Multi-agent AGV Conflict Free Path Planning Based on Improved Speed Control Method in Automated Terminals

Kunlun Guo[a], Jin Zhu

Institute of Logistics Science & Engineering, Shanghai Maritime University, Shanghai 201306, China.

[a]1309154711@qq.com

Abstract

Aiming at the problem that the increasing number of automated guided vehicles (AGVs) will lead to more frequent conflicts between AGVs.In this paper, a conflict-free path planning model for multi-AGV is established, aiming to minimize the blocking rate of AGVs between the quay crane and the yard crane, considering the travel speed, operation time and conflict distance of AGVs. Designing an architecture of AGV's system based on Multi-Agent System (MAS), the interactive protocol based on blackboard model is used as the communication method of AGV, the improved speed control method and the method of determining AGV priority based on time cost are used as the AGV negotiation strategy, improving Dijkstra algorithm calculates the conflict-free path of each AGV. The results show that the average blocking rate of MAS control mode is 0.28% lower than average blocking rate of task priority control mode. In different scale maps, when the number of AGVs is more than 30, the increase of average blocking rate based on MAS is lower. The study shows that the control method based on MAS is more suitable for the conflict free path planning of more than 30 AGVs.

Keywords

Automated terminals, Multi-AGV, Multi-agent system (MAS), Improved speed control method, Conflict-free.

1. Introduction

Automated Guided Vehicle (AGV) is the main means of transporting containers in an automated terminal and one of the important equipment for terminals automation. Under the influence of increasing difficulty of manual operations, large-scale operations, and intelligent terminals, the number of AGVs has been increasing. At the same time, the increase in number has also led to frequent occurrences of equipment waiting, conflicts, and deadlocks during their operations. Automated terminals are more concerned and urgently need to be resolved.

For the problem of multi-AGV conflict-free path planning, many scholars have made research, for example, Wang Yu et al. [1] used improved ant colony algorithm to solve AGV conflict-free path planning problem, Fang Hua et al. [2] used A ~ * algorithm Solve three-dimensional AGV conflict-free path planning model. Zhang Su-yun et al [3] established a multi-parameter optimization control model, which considered the number of AGVs in the path, the safe distance and speed of AGV, and improved the speed control strategy to resolve conflicts. Zhong Mei-su et al. [4] considered the AGV driving speed, AGV reload rate and conflict time, and established the multi-AGV conflict-free path planning model with the goal of minimizing the AGV driving distance between the quay crane and the yard crane, used speed control strategies to resolve conflicts. Cao Xiao-hua et al. [5] proposed a multi-AGV collision avoidance decision optimization method based on conflict prediction, and applied an improved particle swarm optimization algorithm to the optimization of collision avoidance strategies to resolve conflicts. Liu et al. [6] modeled the goal of minimizing the completion time of the AGV task, considered the AGV travel speed to locate and avoid the constraints of conflict, and verified the model and algorithm by simulation to avoid AGV conflict and congestion. There are a large number of literatures about the research scale of multi-AGV conflict-free path is between 20 and 30. However, at present, the number of AGVs in the main domestic automated terminals is 18, 38 and 50, and the design scale of Yang shan phase IV is as much as 130. Therefore, it is more practical to solve the conflict free path planning of multi-AGV, especially more than 30.

At present, a large number of documents mainly apply multi-agent system (MAS) to the workshop and logistics warehouse AGV. Li Xiao-meng et al. [7] established an AGV scheduling system based on multi-level decision-making and cooperative learning method, and designed a distributed dynamic scheduling strategy for AGV. Koen H et al. [8] introduced the concept of agents and proposed a way to improve the accident handling of multi-AGV systems through the application of cooperative control. Compared with the existing multi AGV system in the automation terminal, it is found that the MAS based control mode can improve the operation efficiency of the system. Fauadi et al. [9] proposed a distributed architecture based on MAS to control the AGV operating system and agent architecture in the manufacturing industry, aiming to achieve control Material handling activities. Erol Rizvan et al. [10] proposed the collaborative scheduling of AGV and manufacturing systems in a real-time environment based on MAS, and used the bidding and winning mechanism in MAS as the negotiation strategy for multiple AGV systems. Jing Jian-feng [11] established an agent-based distributed multi-AGV control system. After the AGV tasks are assigned, path planning can be autonomously performed, and at the same time, AGVs can communicate and collaborate with each other when conflicts occur to task. However, due to the more complex operation environment and larger map scale of AGV in automated terminals, there are few literatures that apply MAS to the terminal AGV to solve the conflict-free path planning of multi AGV. Moreover, most of the literature considers the use of speed control and parking waiting to resolve conflicts when the AGV reaches the conflict node, which affects the system's operating efficiency, and rarely considers using speed control to reduce the parking waiting time before reaching the conflict node.

In summary, to minimize the blocking rate of the AGV between the quay crane and the yard crane, considering the AGV's driving speed, operating time and conflict distance, an automated terminals multi-AGV conflict-free path planning model is established for the problem of conflict-free path planning for more than 30 AGVs. At the same time, this paper designs a multi-AGV architecture based on MAS, improves Dijkstra algorithm to plan AGV's paths, improves speed control method considering conflict distance of AGV to control the speed of AGV before reaching the conflict node to reduce negotiation time and resolve conflict. Two comparative experiments are designed based on MAS control mode and task priority control mode to compare the average blocking rate, average waiting time, average completion time of AGV under different numbers and the average blocking rate of multi AGV under different scale maps.

2. Model establishment

2.1 Model assumptions

(1)The positions of the quay crane and yard are fixed and known, and one quay crane corresponds to all yards.

(2)Setting up a buffer zone at the quay crane and yard, so that the AGV don't form a conflict with other running AGVs in the path while waiting in line in the quay crane operation area and the yard operation area.

(3)Does not consider the influence of force majeure factors such as faults and weather during AGV driving.

(4)AGV speed remains unchanged during turning.

