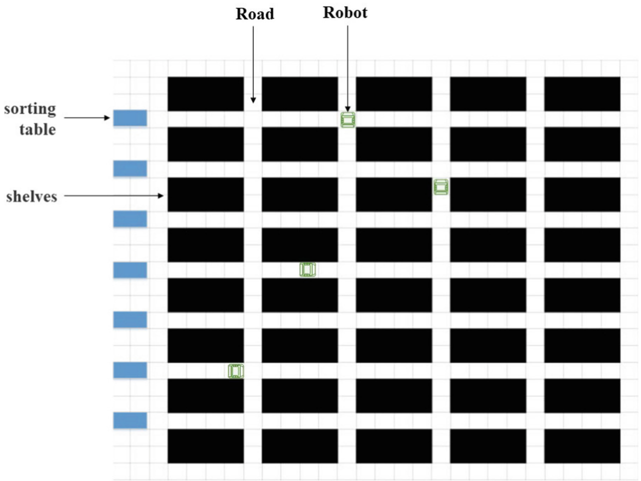

Fig. 1. Raster map and warehouse structure. (Color figure online)

Improved Appointment Form. In intelligent storage systems, multiple robots are involved in transporting goods in and out of the warehouse at the same time, and the environment is dynamic and variable, which can lead to congestion, collision and even deadlock. To address this problem, this thesis uses a reservation table to monitor and record the motion paths of robots, which facilitates the subsequent path optimization algorithm to query the historical paths of robots by calling the reservation table to avoid path conflicts.

The reservation table created is shown in Fig. 2. Variable k indicates the number of reservation tables corresponding to the current moment, the reservation table is arranged in time order, and the time interval is the time consumed by the robot to move from the center position of the grid to the center position of the adjacent grid Δt. The system records the appointment table every Δt. The kth reservation table records the position states of all robots at the moment of

$k \cdot \Delta t$, and the (k-c)th reservation table records the information at the moment of $(k - c) \cdot \Delta t$. The $r_i x$ in the kth reservation table denotes the horizontal coordinate of the warehouse location where robot i is at the moment of $k * \cdot \Delta t$, and $r_i y$ denotes the vertical coordinate of the location where robot i is at this moment. When the robot updates its path, the position information in the reservation table will be updated simultaneously.

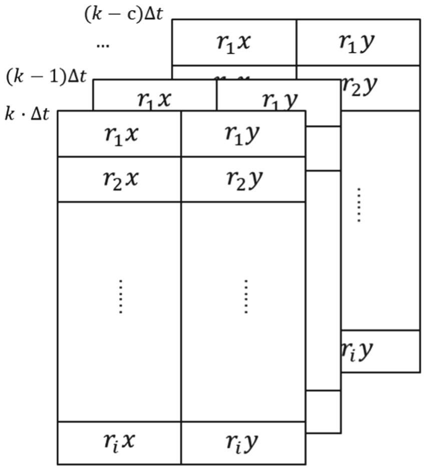

Fig. 2. Improved reservation form.

The number of rows and columns of the matrix of the unimproved reservation table is equal to the number of rows and columns of the raster map, and the route is circumvented by querying the "0" and "1" status of a location. This method occupies a lot of useless matrix space, causing matrix redundancy, and the query step is tedious. In the improved reservation table, the number of rows in each reservation table is equal to the number of robots, the number of columns is 2, and only the X and Y coordinates of the grid location of each robot at the current moment are stored. When the algorithm calls the improved reservation table, it only needs to compare whether the same coordinates appear at the same moment to determine whether there is a conflict and achieve the purpose of collision avoidance.

2.2 Improved A* Algorithm

A* algorithm, as a heuristic algorithm, prioritizes the extended nodes according to the estimated cost function, selects the best node, and repeats the above steps at this node until it reaches the target point, which is the most effective direct search method for solving the shortest path in static maps. The warehouse map built in this thesis is simple and regular, which is suitable for calculating the shortest path using A* algorithm.

The robot starts from the starting raster point and expands the surrounding raster at the current raster point. In this paper, the robot in the warehouse only moves up, down, left and right, so the four-neighborhood search method is chosen. The current position is called the parent node, and the estimated cost of the surrounding four directional grids is calculated and put into the candidate table for storing the extended nodes. After all the surrounding nodes finish estimating the cost, the grids with the smallest estimated cost are selected as the new moving position, and this position becomes the new parent node. The new parent node is used as the center to expand the grid, and this step is repeated until the robot reaches the target point. Finally, an optimal path is obtained from the starting point to the target point with the minimum cost.

