IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 25, 2020, accepted December 15, 2020, date of publication December 30, 2020, date of current version January 7, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3047916

An Improved Acceleration Method Based on Multi-Agent System for AGVs Conflict-Free Path Planning in Automated Terminals

KUNLUN GUO, JIN ZHU, AND LEI SHEN
Institute of Logistics Science and Engineering, Shanghai Maritime University, Shanghai 201306, China
Corresponding author: Jin Zhu (jinzhu@shmtu.edu.cn)

This work was supported by the Shanghai Pujiang Program under Grant 16PJC043.

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. An architecture of AGV's system based on Multi-Agent System (MAS) is designed, the improved interactive protocol based on blackboard model is used as the communication method of AGV, the improved acceleration control method is combined with the AGV priority determination method based on time cost as the negotiation strategy of AGV, the improved Dijkstra algorithm calculates the conflict-free path of each AGV. By comparing the acceleration control method based on MAS with the speed control method based on MAS and the task priority control method, the effectiveness of this method for solving multiple AGVs conflict-free path planning in automated terminals is verified.

INDEX TERMS Automated terminals, multi-AGV, multi-agent system (MAS), improved acceleration control method, conflict-free.

I. 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. The problem of automated terminals is more concerned and urgently need to be resolved.

The problem of multi-AGV conflict-free path planning is concerned and investigated by many scholars, such as using Dijkstra or swarm optimization algorithm, Lyu *et al.* [1] proposed a genetic algorithm combined with the Dijkstra algorithm that is based on a time window, which is to solve the machine and AGV scheduling problem in a flexible manufacturing system by considering the optimal number of AGVs, the shortest transportation time, the problem of path planning, and the conflict-free routing problem, is based on a time window. Zhang *et al.* [2] proposed a collision-free routing method for AGVs based on collision classification, using Dijkstra algorithm to plan the collision free path for each collision. Cao and Zhu [3] 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. Centralized control method can not solve the path planning problem of multi AGV, distributed control method combined with intelligent optimization algorithm to solve conflict problem has become the mainstream. Similarly, in addition to the research on workshop and logistics warehouse AGV, there are more and more documents about the conflict-free path planning of automated terminal AGV, for example, Zhang *et al.* [4] established a multi-parameter optimization control model, which considered the number of AGVs in the path, the safe distance and speed

The associate editor coordinating the review of this manuscript and approving it for publication was Heng Wang.

of AGV, and improved the speed control strategy to resolve conflicts. Zhong *et al.* [5] considered the driving speed of AGV, the reload rate and conflict time of AGV, 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, using speed control strategies to resolve conflicts. Liu *et al.* [6] set the shortest task completion time of AGV as the goal, and considered the travel speed of AGV to locate and avoid conflict constraints. It is verified that the model and algorithm by the simulation to avoid AGV conflict and congestion. Zhong *et al.* [7] established a mixed-integer programming model based on path optimization, integrated scheduling, conflict, and deadlock, which is to solve the conflict-free path planning and integrated scheduling problem of multiple AGVs and minimize the delay time of AGVs under known tasks. Li *et al.* [8] studied the impact of the uncertainty of the driving time on the AGV's catch-up conflict considering the uncertainty of the driving time of the AGV, designed a dynamic adjustment based on the AGV's conflict probability, which verified the probability characteristics and the effectiveness of the performance related to dynamic adjustments. Singgih *et al.* [9] studied the AGV path planning problem of automated container terminals with consideration of congestion. An integer programming model was constructed to minimize the transiting time and wait time of the AGV, which was solved by the Dijkstra algorithm. Legato *et al.* [10] built an optimization model to minimize the collision time caused by the AGV path conflict, with consideration of the non-constant efficiency of the quay crane, and solved it using a simulated annealing algorithm. Yang *et al.* [11] studied the integrated scheduling of quay crane, AGV, and yard crane for simultaneous loading and discharging operations, and designed a general algorithm based on preventive congestion rules. Xin *et al.* [12] proposed a collision-free scheduling algorithm to generate timetables for AGV, quay crane, and yard crane, which could reduce the average distance of AGV operation.

It can be obtained from the above literature that for the multi-AGV path planning problem, even for the multi-AGV conflict-free path planning in the automated terminal, most scholars have adopted the idea of distributed control to solve the problem. That is when the node becomes a conflict, according to the different priorities, adopt the methods of parking and waiting, speed control to solve the conflict problem. However, the number of AGVs for AGV conflict-free path planning of automated terminals studied in the above literature is 20-30, while the actual number of AGVs in China's automated terminals is 18, 38, and 50, especially the number of AGVs in Yangshan Phase IV, which is designed to reach 130. Therefore, it is more practical to study the conflict-free problem of AGVs for more than 30 automated terminals, and it is more necessary to design a mathematical model of the conflict-free path for AGVs of more than 30 in automated terminals.

In flexible manufacturing system and warehouse, it is generally believed that agents are independent individuals in distributed systems or cooperative systems that could continuously make autonomous decisions and actions. The agent has the characteristics of autonomy, interaction, reactivity, and initiative [13], [14]. Many scholars apply the multi-agent system to the AGV operation process, that is, a cooperative intelligent system is established, where an intelligent software agent "moves" from one physical platform to another (e.g., due to a low battery event in the current platform) by using the agent mobility capabilities [15]. Herrero-Pérez *et al.* andMartinez-Barbera [16] proposed a methodology for modeling and controlling a flexible material handling system (MHS), composed of multiple automated guided vehicles, suitable for Flexible Manufacturing Systems. Branisso *et al.* [17] proposed a multi-agent AGV system, which successfully reduced the average waiting time of tasks by using a fuzzy scheduling rule and compared it with simple rule FCFS and CNET protocol. The results show that compared with the other two decision-making methods, the fuzzy decision-making method can reduce the average waiting time of tasks more effectively. Other works like [18] made use of a service-oriented multi-agent platform [19] for the analysis, design, and implementation of complex systems where the data sources and data processing are distributed. Frego *et al.* [20] proposed a combined minimum time - minimum jerk traffic management system for the vehicle coordination in an automated warehouse, a piecewise constant velocity profile was decided for each agent to guarantee the execution of their missions and to avoid collisions. Koen *et al.* [21] introduced the concept of agents and proposed a way to improve the accident handling of multi-AGV systems through the application of cooperative control. Rizvan *et al.* [22] proposed the cooperative scheduling of AGV and manufacturing system based on MAS in a real-time environment, took the bidding mechanism in MAS as the negotiation strategy of multi AGV system.

The architecture of MAS refers to the organization relationship and control relationship among agents in the system, as well as the distribution mode of problem-solving [23]. Architecture is a very important aspect of the system. The architecture defines the functional roles, relationships, and interaction mechanisms of each agent in the system. The architecture also includes environmental information and the communication mode between agents. Jing [24] 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 when conflicts occur to the task. It can infer the next action according to its perception of the external environment, including the environmental information obtained by sensors and the communication information with other AGV agents. At the same time, the system needs AGV to communicate with other AGVs and control center, and the communication mode of the blackboard model [25] meets this requirement. However, this communication mode requires communication between each AGV. If there is no conflict between any two or more AGVs, the system resources will be wasted. Finally, when each AGV completes its task independently, it will conflict with other AGVs, such as competing for paths and resources. Micieta *et al.* [26] improved the entrusted MAS by adding Path Load pheromone, road pheromone, intersection decision-making pheromone, and compound reservation pheromone to solve some coordination problems called live-lock and dead-lock. Here, the commonly used methods to resolve conflicts include parking and waiting method [27], priority selection based on priority [28], and so on.