2.2 Variable setting

When representing the operation path network of multiple AGVs in an automated terminal, use the $G=(N,W)$ directed weighted graph to represent the path network of the AGV. Where N is the set of all node numbers in the AGV running map, W is the set of G edges, $W_{k(i,j)}$ represents the length of the edge from the i-th node to the j-th node on the path k. The following variables are introduced below for convenience.

L is the AGV device's own length.

R is the radius of the detection range of the conflict distance sensor during AGV operation.

L_s is the safety distance between AGVs during driving.

v is the average speed of the AGV operation.

α is the acceleration / deceleration of AGV when a conflict is detected.

D_k is the length of the k-th path, $k=1,2,...,\mathrm{K}$, and K is the set of all AGVs' path.

AGV_{km} is the number of the m-th AGV in path k.

$C(k_1,k_2)$ are the paths k_1 and k_2 of the two conflicting AGVs, and $k_1,k_2\in K$.

w is the number of collisions of the m-th AGV in path k.

s is the starting point in path k.

e is the ending point in path k.

A_{km} is the priority of the m-th AGV in path k.

t_{kms} is the start time of the m-th AGV in path k.

t_{kmep} is the running end time of the m-th AGV in path k.

t_{kmeop} is the estimated task completion time of the m-th AGV in path k.

t_{ck} is the waiting time of the AGV at the collision node in path k.

t is the delay time of AGV in the path due to conflict.

T is the total time consumed by the AGV from the starting point to the ending point.

2.3 Distributed control model

The construction model for the AGV path conflict problem is as follows:

During the operation of the AGV, Eq (1) and Eq (2) indicate that each node is visited by the AGV at most once at the same time, that is, the AGV does not repeatedly drive the same road segment in the path, and only one AGV can pass through the same node; Eq (3) Represents the length of any AGV traveling in any path; Eq (4) represents the end running time of any AGV in any path to complete the task.

$$X_{\mathrm{ij}}=\begin{cases}1,\text{AGV visits node i first and then node j}\\0,\text{otherwise}\end{cases}\tag{1}$$

$$\sum_{i=1}^{N}\sum_{j=1}^{N}X_{ij}=1\tag{2}$$

$$D_k=\sum_{i=1}^{N}\sum_{j=1}^{N}W_{k(i,j)}X_{ij}\tag{3}$$

$$t_{kmep}=\frac{D_k}{v}\tag{4}$$

After the task is assigned to the AGV, the driving speed does not change to v_0 during the operation of the AGV, and in order to avoid conflicts between multiple AGVs on the same path from the same starting point, the AGV needs to maintain a minimum safety distance L_s, that is, safety detection Distance 2R.

If two or more AGVs in different paths reach the cross node of the two paths at the same time, the AGV may conflict at the cross node. If a conflict occurs, the AGVs need to be negotiated to determine which AGV passes the conflict node to solve the conflict problem. When a conflict occurs, the AGVs firstly extract the AGV number for negotiation, and compare the priority of the AGV first. Eq (5) is to calculate the estimated task completion time according to the path planning of a single AGV; Eq (6) is to determine the priority of the AGVs according to the estimated task completion time. The AGV with a long task completion time has a low priority; otherwise, the priority is high.

$$t_{kmeop} = t_{kmep} + t_{kms} \tag{5}$$

$$t_{k_1 meop} > t_{k_2 meop} \rightarrow A_{k_1 m} < A_{k_2 m} \tag{6}$$

After negotiation, the AGVs make adjustments according to their own priorities. AGVs with low priority may start to slow down before reaching the conflicting node, and may stop until the AGVs with high priority pass through the conflicting node at a uniform rate. Eq (7) is that when the AGV in k_1 and the AGV in k_2 detect each other through the distance sensor, the safety distance of the two AGVs is less than the distance of the conflict detection range; Eq (8) is that the AGV in k_1 and the AGV in k_2 are detected by the distance sensor , when reaching the other party, negotiate and agree that the AGV in k_2 first passes the conflicting node, and the AGV in k_1 decelerates from v_0 to v_1 by the distance l_s; Eq (9) is the time t_{c11} for the AGV in k_1 to decelerate from v_0 to v_1, which is also the AGV The time t_{c12} for accelerating the recovery to the original speed; Eq (10) is the final deceleration v_1 in the conflict resolution process of the AGV in k_1; In Eq (11), the time of arrival of the m-th AGV after w collision in k_1 is $T_{k_1 m}$; Eqs (12) ~ (13) are the waiting time of n AGVs in the path due to conflict delay, and the total time for n AGVs to complete the task; Eq (14) is to minimize the blocking rate during the operation of n AGVs, the ratio TJ of total waiting time and task completion time.

$$L_s < 2R \tag{7}$$

$$l_s = \frac{v_0^2 - v_1^2}{2\alpha} \tag{8}$$

$$t_{c11} = \frac{4R^2 - (\sqrt{2}R - l_s)^2}{2\alpha} \tag{9}$$

$$v_1 = v_0 - \alpha t_{c11} \tag{10}$$

$$T_{k_1 m} = t_{k_1 mep} + w(t_{c11} + t_{c12}) \tag{11}$$

$$\sum_{k=1}^{n} t = t_{c1} + t_{c2} + \ldots + t_{cn} \tag{12}$$

$$\sum_{k=1}^{n} T \tag{13}$$

$$\min TJ = \frac{\sum_{k=1}^{n} t}{\sum_{k=1}^{n} T} \tag{14}$$

3. The architecture of MAS application in AGV of automated terminals

3.1 AGV's running map model

When MAS is applied on the AGV of the automated terminals, the AGV as an agent can determine its own position through the sensor, and then compare it with the known coordinate value road signs, so as to match with the global coordinate system to obtain its own real-time position. Therefore, in this paper, when studying the running path of multi-AGV based on MAS, the topology map is selected, and the topology method uses quay crane, yards, and the intersections of paths in the terminals as nodes in the topology map, between nodes the connection indicates the AGV operation route in the actual terminals.