The expression of the improved cost estimation function of the A* algorithm is:

$$f(n) = g(n) + h(n) + \sum_{j=1}^{m} t_{j(turn)} + \sum_{j=1}^{m} t_{j(wait)} \tag{1}$$

where $\sum_{j=1}^{m} t_{j(turn)}$ is the sum of the extra time spent by the robot in steering from the starting grid to the current grid, and $\sum_{j=1}^{m} t_{j(wait)}$ is the extra time spent by the robot in waiting in place due to path conflicts in the process from the starting grid to the current grid. $g(n)$ denotes the actual cost of moving from the starting grid to the current grid n. The actual cost is generally expressed by distance or time, and this paper uses time as a uniform scalar to compare the size of the cost of the function $f(n)$. $g(n)$ is expressed as:

$$g(n) = \frac{d}{v} \tag{2}$$

d is the actual distance traveled by the robot from the starting grid to the current grid n, and v is the speed at which the robot travels at a uniform speed. $h(n)$ denotes the heuristic estimation cost from the current grid n to the target grid point, and the expression is:

$$h(n) = \frac{d_n}{v} \tag{3}$$

d_n is the estimated shortest distance of the robot from the current raster n to the target raster point, where the estimated distance is calculated using the Harmattan distance:

$$d_n = abs(n.x - goal.x) + abs(n.y - goal.y) \tag{4}$$

The Harmattan distance is the sum of the horizontal and vertical distances of the current node n and the target point.

However, when the A* algorithm expands nodes in the parent node, there will be a situation that the expanded nodes are already occupied by other robots, which will cause vertex conflict among robots if not avoided. Adding a query step to the reservation table in the A* algorithm can achieve the purpose of avoiding the vertex conflict.

Meanwhile, since the expansion interval of the A* algorithm is a fixed value, it generally takes a raster as a unit and expands according to the raster. This will lead to the situation of edge conflict, i.e., the reservation table is not queried to have robots in the same position at the same time, but the run will collide at the raster junction, as shown in Fig. 3.

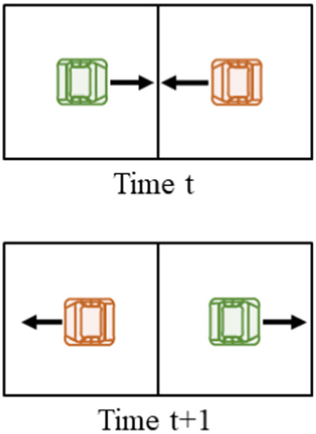

Fig. 3. Situations of edge conflict.

In order to avoid edge conflict, the reservation table is cross-called in the A* algorithm and brought into the decision formula. If the decision formula is satisfied, it is determined that this extended node will have edge conflict and the node needs to be reselected. The determination formula is as follows:

$$F_{point}(x, y) \in R_{(k+1)} \bigcap N_{point}(x, y) \in R_{(k)} \tag{5}$$

where $F_{point}(x, y)$ denotes the coordinates of the parent node, $N_{point}(x, y)$ denotes the coordinates of the extended node, R_k denotes the reservation table matrix at the moment of the parent node, and $R_{(k+1)}$ denotes the reservation table matrix at the moment of the extended node.

2.3 Heat-Map Algorithm

By improving the A* algorithm and the control of the reservation table, the collision problem is avoided, but only local conflicts can be solved, and the operation of the whole raster map path cannot be observed globally. As the number of robots increases, there may be local abnormal congestion while other locations in the warehouse are free, which greatly reduces the efficiency of the robot system in delivering goods.

For this reason, this thesis incorporates a heat map algorithm that reacts to the congestion level of the aisles, and takes the shelf groups and picking tables as units, and reacts the congestion level of the surrounding aisles to the corresponding shelf groups and picking tables as the basis for selecting the target points and picking tables for new tasks, as shown in Fig. 4. By calling the reservation table data for a certain time period, the number of robots passing through each aisle during this time period is calculated and used as the congestion level value for each aisle, which is involved in the calculation of the heat value for each shelf group.