The above documents show that in flexible manufacturing system and warehouse, many documents apply agents to multiple AGV systems, each AGV agent is modeled, and corresponding responses and operations can be generated to the surrounding environment so that the AGV can coordinate and complete the task. Resolve conflicts through negotiation, improve the efficiency of the AGV system, and reduce time wasted. However, compared with FMS and warehouse, automated terminal is obviously different. The length of AGV, the scale of the AGV operation map, the interaction between AGV and control platform should be considered and designed according to the situation of the terminal.

There is little literature about the application of MAS in automated terminals. Henesey *et al.* [29] developed an agent-based simulator to evaluate tape cartridge-based systems and compare them to more traditional AGV systems. Also, to find the most efficient configuration, many different configurations of container terminal equipment, such as the number of AGVs and cassettes, had been studied. Henesey [30] also developed a simulation tool called SimPort for evaluating container terminal management policies. The methods for modelling the entities in a container terminal were presented along with the simulation experiments conducted. The results show that some certain strategies can shorten the turnaround time of ships, and certain stacking strategies can increase productivity. However, the above-mentioned research mainly focuses on the resource allocation problem of the automated terminal, and does not involve the conflict-free path planning of the AGV.

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 more than 30 AGVs conflict-free path planning. At the same time, this paper designs a multi-AGV architecture based on MAS and improves the Dijkstra algorithm to plan AGV's paths, improves acceleration 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. Three comparative experiments are designed based on MAS control mode and task priority control mode to compare the average blocking rate, average waiting time, the average completion time of AGV under different numbers, and the average blocking rate of multi AGV under different scale maps.

II. MATHEMATICAL MODEL

A. MODEL ASSUMPTIONS

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

2) Setting up a buffer zone at the quay crane and yard crane, 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 crane 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.

B. 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.

Ls 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.

Dk is the length of the path k, $k = 1,2,\ldots,$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_{dm} is the waiting time of the m-th AGV at the conflict node.

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.

C. 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.

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

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

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

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

After the task is assigned to the AGV, the driving speed does not change during the operation of the AGV, and to avoid conflicts between multiple AGVs on the same path from the same starting point, the AGV needs to maintain a minimum safety distance, 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 meet 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.

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

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

1) IMPOVED SPEED CONTROL METHOD

After negotiation, the AGVs adjust according to their own priority. The AGV with low priority begins to slow down before reaching the collision node, while the AGV with higher priority passes through the collision node at a constant speed and reaches the detection threshold of the collision AGV. Eq (7) is the linear safety distance between two AGVs in different paths $k1$ and $k2$, which is less than the collision detection range distance; Eq (8) is the deceleration distance ls from $v0$ to $v1$ for AGV with low priority; Eq (9) is the deceleration time $td1$ of AGV with low priority and the time $td1'$ when AGV returns to the original speed after the conflict is resolved; Eq (10) is the deceleration of AGV with low priority, and AGV will not stop and wait in the end; Eq (11) is the relationship between the distance l' of AGV with high priority and the deceleration distance ls of AGV with low priority after solving the conflict; Eq (12) is the time delay of the m-th AGV due to conflict among n AGVs; Eq (13) is the time when the m-th AGV arrives at the terminal due to w times of conflicts; Eqs (14) ∼ (15) 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 (16) is the minimization of the ratio TJ of total waiting time to task completion time, that is, minimizing the blocking rate during n AGVs operation.

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

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

$$t_{d1} = t'_{d1} = \frac{v_0 - v_1}{\alpha} \tag{9}$$

$$v_1 = v_0 - \alpha t_{d1} (v_1 > 0) \tag{10}$$

$$\begin{cases} (l' - \sqrt{2}R)^2 + (\sqrt{2}R - ls)^2 \geq 4R^2 \\ l' - ls \geq 2R \end{cases} \tag{11}$$

$$t_{dm} = \begin{cases} t_{d1}, & \text{time delay due to conflict} \\ 0, & \text{there is no conflict or conflict, but AGV has high priority} \end{cases} \tag{12}$$

$$T_m = t_{kmep} + w(td1' + td1) - \frac{2\omega\, l_s}{v_0} \tag{13}$$

$$\sum_{k=1}^{n} t = t_{d1} + t_{d2} + \ldots + t_{dn} + t'_{d1} + t'_{d2} + \ldots + t'_{dn} \tag{14}$$

$$\sum_{k=1}^{n} T = T_1 + T_2 + \ldots + T_n \tag{15}$$

$$\min TJ = \frac{\sum_{k=1}^{n} t}{\sum_{k=1}^{n} T} \tag{16}$$

2) IMPOVED ACCELERATION CONTROL METHOD

Considering the deceleration of AGV with low priority, the AGV with higher priority accelerates. When the AGV with higher priority passes through the collision node and meets the safety distance between AGV and AGV with low priority, both sides recover to the original speed. The speed control model of AGV with low priority is consistent with the above model. Eq (17) is the distance la from $v0$ to $v2$ for the AGV with high priority; Eq (18) shows that the acceleration time of high priority is the same as that of AGV with low priority; Eq (19) is the relationship between the acceleration distance of AGV with high priority and the deceleration distance of AGV with low priority; Eq (20) is the time to complete the task after w times of conflicts (in case of low priority) and w' times of conflicts (in case of high priority) of the m-th AGV; Eqs (21) ∼ (22) are the waiting time of n AGVs in the path due to conflict delay, and the total time for AGVs to complete the task; Eq (23) is to minimize the blocking rate during the operation of AGVs, the ratio of total waiting time and task completion time.

$$l_a = \frac{v_2^2 - v_0^2}{2\alpha} \tag{17}$$

$$t_{d1} = \frac{v_0 - v_1}{\alpha} = t_{a1} = \frac{v_2 - v_0}{\alpha} = t'_{a1} \tag{18}$$

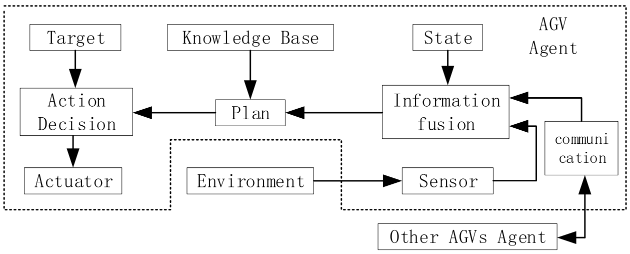


FIGURE 1. Agent modeling on AGVs.

$$\begin{cases} (la - \sqrt{2}\,R)^2 + (\sqrt{2}R - ls)^2 \geq 4R^2 \\ la - ls \geq 2R \end{cases} \tag{19}$$

$$T_m = t_{kmep} + w(td1' + td1) - \frac{2\omega\, l_s}{v_0} + w'(ta1' + ta1) - \frac{2\omega\, l_a}{v_0} \tag{20}$$

$$\sum_{k=1}^{n} t = t_{d1} + t_{d2} + \ldots + t_{dn} + t'_{d1} + t'_{d2} + \ldots + t'_{dn} \tag{21}$$

$$\sum_{k=1}^{n} T = T_1 + T_2 + \ldots + T_n \tag{22}$$

$$\min TJ = \frac{\sum_{k=1}^{n} t}{\sum_{k=1}^{n} T} \tag{23}$$

III. MAS SUIT FOR AGVS IN AUTOMATED TERMINALS

A. AGENT MODELING ON AGVS SYSTEM

The agent is a knowledge-based system, which can contain environment description and rich intelligent behavior logical reasoning ability. Taking AGV as an agent, it is necessary to combine multiple independent parallel executions in one agent. Its structure includes perception, action, response, modeling, planning, communication, and decision-making modules. As shown in Fig.1, the AGV makes a preliminary decision by using the improved Dijkstra algorithm according to the starting point and ending point of the transportation task to start the transportation task. Next, according to the environment, AGV makes an abstraction of the environment information by the sensor, and then sends it to the next processing module. If there is no interference, the AGV will continue to drive and complete the transportation task. If there is a conflict, AGV will fuse the information from the sensor and other AGVs, and negotiate according to the knowledge base, namely priority, and speed / acceleration control method, to solve the conflict and continue to complete the transportation task.

