Proceeding of the 2015 IEEE
International Conference on Information and Automation
Lijiang, China, August 2015

Path Planning of Automated Guided Vehicles Based on Improved A-Star Algorithm *

Chunbao Wang, Lin Wang, and Jian Qin
The First Affliated Hospital
Sun Yat-sen University
No. 135, Xingang Xi Road, Guangzhou, 510275, P. R. China
wangchb6@mail.sysu.edu.cn

Xia Su, Weiguang Li, Zhijiang Lu, and Mengjie Li
School of Mechanical and Automotive Engineering
South China University of Technology
Wushan RD.,Tianhe District,Guangzhou,P.R.China,510641
Wguangli@scut.edu.cn

Zhengzhi Wu, Lihong Duan, Zhongqiu Li Mequn Cao and Xicui Ou
Shenzhen Institute of Geriatrics
No. 3002, Sungang Xi Road, Shenzhen, 518037, P.R China
Maria.duan@gmail.com

Yulong Wang, Jianjun Long, Meiling Huang
Yinghong Li and Qiuhong Wang
Shenzhen Second People's Hospital
No. 3002, Sungang Xi Road, Shenzhen, 518037, P.R China
mrwang@fuji.waseda.jp

***Abstract*—With the development of automated logistics systems, flexible manufacture systems (FMS) and unmanned automated factories, the application of automated guided vehicle (AGV) gradually become more important to improve production efficiency and logistics automatism for enterprises. The development of the AGV systems play an important role in reducing labor cost, improving working conditions, unifying information flow and logistics. Path planning has been a key issue in AGV control system. In this paper, two key problems, shortest time path planning and collision in multi AGV have been solved. An improved A-Star (A*) algorithm is proposed, which introduces factors of turning, and edge removal based on the improved A* algorithm is adopted to solve k shortest path problem. Meanwhile, a dynamic path planning method based on A* algorithm which searches effectively the shortest-time path and avoids collision has been presented. Finally, simulation and experiment have been conducted to prove the feasibility of the algorithm.**

Keywords—AGV; path planning; A* algorithm; improved A * algorithm

I. INTRODUCTION

Logistics system consisting of AGV has become increasingly popular as industry develops. However, there are still has some problems to be solved in application. As one of the most serious problems, AGV path planning is different from the general vehicle routing problem. It applied for different applications and different tact. What is more, the constraints, the optimal solution and strategies are different from general vehicle routing.

Path planning means that to search for an optimal feasible path in certain environmental constraints, according to the given starting and ending point, referring to one or some factors, such as the shortest path, the shortest time, least cost and other factors. There are static routing, and dynamic path planning in AGV path planning [1]. Static path planning means that the route has been identified in advance between sites. Path planning such as moving from site i to site j, the AGV just need to run along a fixed path. Static path planning is simple, but cannot adapt to changes in the environment and transport situation. Dynamic programming could avoid conflicts between multi AGV based on the current environment and real-time traffic information, to search optimal path planning for each AGV. This is a difficulty in AGV path planning.

Resource competition and conflict need to be solved in multi-path planning for AGV systems. Researchers have conducted many studies, different application and put forward many solutions strategy. LING QIU, WEN JING [2], take the lead to describe AGV path planning and scheduling issues in detail and summarize the method of AGV path planning. Broadbent etc. [3], consider conflict and the optimal time for the purpose, and the priority method using Dijkstra algorithm to solve a one-way route network planning issues. Kim and Tanchoco [4] give an algorithm combining Dijkstra algorithm and time window on bidirectional path network. This algorithm convert path planning problem into a plan node idle time window problem. However, the process of the algorithm is complex and with low search efficiency. In order to improve search efficiency of the algorithm, Taghaboni-Dutta and Tanchoco [5], etc. proposed an incremental planning approach. But this approach sacrifices the optimality path. Nenad Smolic-Rocak et al. [6] discussed a number of AGV dynamic path planning time window based on the algorithm edges time planning regulations. This algorithm allow only one AGV occupy one side, therefore the utilization the path is relatively low. In China, Zhifan Luo et al. [7] proposed a two-stage path planning method combining fuzzy control global path planning and local planning, when the AGV running path reduction while achieving the car run without obstacles. Guodong Liu et al. [8] [9] have proposed a two-stage path planning method. The method first generation AGV exercised within the site offline candidate path set, then the path candidate centralized online options exercised AGV path. However, these two methods cannot guarantee the optimal solution search path. Other scholars have research planning algorithm based on path time window [10] [11]. They combine the traditional path planning method with the time window to planning AGV path. There are also some researchers combine the time windows