A single-lane one-way path network means that the lanes on a path have only one direction and one lane. There is no two-way dual-lane, and there can only be one AGV in the vertical direction of each path. At the same time, the AGV operation line in the actual terminals is positive, so the established topological map is a weighted directed graph, and the weights of the edges are all positive. The adjacency list method is used to build a directed weighted graph. The adjacency list method is to store all other vertices connected to a certain vertex into a linked list, and associate the linked list to the vertex. In addition, the AGV running path in the terminals is mostly right-angled, that is, there may be edges connected to other nodes in the top, bottom, left, and right directions of the path node in the topology map.

The design code for implementing the topology map for the adjacency list is as follows:

```
a) struct ArcNode_t
    {
        Vertex_t _VertexIndex;
        Weight_t _Weight;
        ArcNode_t* _ArcNext;
    };
b) struct VertexNode_t
    {
        Vertex_t _VertexIndex;
        uint32_t _InDegree;
        uint32_t _OutDegree;
        ArcNode_t* _FirstArc;
        ArcNode_t* _TailArc;
    };
```

3.2 Interactive protocol based on blackboard model

In the multi-AGV architecture of MAS-based automated terminals, the interaction protocol is the communication between AGV and AGV, and AGV and console. The AGV needs to send and receive information during operation, that is, the number of the trolley, the task of the trolley, the current location, the current time, and the location of the next node that the AGV needs to send to the console. The console needs to send a path table to all AGVs to occupy a node, so that AGVs can negotiate to solve the problem before reaching the conflict node. The interaction information between the AGV and the AGV is mainly when the AGV conflicts, and the negotiation needs to send their current position, current time and their own priority level.

According to the interactive information required in this article, synchronization and cognition of shared resources is an important prerequisite for collaboration between AGVs. The blackboard model in MAS can manage global resources to a greater extent, so this paper uses the blackboard model to design multiple AGVs interactive protocols, as shown in Fig. 1.

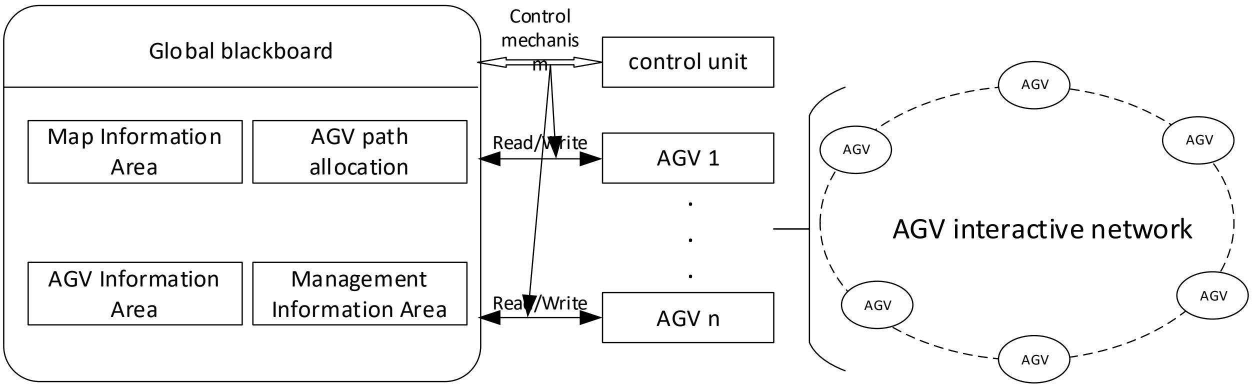


Fig. 1 Multi AGV interaction protocol based on blackboard model

(1)The main function of the blackboard is to assign the path from quay crane to yard and yard to quay crane in the map to each AGV, monitor the position, speed, and time of each AGV, and save it in the form of a vector, that is, vector <int> :: iterator it, it is a position, speed, and time element that can be read and written. At the same time, the blackboard also needs to send the position, speed, and time of each AGV to each AGV in the form of a vector.

(2)The AGV running in the system is the source of knowledge in the blackboard model. (a)Each AGV has its own operating characteristics, such as possible conflicts and its own operating parameters (speed, time, priority, etc.), which are all information required by the blackboard control mechanism. (b)When AGVs are about to conflict, the blackboard control mechanism allows the AGV to negotiate and extract information such as priority from the blackboard for negotiation.

(3)The function of the blackboard control unit in the system is completed by the wireless network, and the interval between AGV and AGV interactive information and AGV and blackboard interactive information in this article is 0.02s.

3.3 AGV conflict negotiation strategy

The negotiation strategy of AGV is mainly to judge based on the priority of AGV, and then make concessions to resolve conflicts. This paper uses the method based on time cost to determine the priority of AGV, and combines the speed control method as the negotiation strategy of AGV.

3.3.1AGV priority determination based on time cost

This article uses a time-cost-based method to determine the priority of the AGV. Fig. 2 shows, after assigning tasks to each AGV, AGV uses the shortest path algorithm to get the shortest path according to the starting point and end point, and calculates the expected completion time of the task, and determines the expected priority of AGV according to the time. When resolving the conflict, the low-priority AGV decelerates until the high-priority AGV passes the conflicting node. At this time, the low-priority AGV takes more time and recalculates the time to determine the priority. The level of the level will also change accordingly.

3.3.2Improved speed control method

In the multi AGV system of automated terminals based on MAS, most studies focus on the conflict of intersection when one AGV reaches the conflict node, and the other AGV uses the method of speed control and parking to solve the conflict. If the two AGVs are close, they have to stop and wait, which seriously affects the efficiency. According to the collision distance detection of AGV, this paper improves the speed control method, which makes AGV decelerate before reaching the collision node, avoids stopping and waiting, and improves the operation efficiency.