B. MODELING OF AGV RUNNING MAP

When MAS is applied on the AGVs in the automated terminals, the AGV as an agent can determine its position through the sensor, and then compare it with the known coordinate value road signs, to match with the global coordinate system to obtain its 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 AGVs 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. Besides, 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:

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

C. IMPROVED INTERACTION PROTOCOL OF BLACKBOARD MODEL

In the architecture of MAS, the blackboard model can manage the global resources to a large extent, at the same time, it will produce a lot of redundant information. In this paper, when assigning tasks from the system, considering the nodes that are expected to have conflicts, we quickly search for AGVs agents that arrive at the conflict nodes at the same time or less than 2s interval and put the expected conflicting AGVs agents into the interactive network. In addition, because the travel time of AGV will change after one conflict, so it is necessary to research the nodes that may conflict with each other for AGV with multiple conflicts in one task.

Search for nodes and AGVs pseudocodes that are expected to conflict as follows:

The interaction protocol of improved blackboard model is mainly the communication between AGVs and the communication between AGVs and the console. The AGV needs to

Algorithm 1 Search Nodes and AGVs That Are Expected to Conflict

/*C[b] is the set of conflicting nodes in all paths
I[i][j] = k means that the next target node of the i-th AGV of node j is k
I[a][b] is the set of vertices in the path of a-th AGV
T[a][b] is the set of vertex-time in the path of a-th AGV */
1, Establish the vertex set of all AGVs paths I[A][B];
2, Establish conflict node set C[N];
3, **For** b = 0, b < B// b is the map node number, B is the maximum node number
    C[b] = 0;
4, **For** a = 0, a < A// a is the AGV number and a is the maximum number
5, If I[a][b] = I[a + i][b], then // a + i indicates other AGV, i is greater than or equal to 1
6,   If T[a][b] = T[a + i][b], then
      C[b]++;
    end
  end
end
7, If T[a][b] updates or changes, then **return5**;// If the vertex time of a-th AGV changes due to conflict, the conflict nodes of AGV and other AGVs are recalculated

send and receive information during operation, namely the AGV number, AGV task, current location, current time, and its next node location that the AGV needs to send to the console. The console needs to send the conflicting nodes in the path table to the per-conflict AGV, so that the AGV can calculate the time for the AGV to reach the conflicting node based on the running time and distance. The main interaction information between AGVs is that when AGV conflict, they need to send their respective priority, then request or ask for the next work. The language of exchanging information between AGVs adopts the idea of "stack" protocol, which is divided into two logical levels: content layer and communication layer, as shown in Fig. 2 below.

1) The main function of the blackboard is to assign the quay crane to the yard and the yard to the quay crane path on the map to each AGV, monitor the position, speed, and time of each AGV, and send the position, speed, and time to the pre-conflict AGV.

2) The AGVs running in the system is the source of knowledge in the blackboard model. (a) Each AGV has its operating characteristics, such as possible conflicts and its 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 AGVs 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 AGVs and blackboard interactive information in this article is 0.02s.

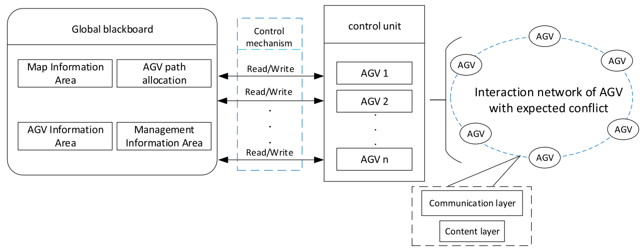


FIGURE 2. Multi-AGV interaction protocol based on improved blackboard model.

4) The content layer is the knowledge information transmitted, which is represented by vector < int >:: iterator it, it is the location, speed, and time information that can be read and written. The communication layer describes a group of data related to both sides of the communication, which is mainly to fuse the encoded information sent by both sides of the conflict and send it to the AGVs for negotiation and comparison.

D. MULTI-AGV CONFLICT NEGOTIATION STRATEGY

The negotiation strategy of multi-AGV is to calculate the position of AGV based on the position and posture of AGV and then negotiate to solve the conflict according to the priority of AGV. In this paper, the priority of AGV is determined based on the time cost method, and the AGV negotiation strategy is compared with speed control and variable acceleration control.

1) AGV PRIORITY DETERMINATION BASED ON TIME COST

This article uses a time-cost-based method to determine the priority of the AGV. Fig. 3 shows that, after assigning tasks to each AGV, using the Dijkstra algorithm to get the shortest path according to the starting point and ending point, calculating the expected completion time of the task and determining 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, and the running time of AGV with high priority is reduced, the priority will change accordingly.

2) CONFLICT NEGOTIATION STRATEGY OF ACCELERATION CONTROL METHODS

The pre-conflict AGV calculates the travel distance and the node to be reached according to the time and path from the starting position. As shown in Fig. 4, the conflict detection sensors of AGV_1 and AGV_2 set the detection radius of R_1 and R_2, and $R_1 = R_2$. The length of R_1 and R_2 is set according to the length of AGV body and path length, and the detection threshold moves with the AGV moving. In (a) of Fig. 4, AGV_1 and AGV_2 drive to the intersection at the same time. The detection thresholds of AGV_1 and AGV_2 do not intersect, so the two AGVs do not detect conflict; when the two AGVs continue to run to the situation (b) in Fig. 4, the two detection thresholds begin to intersect, indicating that AGV_1 and AGV_2

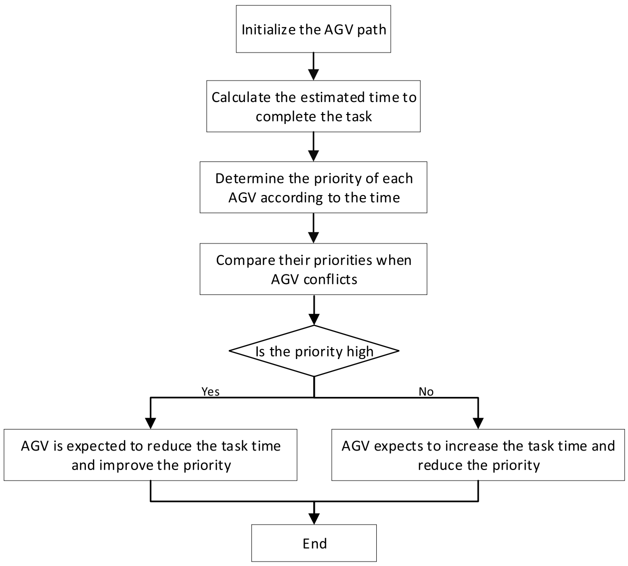


FIGURE 3. Priority flowchart of AGV based on time cost.

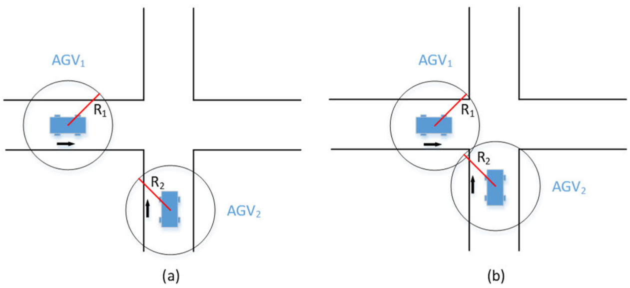


FIGURE 4. AGVs conflict diagram.

intersect at this time, AGV needs to negotiate concession to solve the conflict.