This work is partially supported by Nansha Scientific Research Project #2014CX07 ; Project on the Integration of Industry, Education and Research of Guangdong Province # 2012B091100311; Clinical Doctor-Basic Scientist Combination Foundation of Shenzhen Second People's Hospital. 2015 public interest research and capacity building projects, Science and Technology Department of Guangdong, #2014A020221004 and #2014A020215004;Natinal Foundation, National High Technology Research and Development Program of China (863 Program) #2014AA020907

978-1-4673-9104-7/15/$31.00 ©2015 IEEE

method with Petri net to resolve more AGV path planning [12] [13]. To some extent, these methods are expanded the search to achieve more complicated.

In this paper, we will introduce the path planning algorithm, A* algorithm of shortest time planning and multi AGV path planning which is used in the designed AGV system (Figure 1). In the shortest time planning, factors of turning, and edge removal based on the improved A* algorithm is adopted to solve k shortest path problem. In multi AGV path planning, a dynamic path planning method based on A* algorithm which searches effectively the shortest-time path and avoids collision has been presented.

Figure 1. Overview of the AGV-MKS_1

II. PROBLEM DESCRIPTION

A. *Shortest time planning*

During the AGV path planning, an issue needed to be addressed is to choose reasonable path. Many scholars equaled the shortest path to the most reasonable one, while some researchers considered the smoothness of the path, such as the influence of AGV turning [13]. But both of them did not give a specific approach. The most reasonable AGV path should be short, smooth and so on.

The network diagram of production plant is showed in Fig.2. This paper will simplify the path among loading stations, unloading stations, processing stations to undirected connected graph G. The shortest path of AGV23 from material shipped station 23 to processing station 10 can be path 23-24-14-15-16-17-18-19-9-10 or path 23-24-14-7-8-9-10. But the latter one is more tortuous, with more turning number, which is not conducive to AGV automatic operation control. The traditional algorithms of solving the problem of shortest path cannot select the path with less turning number from the two paths above in this situation.

When the shortest path is being used by another AGV or obstructions, AGV needs to choose the second shortest or third shortest path. Currently, although there are many solutions to seek k times shorter path [14] [15], but they are more complex.

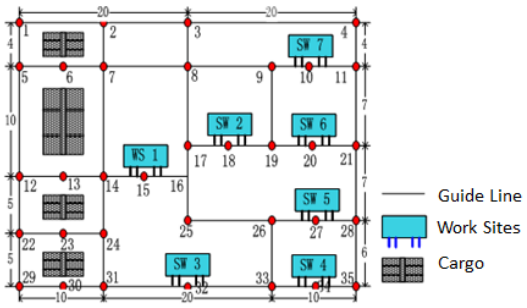


Fig.2 Environment map

B. *Multi AGV path planning*

AS AGV system environment said based on graph theory, the path among work stations, intersections, turning points can be reduced to undirected connected graph G = {V, E, w}, where V is the set of nodes, E is set of edges formed by the node connection, w represents the effective length between two nodes.

(1) AGV running path is the single-channel and two-way one, and the path width can only accommodate one AGV;

(2) AGV running speed is constant and same;

(3) At a time, only one AGV is allowed in the range containing each node, but a number of vehicles are allowed on one AGV path;

(4) According to the rules of the actual vehicle exercise, a safe distance is settled among AGV. if the distance between the two AGV is less than the safety distance, the rear of the car is stopped, until no obstacle in the safety distance is detected.

III. SHORTEST TIME PLANNING

A. *Principle of A* algorithm*

As shown in AGV working environment graph G, di represents the shortest path from node vi to the start site vs, dij represents the shortest path from node vi to node vj. We can draw the following conclusions: (1) the shortest distance between node vi and node vj in graph G is the one between vi and vj ,which located in the shortest path from starting station vs to objective station vt. (2) the second shortest path must go through the nearby node of shortest path. The second shortest path is one of from starting node to terminal point through nearby node of shortest path .

The conclusions above were demonstrated in [14].This article uses the A * algorithm for the shortest path, firstly we must rule out the shortest path, so detecting one side of the shortest path can rule out the shortest path L from start site vs to target site vt. and then using the A * algorithm can get the shortest path L. According to the conclusion (1), (2), one of the shortest paths must be the desired second shortest path by successively delete the sides of shortest paths from start site vs to target site. Seek out all the shortest path length in ascending order, select the former k paths ask times shorter path and store them in path library of AGV system.