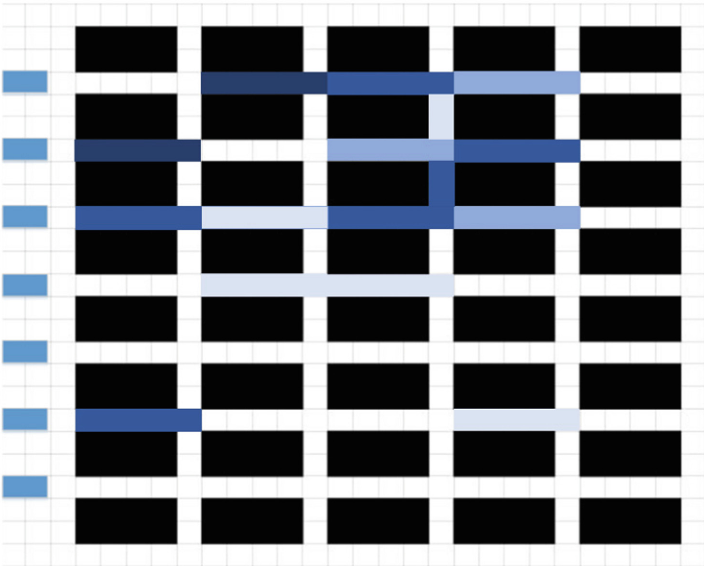

Fig. 4. Diagram of the heat map reflecting the level of congestion.

The congestion level of each lane is calculated by the following formula:

$$Congestion = \frac{N(k - RI \cdot \Delta t, k)}{RI} \tag{6}$$

where $Congestion$ is the congestion level of the lane at the recording range of RI, and $N(k - RI \cdot \Delta t, k)$ is the number of robots passing through this lane from the moment of $k - RI \cdot \Delta t$ to the current moment of k.

To facilitate the calculation of the thermal value, the coordinates of the shelf groups and the lanes are numbered, both using the top-down and left-to-right numbering, with the coordinates of the first shelf at the top left being (1, 1), and so on, to obtain the following formula for calculating the thermal value of each shelf group.

$$Wh(i, j) = C_w(i, j) + C_w(i, j + 1) + C_l(i, j) + C_l(i + 1, j) \tag{7}$$

where $Wh(i,j)$ is the heat value of the shelf group with coordinate (i,j), $C_w(i,j)$ is the congestion degree value of the short aisle to the left of this shelf group, and $C_l(i,j)$ is the congestion degree value of the long aisle in front of this shelf group.

The unassigned tasks correspond to the shelf group where their target points are located one by one, and are sorted in ascending order based on the heat value of the shelf group where they are located; the smaller the heat value is, the more forward the task point is, and if the heat values are the same, the task point with the closest distance to the matching transport robot takes precedence. The reordered task list avoids orders for goods in shelves within the congestion range, as well as choosing to avoid picking stations with long waiting times, to achieve dynamic allocation.

3 The Simulation Results

For demonstrating the effectiveness of the proposed algorithm, the reservation table-based A* algorithm was compared with the improved A* algorithm based on the heat map algorithm (hereafter referred to as the optimized algorithm) designed in this paper in the warehouse raster map model created in this thesis. Simulation experiments are conducted using MATLAB, and the simulation experiments are as follows.

(1) Compare the situation of transporting different number of task orders with the same number of robots, the number of robots is set to 8, and the number of orders is 50, 100, 150, 200, 250 in order. Simulation results are shown in Fig. 5, compared with the A* algorithm, the efficiency of order completion under the optimization algorithm are greatly improved, the more the number of orders, the more obvious the efficiency improvement of the optimization algorithm, from 50 groups of tasks 7.91% improvement, gradually increasing to 19.36% improvement for 250 groups of tasks.

(2) Comparing different numbers of robots delivering goods under the same order, the number of goods is set to 100, and the number of robots is 2, 5, 8 and 12. simulation results are shown in Fig. 6, compared with the A* algorithm, the order completion time under the optimization algorithm are shortened, and the effect of the optimization algorithm gradually decreases with the increasing of robots number, from 38.28% improvement of 2 robots to 15.06% improvement of 12 robots, but it still has a significant lifting effect.

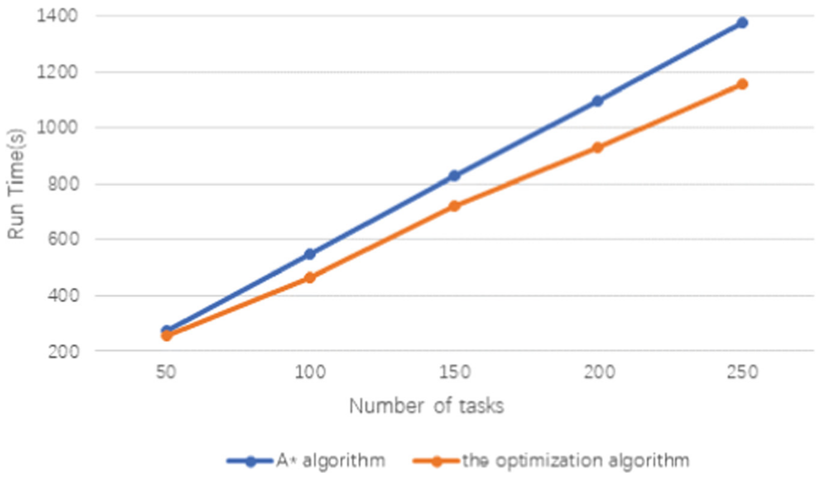

Fig. 5. Simulation results with different number of orders.