The following is an example of one of the two AGVs in conflict to explain the process of conflict resolution

1) When the critical value of AGV conflict detection begins to intersect with other detection thresholds, the AGV confirms that the collision will occur according to the information provided by the improved blackboard model.

2) AGV extracts the number of the other AGVs, determines the negotiation object AGVs and communicates with it, and requests to pass through the front possible conflict nodes.

3) Both parties obtain the results according to their respective priorities and determine an AGV to lock the conflicting node and speed up the passage of the conflicting node.

4) The AGV with low priority slows down until the higher priority AGV passes through the node, and then recovers to pass through the node to solve the conflict problem.

E. 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 nodes closest to the starting point and never visited before reaching the target point appear in the search range. In AGV path planning, the weight value represents the length of the edge. If two points are not connected, the value corresponding to their weight value is infinite. However, the Dijkstra algorithm traverses and calculates each node, which is inefficient in calculation time and wastes calculation space. Therefore, this article uses heap optimization to improve the 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. 5.

IV. EXPERIMENTAL ANALYSIS

A one-way single channel path network is established as shown in Fig. 6. The mapped network is a test environment that simulates automatic terminal transportation scenarios. In the figure, the cyan number represents the distance between nodes, and the black number represents the node in the path. Nodes 4, 8, and 12 are the quay crane position, and nodes 62, 64, 66, 68, 70, 72, 74 are the yard crane positions. The arrow indicates that all paths have the only direction, i.e. up, down, left, and right.

The starting point and terminals point are selected by the operation mode of quay crane to yard crane and yard crane to quay 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 deceleration of AGV is 2m/s^2 based on task priority; the length of AGV is 1m; the conflict detection distance of AGV is 2m; 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 intel (R) Core (TM) i7-8750h CPU @ 2.20GHz 2.21 GHz and 16GB memory.

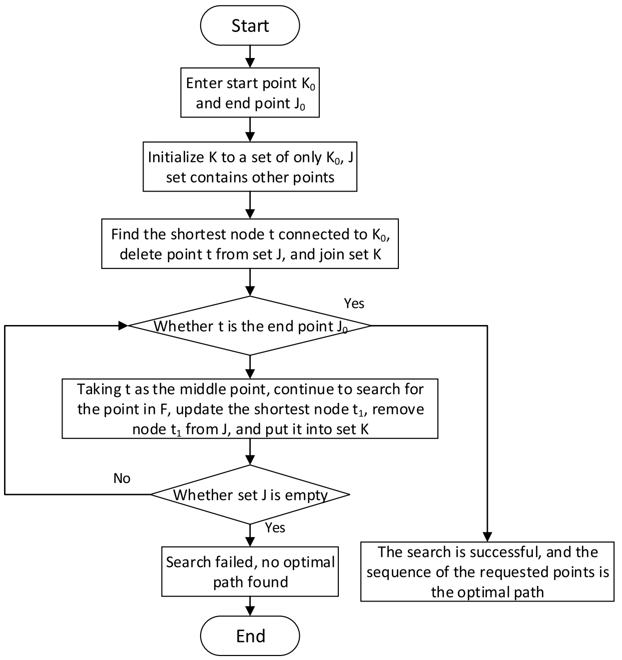


FIGURE 5. Improved Dijkstra algorithm flowchart.

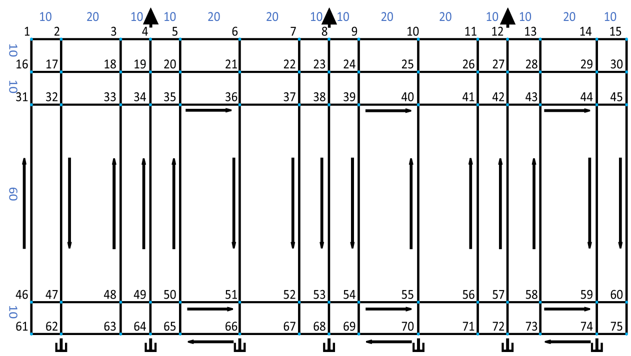


FIGURE 6. AGV path map.

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, to avoid 42 AGVs choosing a path without conflict.

A. COMPARATIVE EXPERIMENT OF THREE METHODS TO CALCULATE THE AVERAGE BLOCKING RATE OF 42 AGVS

An experimental Gantt chart of 42 AGVs paths and conflicts is shown in Fig.7, the horizontal coordinate represents the expected completion time of the AGV, the vertical coordinate represents the number of the AGV. Number in the rectangle represents the node to which the AGV arrives, and number with the red circle is the node that is expected to conflict. The red box represents the path of the conflict.

TABLE 1. Table of starting point and ending point.

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 first conflict: In the Fig.7, we can see that at node 68, AGV_8 and AGV_{40} will conflict. In our multi-agent system, the two AGVs will negotiate before arriving at node 68 to resolve the conflict. When the priority of AGV_8 is high, it will pass through the conflict node first, and AGV_{40} will decelerate to pass through the conflict node.

The second conflict: The time for AGV_{40} to resolve a conflict is 2.56s, and the time for AGV_{40} to pass through the conflict node 68 is 46.16s. That is to say, in the process of conflict resolution by AGV_{40}, the distance between AGV_9, AGV_{40} and AGV_{41} don't meet the conflict conditions. Therefore, at 47s, AGV_9 and AGV_{41} will conflict at node 68. The time for AGV_{41} to resolve a conflict is 2.56s.

The third conflict: When AGV_{40} decelerates and passes through the collision node 68, AGV_9 accelerated through the collision node 68, they will then conflict at node 67 at 51.16s. The time for AGV_{40} to resolve a conflict is 2.56s.

The fourth conflict: At 53.72s, AGV_{40} and AGV_{41} will conflict at node 67, AGV_{41} will decelerates, and AGV40 passes through the collision node.

The fifth conflict: At 62s, AGV_2 and AGV_9 will conflict at node 66.

Finally, the waiting time in the system is 25.6s, and the total time for AGVs to complete the task is 4655s, so the blocking rate is 0.55% according to the Eqs 21, 22, and 23. This result is the minimum value solved by the acceleration control method, which can be found in Fig. 8. In the 100 experiments in this paper, each experiment is to reallocate the path of 42 AGVs. The calculation process is like the description of Gantt chart to get the blocking rate and compare the advantages and disadvantages of the three control methods.

The experimental results of 42 AGVs in different control methods are shown in Fig. 8. The control method of task priority mainly refers to the decision of collision avoidance according to the priority of each AGV. The AGV with low priority needs to stop and wait in the process of collision avoidance to give way to other AGVs. The green dash-dotted line represents the control method of task priority(method1), the blue dotted line represents the speed control method based on MAS(method2) [31] (This method is quoted from my published papers.), and the red solid line represents the