The AGV conflict is shown in Fig. 3. When two AGVs are about to reach the same intersection, the collision detection sensors of AGV1 and AGV2 are set to the detection radius of R1 and R2, and R1 = R2. Factors such as length and path length are set, and the detection critical circle moves as the AGV moves. In (a) of Figure 3, AGV1 and AGV2 are driving towards the intersection of the route at the same time. At this time, the detection critical circles of AGV1 and AGV2 do not intersect, so the two AGVs have not detected the conflict; when the two AGVs continue to run to the map in the case of (b) of 3, the two detection critical circles begin to intersect, indicating that AGV1 and AGV2 may have a conflict at the intersection node. At this time, AGV needs to negotiate a concession to resolve the conflict.

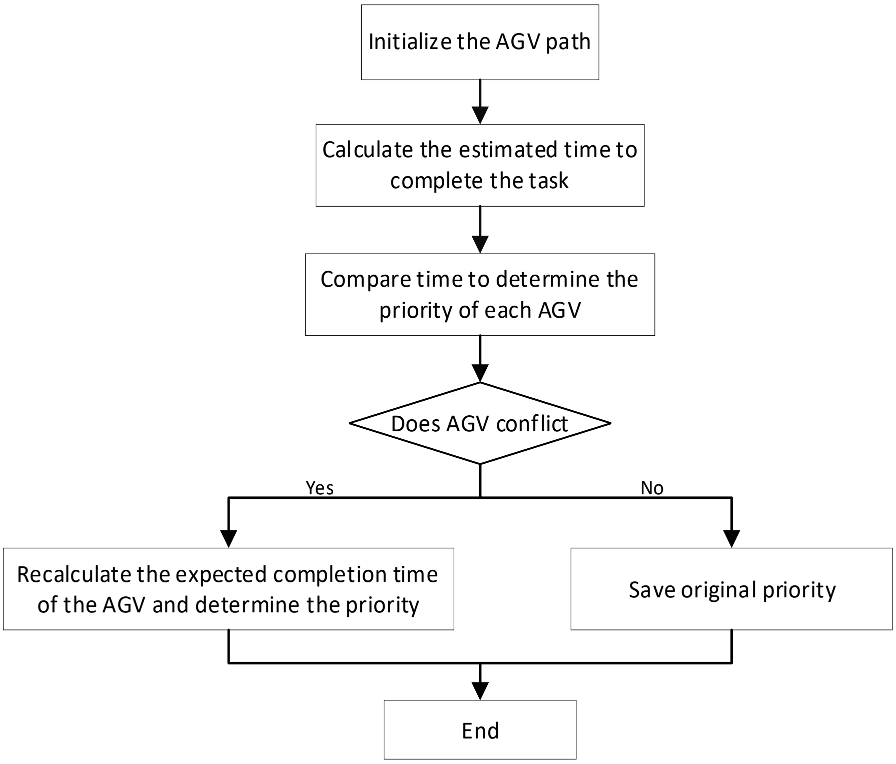
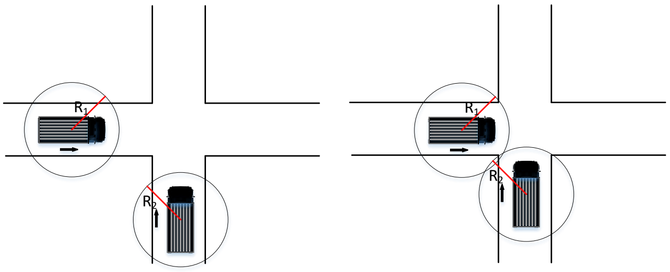


Fig. 2 Priority flowchart of AGV based on time cost



(a) (b)

Fig. 3 AGV conflict diagram

Let's take one of the two AGVs in conflict as an example to explain the conflict resolution process:
(1)When the detection critical circle of the AGV begins to intersect with other detection critical circles, the AGV confirms that a conflict is imminent based on the information provided by the blackboard model.

(2)The AGV extracts the AGV number of the other party, determines the negotiation object AGV and communicates with it, and requests to pass the front possible conflict node.

(3)The two parties obtain the result according to their own priorities, determine one of the AGVs to lock the collision node, and pass at a uniform.

(4)The low priority AGV slows down, waits until the high priority AGV passes the node, and then restores the original speed to pass the node to solve the conflict problem.

3.4 Improved Dijkstra algorithm

The Dijkstra algorithm can effectively solve the shortest path on the topological map with weighted directed connections. After determining the starting point and the ending point, the starting point is taken as the center, and the idea of the greedy algorithm is adopted. The node that is closest to the starting point and never visited until it reaches the target point appears in the search range. In AGV path planning, the weight value represents the length of the edge. If the two points are not connected, then the value corresponding to their weight value is infinity. However, the Dijkstra algorithm traverses and calculates each node, which is inefficient in calculation time and wastes calculation space. Therefore, this paper uses heap optimization to improve Dijkstra algorithm.

Heap optimization can effectively reduce the running time of the algorithm. The idea is to use a priority queue method. The main idea of this method is that every pop-up element must be the smallest element in the whole queue, and the smallest element replaces the shortest distance edge of each search, that is, using adjacency table instead of adjacency matrix, heap optimization can greatly reduce the calculation time. The heap optimization method is implemented as follows: Firstly, a priority queue needs to be defined. The priority queue stores and quickly finds the closest point. The elements of the queue are the node number and the distance from the node to the next node. Secondly, the starting point needs to be initialized, which is the starting point is added to the priority queue. The number of the starting point is the node number of the element in the priority queue. At this time, the distance between the node and the starting point in the calculation process is 0. Finally, if in another calculation process, a node in the priority queue reaches The shortest distance of the starting point has changed. The elements in the original priority queue need not be deleted, but the shortest node element after the change is stored again as the priority queue and popped as the smallest element. The flowchart of the improved Dijkstra algorithm is shown in Fig. 4.