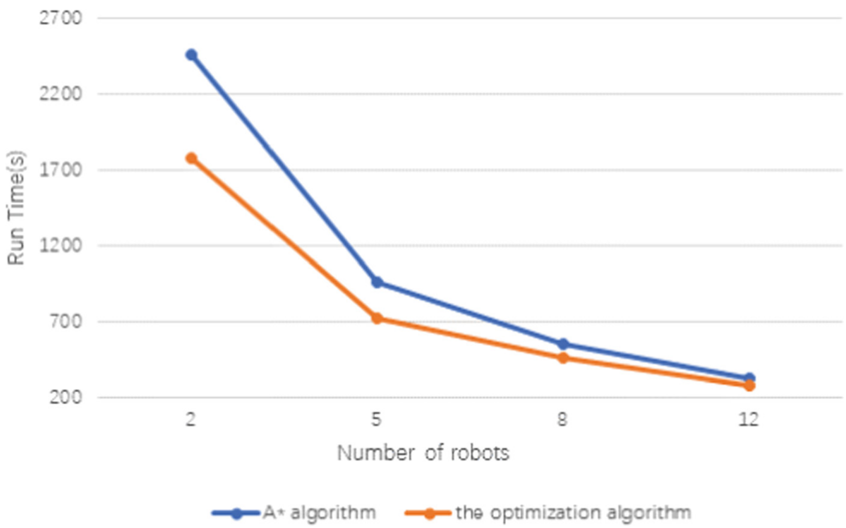

Fig. 6. Simulation results with different number of robots.

4 Conclusion

In this paper, a heat map based improved A* algorithm is proposed for the multi-vehicle scheduling problem in the intelligent storage system. The following work is accomplished: the warehouse raster map is established; the reservation table is improved, which greatly reduces the storage space occupied by the reservation table while improving the operation efficiency. The A* algorithm was improved to avoid vertex conflicts and edge conflicts at the same time. The heat map algorithm is added to reasonably allocate tasks, avoiding congested areas and realizing the dynamic assignment of tasks. Finally, the feasibility and effectiveness of the scheme are verified by simulation.

Acknowledgements. This paper is supported by the Natural Science Foundation of Shandong Province (ZR2020KF027), the National Key R & D Program of China (2018AAA0101703).

References

1. Zhang, Y., Jun, Y., Wei, G., Lenan, W.: Find multi-objective paths in stochastic networks via chaotic immune PSO. Expert Syst. Appl. **37**(03), 1911–1919 (2010)
2. Zhang, Y., Lenan, W., Wei, G., Wang, S.: A novel algorithm for all pairs shortest path problem based on matrix multiplication and pulse coupled neural network. Digit. Sig. Process. **21**(04), 517–521 (2011)
3. Hidalgo-paniagua, A., Vega-rodrguez, M.A., Ferruz, J.: Solving the multi-objective path planning problem in mobile robotics with a firefly-based approach. Soft. Comput. **21**(04), 949–964 (2017)
4. Uzol, O., Yavrucuk, I., Sezer-Uzol, N.: Panel-method based path planning and collaborative target tracking for swarming micro air vehicles. J. Aircr. **47**(02), 544–550 (2010)
5. Li, S., Weihua, S., Guo, P., Zhang, S., Xie, P.: Research on SAR drone global path planning based on improved A* algorithm. Chin. Med. Equip. J. **41**(12), 16–20 (2020)
6. Chen, M., Qian, T., Zhang, S., Wang, J.: Obstacle avoidance and cooperative path planning method of warehouse logistics robot cluster. Mod. Electron. Tech. **42**(22), 174–177+182 (2019)
7. Zhang, Y., Lenan, W., Wang, S., Balochian, S.: UCAV path planning by fitness-scaling adaptive chaotic particle swarm optimization. Math. Probl. Eng. **2013**(03), 1–9 (2013)
8. Ouyang, H., Quan, Y., Gao, L., Zou, D.: Hierarchical path planning method for mobile robots based on hybrid genetic particle swarm optimization algorithm. J. Zhengzhou Univ. (Eng. Sci.) **41**(04), 35–40 (2020)