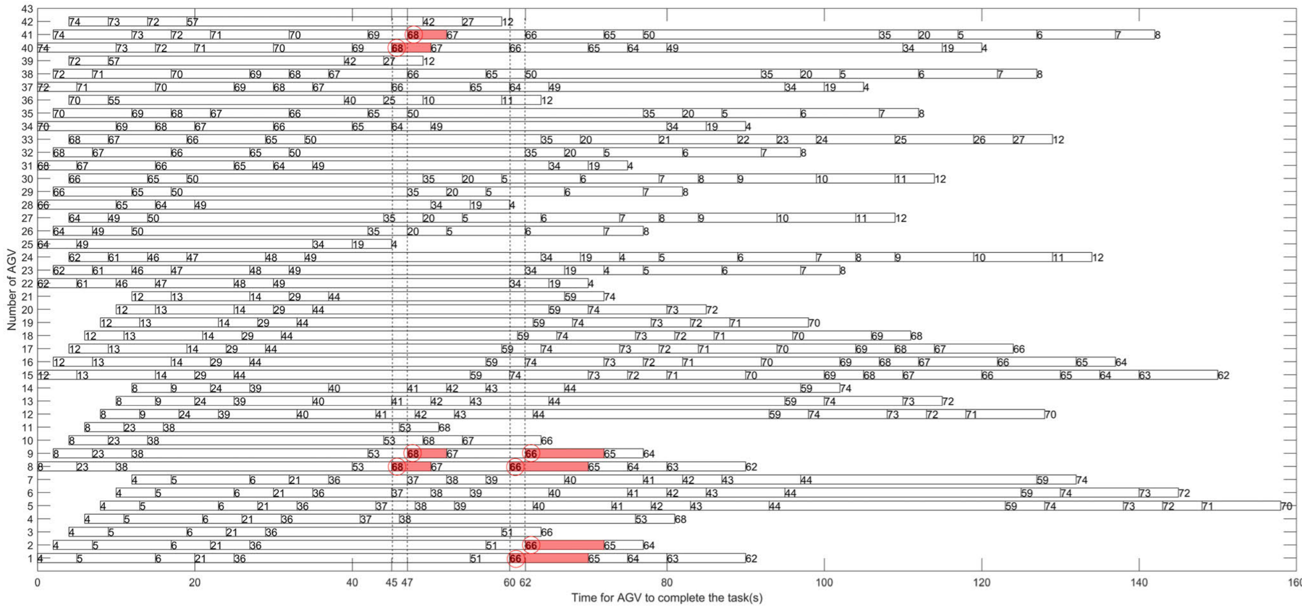


FIGURE 7. An experimental Gantt chart of 42 AGVs paths and conflicts.

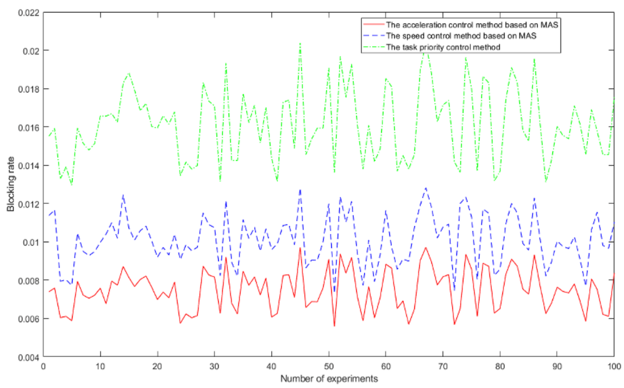


FIGURE 8. Blocking rate of AGVs with different control methods.

acceleration control method based on MAS(method3) in this paper.

In 100 experiments, 42 AGVs were randomly assigned starting point and ending point in each control method, and the same condition was simulated. In Fig. 8, the average blocking rate of AGV in 100 experiments of method1 is 1.29%-2.04%, that of method2 is 0.73%-1.28%, and that of method3 is 0.55%-0.97%.

According to Table 2, (1) from the time to resolve a conflict, method3 is significantly shorter than the other two control methods, which are 0.78s and 2.64s, respectively, which takes less time to resolve the conflict.

(2) From the perspective of average waiting time, the average waiting time of method2 is 13.6 times the time to resolve a conflict, which means that an average of 13.6 conflicts occurs per experiment in 100 experiments. The average waiting time of method3 is 12.1 times the time to resolve a conflict, and there are 12.1 conflicts per experiment on average, which is lower than the number of conflicts in method2. It shows that in the process of resolving the conflicts of 42 AGVs, method2 cannot quickly resolve the conflicts, which in turn affects other AGVs through the conflicting nodes and generates new conflicts. At the same time, the average waiting time of method3 is lower than that of method1.

TABLE 2. Experimental data table of 42 AGVs under different control methods.

Control method	Time to resolve one conflict	Average waiting time	Average blocking rate
Method1	5.2s	72.8s	1.74%
Method2	3.34s	45.5s	1.09%
Method3	2.56s	31.2s	0.75%

(3) The average blocking rate of the method3 is lower than the other two control methods, especially compared with the control method1, the average blockage rate of the system is reduced by 1/2. Therefore, the control method3 can reduce the average blockage rate of the system compared with the other two methods, especially compared with method2, it can deal with the conflict faster and reduce the possibility of secondary conflict.

The experimental data set is the same as the experimental data in reference [6], the constant speed of each AGV is 5.26m/s, the path and time of arrival are shown in Table 3. The bold number 49 is conflict node, numbers 49 and 54 are conflict path. Method3't represents the time to arrive at the node calculated by method3, SiPaMoP't represents the time to arrive at the conflict node calculated by SiPaMoP method proposed in reference [6].

Assuming that the mission emergency of AGV is not taken into account, from the below Table, we can see that the

TABLE 3. Table comparing experimental data with reference [6].

AGV_1				AGV_2			
Start node	End node	Arrival time	Meth od3't	Start node	End node	Arrival time	SiPa MoP't
65	65	0	0	17	17	0	18.5
65	49	6.3	12.2	17	33	3.8	22.3
49	**54**	8.7	14.6	33	54	8.6	27.1
54	33	13.5	19.4	**54**	**49**	11	29.5
33	34	18.5	24.4	48	47	16.5	35
34	35	23.5	29.4	47	31	20.5	39
35	19	27.3	33.2	31	15	24.1	42.6
				15	16	26.1	44.6

SiPaMoP method proposed in reference [6] is to let AGV_2 replan the path to solve the path conflict with AGV_1 from node 49 to node 54. The time to resolve the conflict is increased by 18.5s (the comparison of data is generated by the two columns of SiPaMoP't and the second group of Arrival time in Table 3), that is, the completion time of AGV_1 and AGV_2 is increased by 18.5s. With the Method3 proposed in this paper, AGV_1 and AGV_2 negotiate before reaching node 49. AGV_2 accelerates to pass through the conflict path first, and AGV_1 decelerates. Finally, the running time of AGV_1 is increased by 5.9s (the comparison of data is made by the two columns of Method3't and the first group of Arrival time in Table 3). In other words, the completion time of two AGVs is increased by 5.9s, which is 12.6s less than that of the SiPaMoP method. It is verified that the acceleration control method based on MAS can quickly solve the conflict of AGV.

B. COMPARATIVE EXPERIMENT OF AVERAGE BLOCKING RATE, WAITING TIME AND COMPLETION TIME ON DIFFERENT NUMBERS OF AGVS

At the same time, this paper considers the changes of average blocking rate, average waiting time and average completion time of different number of AGVs under three control methods, as shown in Fig. 9 and Table 4. In Table 4, GAP is the increase in the average blockage rate of method 1 and method 2, which is compared with the average blockage rate of method 3. $ABR(n)'$ is the average blocking rate of method1 $ABR(n)$ is the average blocking rate of method3. $ABR(n)''$ is the average blocking rate of method2.

Fig. 9 shows that (1) when the number of AGVs is less than 30, the average blockage rate curve amplitude of method3 is smaller than that of method2 and method1. In Table 4, the average blocking time of the three control methods are 9.7s-14.5s, 6.5s-9.1s, and 4.9s-7.3s, respectively, which account for 0.49%-0.59%, 0.33%-0.36%, and 0.25%-0.29% of the average task completion time, indicating that when the number of AGVs is less than 30, the number of conflicts during AGV operation is less.