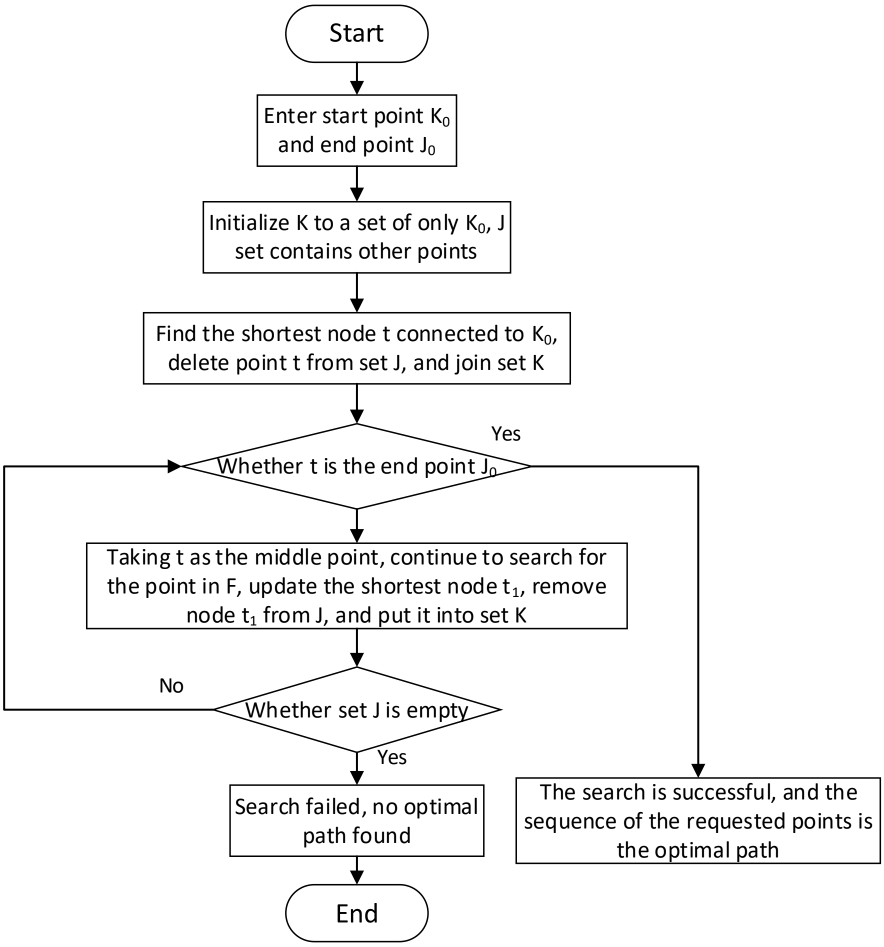


Fig. 4 Improved Dijkstra algorithm flowchart

4. Experimental analysis

4.1 Experiment comparison and analysis of different control modes

The one-way single channel path network as shown in Fig. 5 is established, and the starting point and terminals point are selected by the operation mode of quay crane to yard crane and yard crane to quay crane, among which 4, 8 and 12 are the location of quay crane, 62, 64, 66, 68, 70, 72 and 74 are the location of yard crane. There are 42 combinations of quay crane to yard crane and yard crane to quay crane, corresponding to 42 AGVs; the constant speed of each AGV is 2m/s; the acceleration / deceleration of AGV is 0.5 m/s^2; the length of AGV is 2m; the conflict detection distance of AGV is 2m; the number of map vertices is 75; the number of experiments is 100; the control program of multi AGV system is written in C++, which is implemented on Windows 10 computer with inter (R) core (TM) i7-8750h CPU @ 2.20GHz 2.21 GHz and 16GB memory.

The task is randomly assigned to 42 AGVs, and the path of each AGV is generated by improved Dijkstra according to the starting point and terminals point, as shown in Table 1. The departure time of multiple AGVs starting from the same vertex is 2s in turn, that is, vehicle 1 starts at 0s, vehicle 2 starts at 2s, and vehicle 3 starts at 4s. At the same time, the path of 42 AGVs must contain all the paths from the starting point, so as to avoid 42 AGVs choosing a path without conflict.