B. *Shortest path in k times*

The process of finding K shortest path based on the method above is shown as follows:

1) In this paper, the improved A * algorithm is used to find the shortest path from the start site vs to target site vt. and stored it in the array path [n]. n is the number of edges of the shortest path;

2)The definition of integer i, and i is initialized to 1. it is determined whether or not less than and equal to n, turn 3) if condition is satisfied, otherwise, go to 4);

3) Removing side <path [i-1], path [i]> proves that there is no connection between them. Then call again A * algorithm to find the shortest path from the start site vs to target site the target site vt, store the result in pathi []. i is equal to i pulsing 1,then go to 2);

4) If n is less than k, the data of pathi [] (i = 1 ... n) are sorted in descending order along with storing the path and related length in the database of AGV system. If n greater than or equal The AGV system path in the database path and the corresponding path length is stored; if n is greater than of equal to k, select the former k-1 shortest path of pathi [k-1] based on path length and store the corresponding length with path[n] in database of AGV system.

C. *Simulation*

Simulation map imitate general production automation workshop scene, shown in Fig.1. Using VC ++ 6.0 program, set the value of r. According to practical research applications, r is respectively set to 0.8,0.4,1. Compared with general A * algorithm and Dijstra algorithm, the path length , number of turns and search nodes of this improved A * algorithm (1, 2, 5) are shown in Table 1, the experimental path is shown in Fig.3.

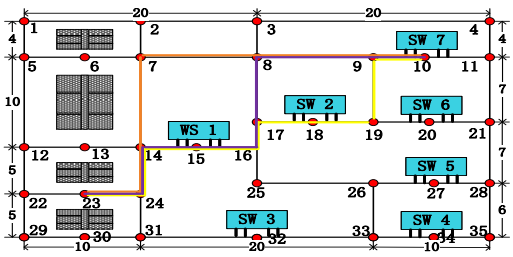


Fig.3 Algorithm simulation results

TABLE 1: THE SIMULATION COMPARISON

Algorithm		Path (color)	Path length	The number of turning	Search node number
Dijkstra		Yellow	45	6	32
A* Algorithm		Yellow	45	6	17
Improved A* Algorithm	$\alpha=1$	Orange	45	2	15
	$\alpha=2$	Orange	45	2	14
	$\alpha=5$	Orange	45	2	16

As can be seen, the path length of three methods above is same, but the number of turning points are different. In this paper, the improved A * path algorithm can search the shortest path with least number of turns of the AGV control and the track was better than the first two. At the same time, we can see that the search efficiency of A * algorithm is improved greatly relative to Dijkstra algorithm and slightly improved relative to general A * algorithm. When s is close to the minimum length of the line, the search efficiency reaches to the lowest.

Respectively using the algorithm of this paper to seeking k times shortest path and results of neighboring point algorithm to seeking former 3 times shortest path of document 8 shown in Table 2, the experimental path is shown in Fig2.

TABLE 2 THE RESULT OF THREE SHORT PATHS

Algorithm	Shortest	The second shortest	The third shortest
Method in this paper	Orange	Purple	Yellow
Nearby Point Method	Orange	Purple	Yellow

The paths of two algorithms are same. The enumeration method of near-point method has complex algorithm and low efficiency. But the improved A * algorithm, the algorithm is simple and the search efficiency is fast.

IV. MULTI AGV PATH PLANNING

A. *Path-time model*

Aiming at shortest running time without conflict, AGV path planning is different from the shortest path planning, which is aiming at the shortest path. Considering there already has a lot of sophisticated algorithms, shortest path problem can be solved based on existing algorithms. Firstly, we should convert path model into a path-time model. Specific methods are as follows.

Parameter can be set at each node. One-dimensional AGV registration information in the node set R. There are three elements {stime, ltime, no} in structure, to represent the arrives at the node of the time, departure time node, AGV identity number respectively. Registration information point j in node i can be represented by $R_j^i.\mathrm{no}$, $R_j^i.stime$, $R_j^i.ltime$.

When a node assigned to one AGV, AGV registration information on the corresponding node indicates that the node is unavailable in this period. When AGV leave node, the corresponding registration information, and node release time are deleted. Other AGV can pass the node, AGV path planning is equivalent to planning the shortest time line from the starting node to the target point on the existing time information in the available period.