(2) When the number of AGVs reaches or exceeds 30, In Table 4, when the number of AGVs is 30-36, the average blockage rate of the three control methods increased by

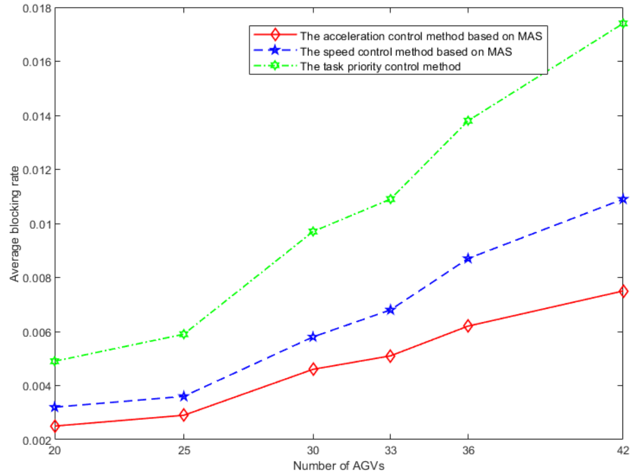


FIGURE 9. The change chart of average blocking rate corresponding to different numbers of AGVs.

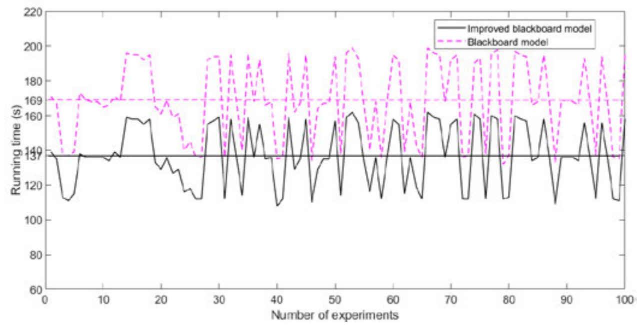


FIGURE 10. Running time.

0.49%, 0.29%, and 0.16%, obviously, the method3 increased less. At the same time, the average waiting time and average completion time of method3 are less. It shows that the average blockage rate of the system begins to increase significantly when more than 30 AGVs are installed.

(3) When the number of AGVs reaches 42, the average blocking rate of the system solved by method1 is significantly higher than that of the other two control methods. Compared with the previous work method [31] and method3 in this paper, the difference of average waiting time of method2 and method3 is 14.3s, the difference of the average time to complete the task is 14.5 s, which shows that the method in this paper takes less time than the previous method. When the number of AGVs is from 36 to 42, the average blockage rate of method2 increases by 0.22%, and that of method3 increases by 0.13%. These results show that when the number of AGVs increases, the method3 can reduce the average blocking rate of the system, complete the task more quickly, and is more suitable for solving the conflict-free path planning problem of more than 40 AGVs.

(4) In GAP1 and GAP2, the average blocking rate of method1 and method2 shows an upward trend, which is compared with the results of method3. It shows that with the increase of AGV number, method3 can greatly improve system efficiency.

TABLE 4. Experimental data sheet of different AGVs under different control methods.

Example	Number of AGVs	Method1 AWT	Method1 ABR	Method1 ACTT	Method2 AWT	Method2 ABR	Method2 ACTT	Method3 AWT	Method3 ABR	Method3 ACTT	GAP1	GAP2
1	20	9.7	0.49%	1984.6	6.5	0.32%	1981.4	4.9	0.25%	1979.7	96%	28%
2	25	14.5	0.59%	2515.1	9.1	0.36%	2509.6	7.3	0.29%	2507.8	103%	24%
3	30	29.0	0.97%	3014.9	17.2	0.58%	3004.1	13.7	0.46%	2999.6	110%	26%
4	33	36.7	1.09%	3407.7	23.1	0.68%	3394.0	17.2	0.51%	3388.7	113%	33%
5	36	51.4	1.38%	3762.3	32.1	0.87%	3740.9	23.2	0.62%	3732.0	122%	40%
6	42	72.8	1.74%	4250.5	45.5	1.09%	4223.2	31.2	0.75%	4208.7	132%	45%

AWT: Average waiting time(s), ABR: Average blocking rate, ACTT: Average completion time(s)

$GAP_1 = [ABR(n)' - ABR(n)] / ABR(n) * 100\% (n=1,2,...,6)$, $GAP_2 = [ABR(n)'' - ABR(n)] / ABR(n) * 100\% (n=1,2,...,6)$

(5) From the change range of the three curves in Fig. 8, when the number of AGVs increases from 25 to 30, the average blockage rate of AGVs has a large increase, of which the maximum increase of the average blockage rate of method1 is 0.38%, the second is method2, which is 0.22%, and that of method3 is 0.17%. It shows that under the current scale map, the number of conflicts of 30 AGVs is significantly increased compared with that of 25 and 20 AGVs, and the average waiting time of method1 is higher than that of the other two methods, so the increase is large. When the number of AGVs increased from 33 to 36, the three curves increased significantly for the second time, which were 0.29%, 0.19%, and 0.11%, respectively. The results show that the secondary conflict may occur in 36 AGVs, which increases the blocking rate of the system. Although the average blockage rate of method2 is lower than that of method1, the average blockage rate solved by method3 is smaller. It is proved that this method can reduce the average blockage rate caused by secondary conflict when solving 36 or more AGVs, which is better than method1 and method2.

C. THE TIME COMPARISON OF 42 AGVS UNDER THE DIFFERENT INTERACTION PROTOCOLS

Fig. 10 shows the running time comparison of 100 experiments of 42 AGVs under the interaction protocol of the improved blackboard model and the blackboard model. The black solid line represents the time value of 100 experiments under the interaction protocol of the improved blackboard model, and the violet dotted line represents the time value of 100 experiments under the interaction protocol of the blackboard model.

It can be seen that on the whole, the distribution of solid lines is lower than that of dotted lines. The running time under the improved interactive protocol is 137s, the running time of the unimproved interactive protocol is 169s, and the average running time is 32s lower. Because the interaction protocol of the blackboard model needs AGV to communicate continuously during the running process, the improved blackboard model interaction protocol only needs to find out the AGV that may conflict, while the AGV without conflict does not need to interact.

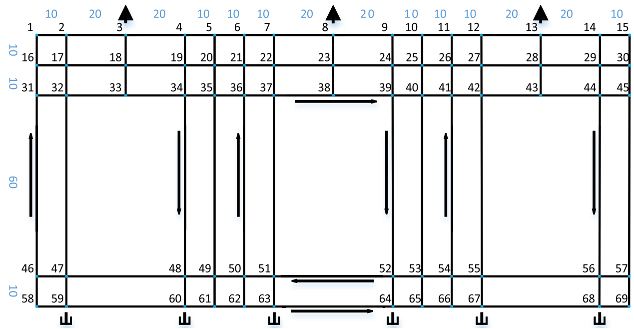


FIGURE 11. Smaller map.

D. COMPARATIVE EXPERIMENT ON AVERAGE BLOCKING RATE OF 36 AGVS UNDER DIFFERENT SCALE MAPS

The description of nodes, arrows, and path length in Fig. 11 is consistent with that in Fig. 6. The difference between Fig. 11 and Fig. 6 is that the number of nodes in Fig. 11 is 69, nodes 3, 8, 13 are the quay crane position, and nodes 59, 60, 63, 64, 67, 68 are the yard crane positions.

In this experiment, the maximum number of AGVs that can be accommodated in the 65-node map(map2) is 36. Similarly, the capacity of the 75-node map(map1) in Experiment A is also 36. A comparative experiment of the average blockage rate under the three methods of two-scale maps is designed. A small-scale AGV operation path network is established as shown in Fig. 11.