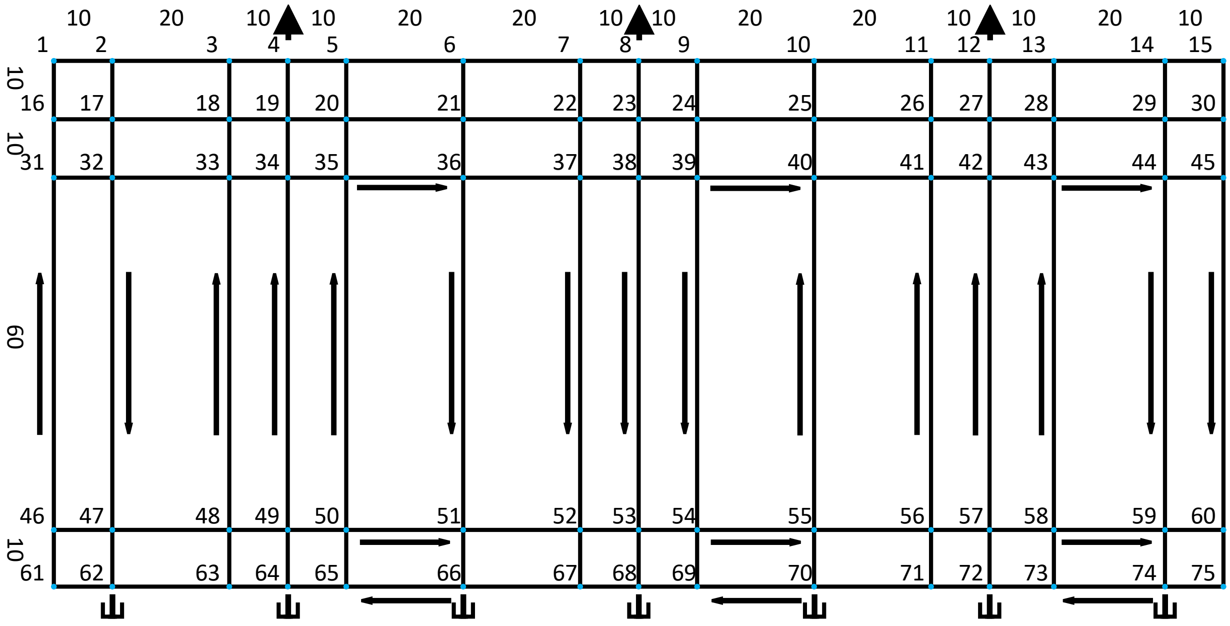


Fig. 5 AGV path map

Table 1 Path start point and end point table

starting point	ending point	starting point	ending point
4	62、64、66、68、70、72、74	66	4、8、12
8	62、64、66、68、70、72、74	68	4、8、12
12	62、64、66、68、70、72、74	70	4、8、12
62	4、8、12	72	4、8、12
64	4、8、12	74	4、8、12

The congestion rate distribution of MAS based control mode and the task priority control method is shown in Figure 6. The control mode of task priority mainly refers to the decision-making of collision avoidance based on the priority of each AGV. The AGV with low priority needs to give way to other AGVs in the process of collision avoidance. The dotted line in the figure represents the control method of task priority, and the solid line represents the control method based on MAS in this paper. In 100 experiments, 42 AGVs are randomly assigned to starting and ending points for each control mode, and the same conditions are simulated. Fig. 6 shows that the blocking rate of the control method based on MAS is lower than that of the task priority control method as a whole.

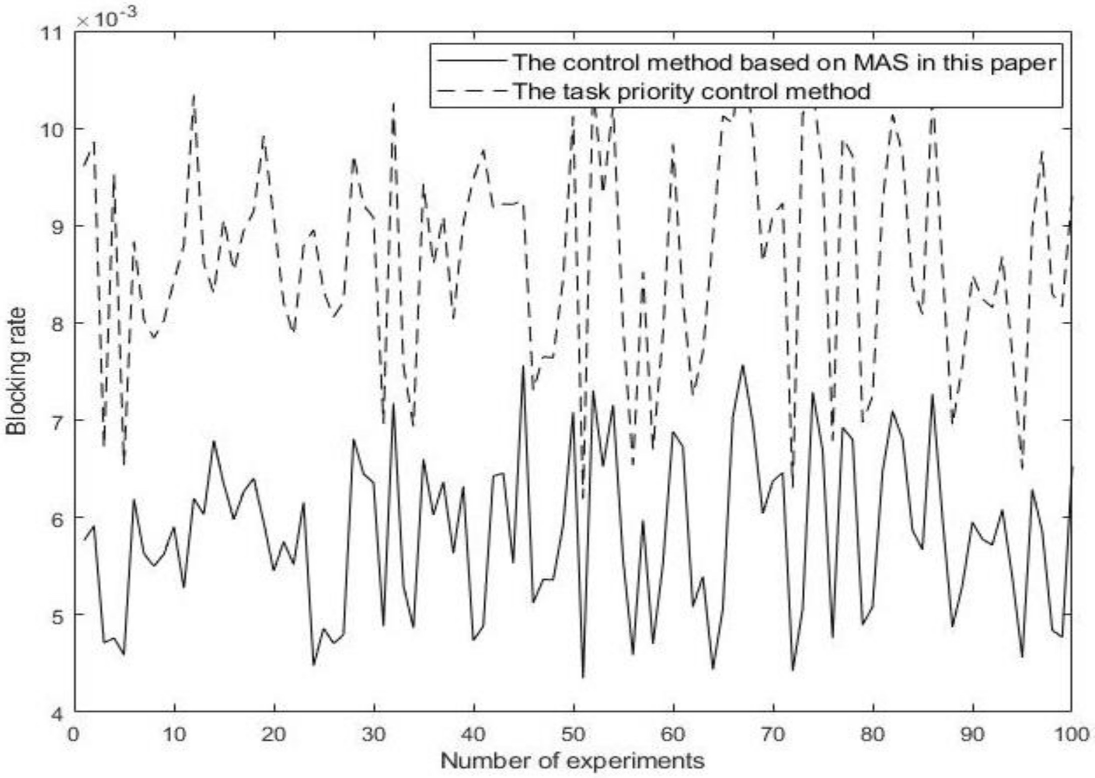


Fig. 6 Blocking rate of AGVs with different control modes

Table 2 Experimental data sheet of different control modes

Control mode	Average waiting time	Average blocking rate
Task priority	36.1s	0.87%
Control mode based on MAS	24.3s	0.59%

Combined with table 2, compared with the task priority control method, the MAS based control method reduces the average waiting time by 11.8s and the average blocking rate by 0.28%, which is obviously superior to the task priority control method.

At the same time, this paper considers the changes of average blocking rate, average waiting time and average completion time under different number of AGVs.

Fig. 7 shows that: (1) when the number of AGVs is 20-30, the difference between the task priority control mode and the MAS based control mode is 0.08%~0.15%, and the average blocking rate curve is nearly parallel, because the number of AGVs is relatively small, the conflict frequency is low and the conflict will not affect the system operation efficiency. (2) When the number of AGVs is more than 30, especially 42, the average blocking rate of MAS based control mode decreased by 0.28%, and compared with the average blockage rate of 30 AGVs, the blockage rate is reduced by nearly 2 times. The results show that the control method based on MAS can reduce the conflict resolution time and the impact of conflict AGV on the system operation.