B. *A * algorithm in time planning*

A * algorithm is a heuristic search algorithm, which uses heuristic function to assess the cost of transit node to the destination node. A * algorithm evaluation function can be expressed as:

$$f^*(x) = g^*(x) + h^*(x) \quad (1)$$

Wherein, g (x) represents the shortest path from starting point s to the current node value of x. h * (x) indicates heuristic value of the shortest path from current node x to the target node. Since h * (x) cannot be known in advance, f (x) is the approximation value of f * (x) , f (x) is expressed as follows:

$$f(x) = g(x) + h(x) \quad (2)$$

Wherein $g(x) \geq g^*(x)$, and $h(x) \leq h^*(x)$.

$g(x)$ and $h(x)$ make the following changes: $g(x)$ indicates AGV entering the earliest time value of the current node x, $h(x)$ representing the current value of the target node x to inspire node time t, is expressed as follows:

$$h(x) = \frac{\sqrt{(v_x.x - v_t.x)^2 + (v_x.y - v_t.y)^2}}{v} \quad (3)$$

Wherein v represents the average running speed of AGV.

Supposing there is no potential conflict form node I to node x, the solution method is as follows:

Times when AGV first access x is $L_{ix} = g(i) + t_{ix}$, t_{ix} shows time AGV pass edge <i, x>, compare $R_1^x.stime$ and the value of L_{ix} , if $L_{ix} + t_x < R_1^x$, t_x represents AGV occupied node x times, then $g(x) = L_{ix}$, otherwise the search R^x , order $T_{arr} = \max\{R_j^x.ltime, L_{ix}\}$, find the T_{arr} values that satisfy $R_{j+1}^x.stime - T_{arr} > t_x$ to obtain $g(x) = T_{arr}$.

When there is existing conflict from node i to node x, $h(x)$ can be find by the following strategy.

C. Collision detection

Issues of AGV node resource contention have been solved by allocation of time, but AGV edge resource contention problems still exist. AGV edge resource contention problems perform on the facing conflict and catching up with. When all AGV are running at a same speed, conflict of catching up with will not occurred. Therefore, there is no need to considerate this kind of conflict. Facing conflict is shown in Fig.4. The time axis of node vi and node vx is shown in Fig.5. The routine of AGV1 has been planned, while the routine of AGV2 has not. The method to tell if there is facing conflict in AGV2 from node vi to node vj are as follows:

Checking in the registration information $R_j^i.stime > g(i)$ in R^i and $R_k^x.stime < g(x)$ in R^x , to make sure if there is the same registration information whether AGV identification number, if any, by the presence of opposing conflict i to x.

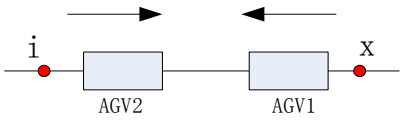


Fig.4 Facing Conflict

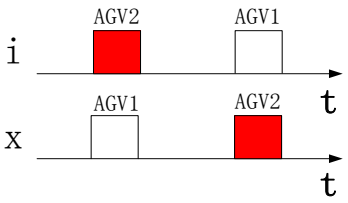
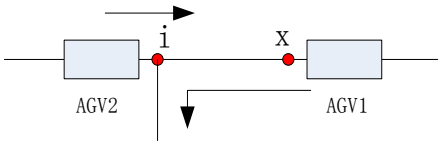
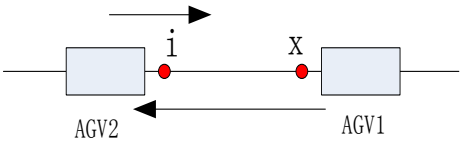


Fig.5. Time Axis of Node i and x

There two conflict exists, the first one is waiting to avoid conflicts, as shown in Fig.6 (a). The second one is the unavoidable conflict, as shown in Fig.6 (b). The first conflict can make AGV2 edge node i wait until the AGV leaving node i to avoid this strategy; the second conflict is inevitable, so for AGV2, the edge <i, x> is unfeasible.



(a) Conflict can be avoided by waiting



(b)Unavoidable conflict

Fig.6 The Type of Conflict

$g(x)$ solution method about these two conflicts are as follows:

1) Waiting to avoid conflict. Searching R^x backwards, to find AGV arrives at a node i make edge <i, x> , and then calculate the time from the first node i to node x earliest arrival time $g(x)$.

2) Unavoidable conflict: $g(x)$ to infinity.