The average blocking rate of AGVs on different scale maps is shown in Fig. 12. (1) In Fig. 12, under the three control methods, the experimental results under 30 AGVs in map2 show that the average blocking rate is higher than that in map1. In Table 5, the difference of average blocking rate of three control methods in different scale maps is 0.01%-0.02%, which indicates that the number of conflicts between 20-25 AGVs in map2 is the same as that in map1, and the difference is reflected in the number of nodes increases, the completion time of AGV increases, so the average blocking rate of AGVs in map1 is lower.

(2) When the number of AGVs is 25-36 AGVs, in Fig. 12, it can be seen that the average blocking rate of the two kinds

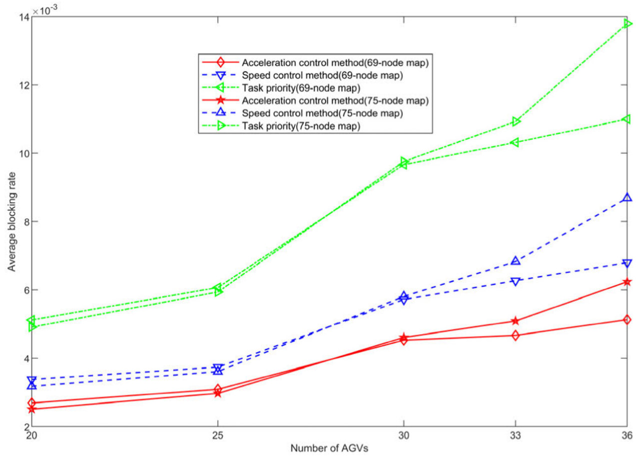


FIGURE 12. Average blocking rate of different scale maps.

TABLE 5. Average blocking rate of AGVs with different scale maps.

Method(map)	Number of AGVs				
	20	25	30	33	36
Method1 (map1)	0.491%	0.594%	0.976%	1.093%	1.38%
Method2 (map1)	0.318%	0.361%	0.581%	0.683%	0.869%
Method3 (map1)	0.251%	0.297%	0.461%	0.509%	0.624%
Method1 (map2)	0.512%	0.607%	0.967%	1.032%	1.1%
Method2 (map2)	0.338%	0.374%	0.572%	0.627%	0.68%
Method3 (map2)	0.269%	0.309%	0.453%	0.467%	0.513%

of maps first almost coincides, and then when the number of AGVs exceeds 30, the average blocking rate in map1 exceeds that in map2, and increases with the increase of AGV number.

(3) When the number of AGVs is 36, the experimental results of solving the average blockage rate of different maps under the three control methods are different in Fig. 12. This difference is obviously caused by the increase of map scale and the possibility of more conflict nodes, which leads to an increase in the blocking rate. The average blocking rate of map1 solved by method2 and method1 increased to 0.19% and 0.28%. However, the average blocking rate of map1 solved by method3 based on MAS only increases by 0.011%, which is 2 times less than that under method1 and is close to 1/2 of the increased amplitude of average blockage rate under method2. Obviously, in the face of a larger map, compared with method2, method3 can solve the conflict situation in the conflict node more quickly and release the node as soon as possible.

(4) The average blocking rate of each AGV in different scale maps is shown in Table 6 below. It can be seen that the BOLD data in the table correspond to the minimum value of the average blocking rate of each AGV in different scale maps. At the same time, the second column where the two minimum values are located, and each data in this column is also the minimum value of all data in its row. Therefore, for two different maps in this paper, placing 25 AGVs can reduce the number of conflicts in the system.

TABLE 6. Average blocking rate of each AGV with different scale maps.

Method(map)	Number of AGVs				
	20	25	30	33	36
Method1 (map1)	0.025%	0.024%	0.033%	0.033%	0.038%
Method2 (map1)	0.016%	0.014%	0.019%	0.021%	0.024%
Method3 (map1)	0.013%	**0.012%**	0.015%	0.015%	0.017%
Method1 (map2)	0.026%	0.024%	0.032%	0.031%	0.031%
Method2 (map2)	0.017%	0.015%	0.019%	0.019%	0.019%
Method3 (map2)	0.013%	**0.012%**	0.015%	0.014%	0.014%

V. CONCLUSION

In this paper, the minimum blocking rate of AGV as the objective was taken, and the travel speed, running time, and collision distance of AGV were considered, a conflict-free path planning model for multiple AGVs was established. At the same time, the AGV system based on MAS was designed, and the acceleration control method was used as the negotiation strategy of AGV. Four groups of comparative experiments had been designed to compare the average blocking rate, waiting time and completion time of different AGVs, the running time under different interactive protocols, and the average blocking rate of different scale maps.

Experiment 1 shows that the acceleration control method based on MAS can solve the conflict faster and reduce the possibility of secondary conflict. In Experiment 2, for different numbers of AGVs, when the number of AGVs is less than 30, the average blockage rate of the acceleration control method based on MAS is slightly lower than the other two control methods. When there are 30-42 AGVs, the average blockage rate of the system is significantly increased. Especially for 42 AGVs, the average blockage rate of the acceleration control method based on MAS is 0.75%, and the average blockage rate from 36 to 42 AGVs is 0.13%, which is lower than that of the speed control method based on MAS. The results show that the acceleration control method based on MAS is more suitable for solving more than 40 AGVs conflict-free path planning problems. Experiment 3 shows that the improved blackboard interaction protocol can reduce the computing time by 32s. In Experiment 4, when the map scale increases, the number of collision nodes would also increase. However, the acceleration control method based on MAS can solve the conflict problem of conflict nodes more efficiently and reduce the impact on other AGVs on the path. The experimental results verify the feasibility of this model, and the acceleration control method based on MAS was more suitable for solving the conflict-free path planning problem of more than 40 AGVs.

The future work will focus on the establishment of a simulation platform for multi-AGV route planning in automated terminals. The simulation platform can be used to do more experiments with verifying the effectiveness of the method3 in this paper.

REFERENCES

[1] X. Lyu, Y. Song, C. He, Q. Lei, and W. Guo, "Approach to integrated scheduling problems considering optimal number of automated guided vehicles and conflict-free routing in flexible manufacturing systems," *IEEE Access*, vol. 7, pp. 74909–74924, 2019.

[2] Z. Zhang, Q. Guo, J. Chen, and P. Yuan, "Collision-free route planning for multiple AGVs in an automated warehouse based on collision classification," *IEEE Access*, vol. 6, pp. 26022–26035, 2018.

[3] X. H. Cao and M. Zhu, "Research on Multi-AGV conflict avoidance decision optimization method based on conflict prediction," *Comput. Integr. Manuf. Syst.*, vol. 8, no. 51, pp. 1–11, 2019.

[4] S. Y. Zhang, Y. S. Yang, C. J. Liang, B. W. Xu, and J. J. Li, "Optimal control of multiple AGV path conflict in automated terminals," *J. Transp. Syst. Eng. Inf. Technol.*, vol. 17, no. 2, pp. 83–89, 2017.

[5] M. S. Zhong, Y. S. Yang, and Y. M. Zhou, "Free-conflict AGV path planning in automated terminals based on speed control," *Comput. Sci.*, vol. 47, no. 7, pp. 308–314, 2019.

[6] D. K. Liu, X. Wu, A. Kulatunga, and G. Dissanayake, "Motion coordination of multiple autonomous vehicles in dynamic and strictly constrained environments," in *Proc. IEEE Conf. Cybern. Intell. Syst.*, Jun. 2006, pp. 1–6.

[7] M. Zhong, Y. Yang, Y. Dessouky, and O. Postolache, "Multi-AGV scheduling for conflict-free path planning in automated container terminals," *Comput. Ind. Eng.*, vol. 142, Apr. 2020, Art. no. 106371.