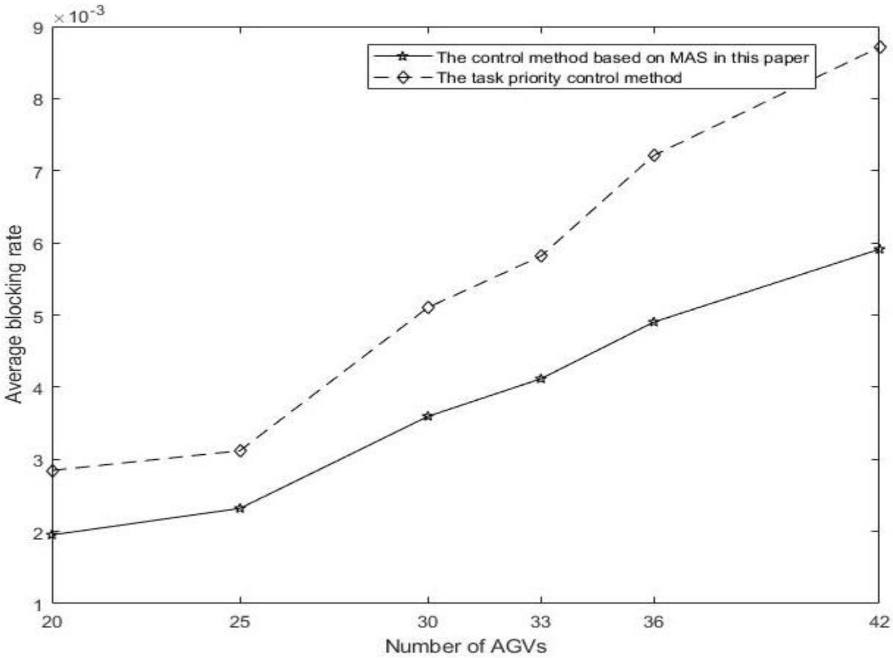


Fig. 7 The change chart of average blocking rate corresponding to different numbers of AGVS

Table 3 Average completion time of different numbers of AGVs

Control mode Number of AGVS	20	25	30	33	36	42
Task priority	1974.8s	2507.5s	2985.9s	3370.9s	3709.2s	4179.2s
Control mode based on MAS	1972.9s	2505.4	2981.1s	3364.8s	3701.3s	4167.4s

From the vertical view of Table 3, the average completion time difference of the two control modes is 1.9s, 2.1s, 4.8s, 6.2s, 7.9s and 11.8s. The time difference between the two control modes increases with the increase of AGV number, and the increasing range is larger and larger, which shows that the control mode based on MAS can significantly reduce the operation time of more than 30 AGVS and improve the operation efficiency of AGVs.

4.2 Comparative experiment and analysis of AGVs with different scale maps

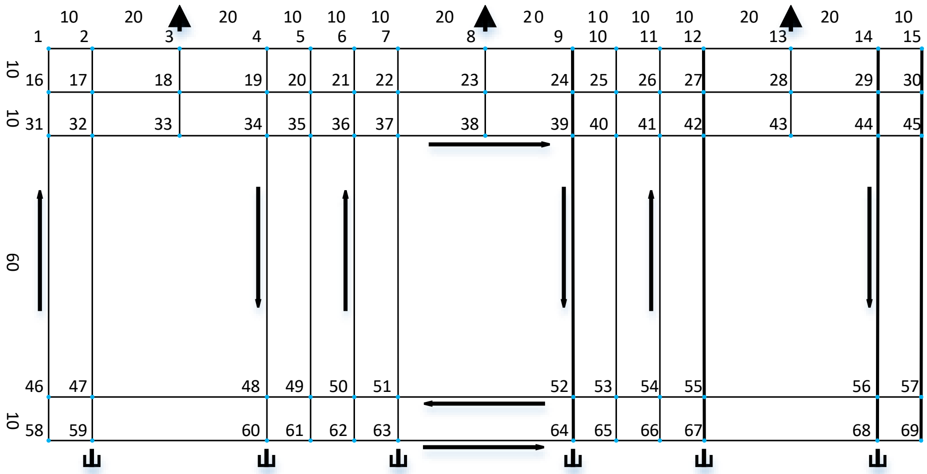


Fig. 8 Smaller map

A small-scale AGV operation path network is established as shown in Fig. 8, in which 3, 8, 13 are the locations of quay crane, 59, 60, 63, 64, 67, 68 are the locations of yard crane. There are 36 combinations of quay crane to yard crane and yard crane to quay crane, corresponding to 36 AGVs, and the method in Section 4.1 is the same.

The average blocking rate of AGVs with different scale maps is shown in Fig. 9. According to table 4 :(1) When the number of AGVs is between 20 and 30, the average blocking rate of AGV in the 75-node map and the 69-node map under the MAS-based control mode changes to 0.03%, 0.01%, and 0.01%, under the task priority control mode the average congestion rate changes are 0.03%, 0.02%, and 0. The broken line in the figure shows that the two broken lines of the 75 node map are significantly lower than those of the 69 node map, mainly due to the increase of nodes, resulting in the corresponding increase of the completion time of AGV after the allocation of nodes, so the average blocking rate is relatively low. (2) When the number of AGVs exceeds 30, the average blocking rate of the AGV in the 75-node map and the 69-node map under the MAS-based control method changes from 0 to 0.01%, and the average blocking rate under the task priority control method changes from 0 to 0.03%. The broken line in the figure shows that the average congestion rate of AGV increases with the increase of nodes, but the increase of the average blocking rate based on MAS is significantly 3 times lower than that of task priority, so the control method based on MAS is more suitable for solving the conflict free road planning problem of more than 30 AGVs.