D. Multi AGV path planning algorithm for conflict-free process

O represents to be extended to define a list of nodes, C indicates expansion node list, s, t, respectively, originating node and the destination node. Multi AGV path planning algorithm conflict as follows:

1) Initialization list C is an empty set, O = {s}, the car recorded the start time for the starttime, $g(x)$ initialized to

starttime, and then calculate value of h(s) and f(s) by the formula;

2) Determining whether set O is empty, if yes, go to 10), else go to 3);

3) Find the smallest f value in the node from O, denoted i, remove it from the O, insert or C;

4) Determining whether node i is equal to the target node t, if yes, go to 9), else go to 5);

5) For each node connected to node i and j, determining whether a conflict exists from i to j, if present, determining what kind is the conflict. If the conflict can be avoided by waiting, go to 6). Else set g(s) to infinity, which means that the path is being planned for the AGV, the edge <i, x> is unfeasible; if there is no conflict, go to 6);

6) Determining whether the node j in O, if yes go to 7), else go to 8);

7) The method of calculating the current mentioned above g (j) values, determines the current g (j) value is smaller than the previous value of g (j), and if so, put the node i as the parent node of node j to the new node j f (j), g (j), the value of h (j) replaces the old value, go to 2);

8) O was added to the node j, and node i to node j as a parent node, calculating f (j), g (j), the value of h (j), and go to 2);

9) Finding the path, the path to the output s - t, and the arrival time of each node on the path stime, namely, the departure time ltime each node is inserted into the registration information on each node {stime, ltime, AGVno};

10) Did not find the path from t to s.

E. Simulation

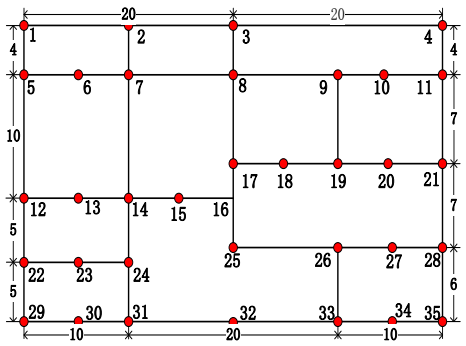


Fig.7 Environment of AGV System

Fig.7 shows a network diagram of a company production workshop. Assuming there are three existing task. Task 1 is that move AGV1 form site 13 to site 9. Set current time 0s as start time. Task 2 is that move AGV2 form site 34 to site 13. Set the start time for 3s; Task 3 is that move AGV3 form site 18 to site 23. Set the start time for 24s. With the above algorithm, seek AGV1 and AGV2 run path and registration information, the result are as shown in Table 3.

TABLE 3: PATH PLANNING OF AGV1 AND AGV2

AGV1			AGV2		
Node	**stime**	**ltime**	**Node**	**stime**	**ltime**
13	0	2	34	3	5
14	10	14	33	13	17
15	20	24	32	33	37
16	30	34	31	53	57
17	36	40	24	63	67
8	50	54	14	73	77
9	70	74	13	83	87

TAB.4 PATH PLANNING OF AGV3

F(x)min				O list
Node	**g**	**f**	**Parent node**	
18	24	67	-	18
17	34	68	18	17,19
16	46	78	17	19,8,16
15	56	78	16	19,8,15,25
14	66	80	15	19,8,25,14
25	54	84	16	19,8,25,7,13,24
13	76	86	14	19,8,7,13,24,27
24	77	87	14	19,8,25,7,24,12
23	87	87	24	19,8,25,7,12,23,31

TAB.5 DATA COMPARISON

AGV3 path information by our algorithm			AGV3 ideal path information		
Node	**stime**	**ltime**	**Node**	**stime**	**ltime**
18	24	26	18	24	26
17	34	38	17	34	38
16	46	50	16	40	44
15	56	60	15	50	54
14	66	70	14	60	64
24	77	81	24	70	74
23	87	91	23	80	84

Contrasting table 4 and table 5, it show that there is a potential conflict on the side of <17,16> in AGV3 and AGV1 and there is a potential conflict on the side of <24,14>with AGV2. And there is no conflict while using the above algorithm. As can be seen from table 5, A * algorithm has advantages of less searching nodes, high efficiency inherits. Meanwhile, on the node search strategy, it calculate the first reach time, instead of searching all the possible time period. Only when there is time to wait for the search to avoid conflicts or inferior early morning time, therefore reducing the search range, further to improve search efficiency.

Hardware configuration consists of PC and four CC2430 wireless modules. The AGV control system software is developed on VC ++ 6.0 using the above algorithm experimental verification. CC2430 acts as a connection between AGV and computer. Three CC2430 modules simulate AGV, AGV control system receives the instructions issued and given real-time upload speed (experimental speed 0.5m / s) at the location and status information.