[8] J.-J. Li, B.-W. Xu, O. Postolache, Y.-S. Yang, and H.-F. Wu, "Impact analysis of travel time uncertainty on AGV catch-up conflict and the associated dynamic adjustment," *Math. Problems Eng.*, vol. 2018, pp. 1–11, Jan. 2018.

[9] I. K. Singgih, S. Hong, and K. H. Kim, "Flow path design for automated transport systems in container terminals considering traffic congestion," *Ind. Eng. Manage. Syst.*, vol. 15, no. 1, pp. 19–31, Mar. 2016.

[10] P. Legato, R. M. Mazza, and R. Trunfio, "Simulation-based optimization for discharge/loading operations at a maritime container terminal," *OR Spectr.*, vol. 32, no. 3, pp. 543–567, Jul. 2010.

[11] Y. Yang, M. Zhong, Y. Dessouky, and O. Postolache, "An integrated scheduling method for AGV routing in automated container terminals," *Comput. Ind. Eng.*, vol. 126, pp. 482–493, Dec. 2018.

[12] J. Xin, R. R. Negenborn, F. Corman, and G. Lodewijks, "Control of interacting machines in automated container terminals using a sequential planning approach for collision avoidance," *Transp. Res. C, Emerg. Technol.*, vol. 60, pp. 377–396, Nov. 2015.

[13] K. Matsubyshi, "A collaboration mechanism on positive interactions in multi-agent environments," in *Proc. 13th Int. Joint Conf. Artif. Intell. (IJCAI)*, 1993, pp. 352–358.

[14] G. Cornuejols, G. Nembouser, and L. Wolsey, *The Uncapacited Facility Location Problem*. Pittsburgh, PA, USA: MSRR, Carnegie-Mellon Univ., America, 2012.

[15] Z. Obdrzalek, "Mobile agents and their use in a group of cooperating autonomous robots," in *Proc. 22nd Int. Conf. Methods Models Autom. Robot. (MMAR)*, Aug. 2017, pp. 125–130.

[16] D. Herrero-Perez and H. Martinez-Barbera, "Modeling distributed transportation systems composed of flexible automated guided vehicles in flexible manufacturing systems," *IEEE Trans. Ind. Informat.*, vol. 6, no. 2, pp. 166–180, May 2010.

[17] L. B. Branisso, E. R. R. Kato, E. C. Pedrino, O. Morandin, and R. H. Tsunaki, "A multi-agent system using fuzzy logic to increase AGV fleet performance in warehouses," in *Proc. III Brazilian Symp. Comput. Syst. Eng.*, Nov. 2013, pp. 137–142.

[18] G. Villarrubia, J. De Paz, J. Bajo, and J. Corchado, "Ambient agents: Embedded agents for remote control and monitoring using the PANGEA platform," *Sensors*, vol. 14, no. 8, pp. 13955–13979, Jul. 2014.

[19] C. Zato, G. Villarrubia, A. Sánchez, I. Barri, E. Rubión, A. Fernández, C. Rebate, J. A. Cabo, T. Álamos, and J. Sanz, "PANGEA-platform for automatic construction of organizations of intelligent agents," in *Distributed Computing and Artificial Intelligence*. Berlin, Germany: Springer, 2012, pp. 229–239.

[20] M. Frego, P. Bevilacqua, S. Divan, F. Zenatti, L. Palopoli, F. Biral, and D. Fontanelli, "Minimum time—Minimum jerk optimal traffic management for AGVs," *IEEE Robot. Autom. Lett.*, vol. 5, no. 4, pp. 5307–5314, Oct. 2020.

[21] H. V. D. Koen, V. L. Zofia, J. A. Ottjes, and L. Gabriel, "Distributed intelligence in autonomous multi-vehicle systems," *Int. J. Crit. Infrastruct.*, vol. 2, pp. 261–272, Jan. 2006.

[22] R. Erol, C. Sahin, A. Baykasoglu, and V. Kaplanoglu, "A multi-agent based approach to dynamic scheduling of machines and automated guided vehicles in manufacturing systems," *Appl. Soft Comput.*, vol. 12, no. 6, pp. 1720–1732, Jun. 2012.

[23] Z. X. Cai, B. F. Chen, L. Wang, L. J. Liu, and Z. H. Duan, "The progress of cooperative technology for heterogeneous multiple mobile robots," *CAAI Trans. Intell. Syst.*, vol. 2, no. 3, pp. 1–7, 2007.

[24] J. F. Jing, "Study and development of autonomous control multi-AGV system based on agent," M.S. thesis, Dept. Electron. Eng., Nanjing Univ. Aeronaut. Astronaut., Nanjing, China, 2014.

[25] Z. Rong, "Information system of emergency materials management based on active blackboard structure design," in *Proc. 5th Int. Conf. Intell. Syst. Design Eng. Appl.*, Jun. 2014, pp. 570–573.

[26] B. Micieta, M. Edl, M. Krajcovic, L. Dulina, P. Bubenik, L. Durica, and V. Binasova, "Delegate MASs for coordination and control of one-directional AGV systems: A proof-of-concept," *Int. J. Adv. Manuf. Technol.*, vol. 94, nos. 1–4, pp. 415–431, Jan. 2018.

[27] A. Okoso, K. Otaki, and T. Nishi, "Multi-agent path finding with priority for cooperative automated valet parking," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 2135–2140.

[28] M. Cap, P. Novak, A. Kleiner, and M. Selecky, "Prioritized planning algorithms for trajectory coordination of multiple mobile robots," *IEEE Trans. Autom. Sci. Eng.*, vol. 12, no. 3, pp. 835–849, Jul. 2015.

[29] L. Henesey, P. Davidsson, and J. A. Persson, "Evaluation of automated guided vehicle systems for container terminals using multi agent based simulation," in *Proc. Multi-Agent-Based Simul. IX*, vol. 5269. Berlin, Germany: Springer, 2008, pp. 85–96.

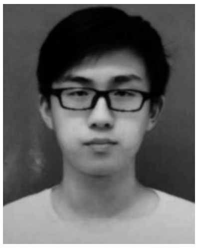
[30] L. Henesey, "Multi-agent container terminal management," Ph.D. dissertation, Blekinge Inst. Technol., Karlskrona, Sweden, 2006.

[31] K. Guo and J. Zhu, "Multi-agent AGV conflict free path planning based on improved speed control method in automated terminals," *Int. Core J. Eng.*, vol. 6, no. 9, pp. 386–398, Sep. 2020.

KUNLUN GUO received the B.E. degree in automation from Jiangsu Normal University, Xuzhou, China, in 2018. He is currently pursuing the M.S. degree under the supervision of Dr. J. Zhu in control theory and control engineering with Shanghai Maritime University, Shanghai, China. His research interest includes conflict-free path planning for large-scale AGVs in automated terminals.

JIN ZHU received the B.E. degree from the Huazhong University of Science and Technology, in 2002, the M.S. degree from Huaqiao University, in 2005, and the Ph.D. degree from the East China University of Science and Technology, in 2008.

Since 2008, she has been an Associate Professor with the Institute of Logistics Science and Engineering, Shanghai Maritime University, Shanghai. She has authored over 40 articles and holds four invention patents. Her work has focused on large-scale optimization, decision-making under uncertainty, and computational optimization, including Markov decision processes and stochastic programming, and coordinated control of autonomous systems. Her awards and honors include the Shanghai Youth Technology Rising-Star2010 and the shanghai Science and Technology Progress Award in 2011.



LEI SHEN received the B.E. degree in automation from the Jinling Institute of Technology, Nanjing, China, in 2017. He is currently pursuing the M.S. degree under the supervision of Dr. J. Zhu in control theory and control engineering with Shanghai Maritime University, Shanghai, China. His research interest includes yard cranes scheduling problems in automated terminals and meta-heuristics.