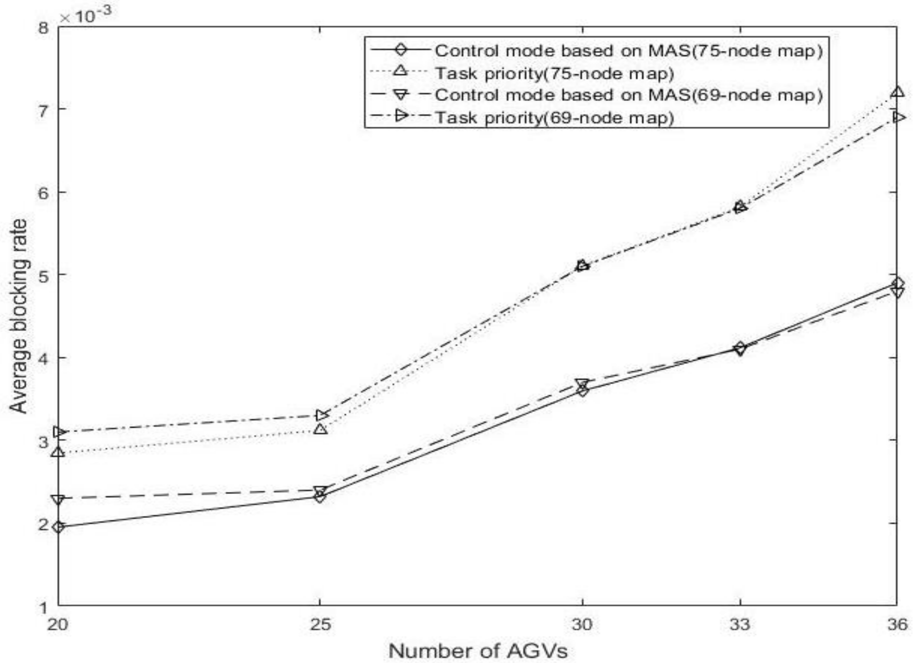


Fig. 9 Average blocking rate of different scale maps

Table 4 Average blocking rate of AGVs with different scale maps

Control mode / Number of AGVs	20	25	30	33	36
Task priority (75-node map)	0.28%	0.31%	0.51%	0.58%	0.72%
Control mode based on MAS (75-node map)	0.20%	0.23%	0.36%	0.41%	0.49%
Task priority (69-node map)	0.31%	0.33%	0.51%	0.58%	0.69%
Control mode based on MAS (69-node map)	0.23%	0.24%	0.37%	0.41%	0.48%

5. Conclusion

In this paper, a multi-AGV conflict-free path planning model is established, minimizing the AGV operation blocking rate, considers the AGV's driving speed, operation time and conflict distance. At the same time, AGV system based on MAS is designed, and improved speed control is used as AGV's negotiation strategy. The control experiments based on MAS and task priority are designed to compare the average blocking rate, average waiting time and average task completion time under the same conditions, as well as AGV average blocking rate comparison experiments under different scale maps. The result of experiment 1 shows that when the number of AGVs is less than 30, the two control methods differ by 0.08%~0.15% in average blocking rate, and the change range is small. When more than 30 AGVs, the control method based on MAS has an average blocking rate reduced by 0.15%~0.28%, the amplitude changes greatly, meanwhile, the average waiting time of AGV based on the MAS-based control method is lower than that of the task priority control method by 11.8s, and the average completion time is reduced by 11.8s. The results of experiment 2 show that when the number of AGVs exceeds 30, the average blocking rate change range under the MAS-based control mode is 0~0.01%, which is lower than 0~0.03% under task priority control. The experimental results verify the feasibility of the model in this paper, and the control method based on MAS is more suitable for solving the conflict-free path planning problem of more than 30 AGVs.

Acknowledgements

This research was supported by the Shanghai Pujiang Program (16PJC043).

References

[1] WANG Yu, XIE Feng, LIU Yu-lei, ZHU Xiang. Research of automatic guided vehicles path planning based on improved ant colony algorithm with potential field[J]. Manufacturing Automation, 2019,41(07): 70-74.

[2] FANG Hua, ZHOU Qi-cai, XIONG Xiao-lei, ZHAO Jiong. Research on 3D path planning of universal warehousing logistics based on A~* algorithm[J]. Manufacturing Automation, 2018, 40(07):26-30+53.

[3] ZHANG Su-yun, YANG Yong-sheng, LIANG Cheng-ji, XU Bo-wei, LI Jun-jun. Optimal control of multiple AGV path conflict in automated terminals[J]. Journal of Transportation Systems Engineering and Information Technology, 2017, 17(2): 83-89.

[4] ZHONG Mei-su, YANG Yong-sheng, ZHOU Ya-min. Free-conflict AGV path planning in automated terminals based on speed control[J]. Computer Science, 2019, 46(07): 308-314.

[5] CAO Xiao-hua, ZHU Meng. Research on Multi-AGV conflict avoidance decision optimization method based on conflict prediction[J/OL]. Computer Integrated Manufacturing Systems, 2019, 08(51): 1-11.

[6] LIU D K, WU X, KULATUNGA A K, et al. Motion coordination of multiple autonomous vehicles in dynamic and strictly constrained environments[C]. 2006IEEE Conference on Cybernetics and Intelligent Systems, IEEE, 2006: 1-6.

[7] LI Xiao-meng, YANG Yu-pu, XU Xiao-ming. Multi agent AGV Dispatching System Using Multi level Decision Methods[J]. Journal of Shanghai Jiao Tong University, 2002, 36(8): 1146-1149.

[8] Koen H. Van Dam, Zofia Verwater-Lukszo, Jaap A. Ottjes, Gabriel Lodewijks. Distributed intelligence in autonomous multi-vehicle systems[J]. International Journal of Critical Infrastructures, 2006, v2, p261-272.

[9] Fauadi Muhammad Hafidz Fazli Bin Md, Lin Haowen, Murata Tomohiro. Dynamic task assignment of autonomous AGV system based on multi agent architecture[C]. Proceedings of the 2010 IEEE International Conference on Progress in Informatics and Computing, 2010, v2, p1151-1156.

[10] Erol Rizvan, Sahin Cenk, Baykasoglu Adil, Kaplanoglu Vahit. A multi-agent based approach to dynamic scheduling of machines and automated guided vehicles in manufacturing systems[J]. Applied Soft Computing Journal, 2012, v12, p1720-1732.

[11] Jing Jian-feng. Study and development of autonomous control multi-AGV system based on agent[D]. Nanjing University of Aeronautics and Astronautics, 2014.