In Task Manager, add the three tasks, system test results are shown in Figure 8.

Experiments show that the AGV path planning is the same as above section. We can also see that AGV3 from site 14 to site 17 waiting until they are sites run by other AGV, thus avoiding potential conflicts.

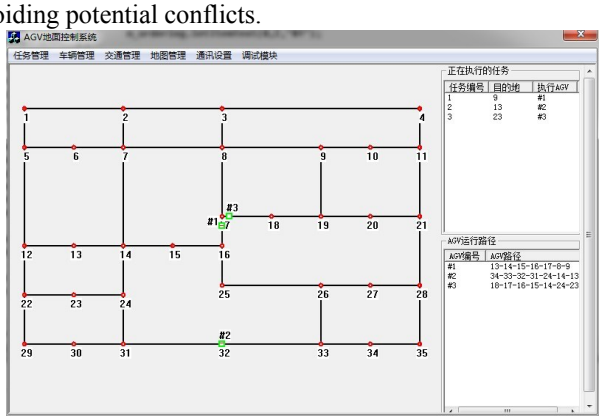


Fig.8. Simulated Experiment Verification

V. Conclusion and Future Work

In this paper, AGV turning factor is introduced to AGV path planning, to conduct an improved A * algorithm. The algorithm generates k times shorter path by deleting edges. The algorithm is simple and high search efficiency. The simulation results proved that the improved A * path planning algorithm is more smooth, more conducive to automatically run the AGV. The paper proves the feasibility of the proposed algorithm by comparing A* algorithm and other algorithm of deleting edge. Meanwhile, the paper presents a multi-AGV A * algorithm based on collision-free dynamic path planning method. The algorithm classified the potential conflicts to search shortest path with conflict-free effectively. The algorithm presented in this paper of great significance in the FMS application.

As a future work, the module of AGV actual running would be more specific. The algorithm also needs to continue to improve on case of different situation. As the project heading, algorithm of multi AGV run in a real situation should be developed deeply. Furthermore, the ground systems software needs to further improve ground control system interface display.

References

[1] Dongliu Qi. AGV path planning studies based on Intelligent Control. Master thesis. Hefei University of Industry ,2006

[2] LING QIU,WEN-JING HSU,SHELL-Y1NG HUANG .Scheduling and Routing Algorithms for AGVS:a survey .International Journal of Production Research,2002,3(40):744-760.

[3] Broadbent A J, Besant C B, Premi S K, et al. Free ranging AGV systems: Promises, problems and pathways. Proceedings of the 2nd International Conference on Automated Materials Handling. UK: IFS Publication Ltd，1985: 221-237

[4] KIM W C, TANCHOCO J M A.Conflict-free shortest-time bidirectional agv routeing. International Journal of Production Research, 1991, 29 (12): 2375-2391.

[5] TAGHABONI-DUTTA F, TANCHOCO J M A. Comparison of dynamic routeing techniques for automated guided vehicle system. International Journal of Production Research, 1995, 33(10): 2655-2669.

[6] Smolic-Rocak, Nenad,Bogdan, Stjepan,Kovacic, Zdenko,et al.Time windows based dynamic routing in multi-agv systems[J]. IEEE Transaction on Automation Science and Engineering, 2010, 7(1): 151-154.

[7] Zhifan Luo, Zuyao Lu, Qin Zhang.et al. Path Planning Method for a Class AGV [J]. Chinese Journal of Construction Machinery, 2004 (4): 400-403 + 407..

[8] Guodong Liu. Dynamic path planning multi AGV scheduling system in two stages [J] Robotics, 2005, (3): 210-215

[9] Jiancheng Jia. AGV Vision-guided and Path Planning Strategy. Master thesis. Yanshan University,2010.

[10] Qi Sun. AGV system path planning techniques. Master thesis. Zhejiang University,2012.

[11] Lina He. Research of AGV system running path optimization. Master thesis. Nanjing University of Aeronautics and Astronautics,2011

[12] Hongyuan Zhang. Multi AGV path planning and collision avoidance based on Petri nets distributed. Master thesis. Northwestern Polytechnical University,2002.

[13] Qi Sun. AGV system path planning techniques. Master thesis. Zhejiang University,2012.

[14] Yiduo Bai. Analysis and Solution for k shortest path problem. Wuhan University，2009，34(4): 492-494

[15] Wenlan Chen, Yinrong Pan. A practical algorithm times shorter and progressively shorter path. Solving Computer Applications and Software, 2006, 01: 94-96.