Path Planning for Batch Picking of Warehousing and Logistics Robots Based on Modified A* Algorithm

Li-Mei Duan
Kunming Metallurgy College, Yunnan Kunming, China
`dlimeimetal00@126.com`

Abstract—To make the picking path planning of warehousing and logistics robot in warehouse more efficient and real-time, a path planning for batch picking of warehousing and logistics robots based on the modified A* algorithm was put forward. First of all, the path planning of the single robot batch picking was realized. Then, the time cost was introduced to further improve the A* algorithm to realize the batch picking path planning of the multi-warehouse logistics robots. Finally, according to the algorithm proposed, .Net visual programming technology was applied to build a complete warehouse logistics robot picking path planning simulation system to verify the effectiveness of the algorithm proposed. The research results showed that the algorithm proposed could meet the needs of efficient and real-time path planning in the process of completing batch picking tasks. To sum up, it lays a solid foundation for the realization of intelligent and fully automated warehouse management.

Keywords—warehousing and logistics robot, A* algorithm, path planning, batch picking

1 Introduction

Intelligent warehouse is a new automated warehouse built on the basis of artificial intelligence (AI) technology. It has the characteristics of high automation, low labour cost and high efficiency. The warehouse logistics robot plays a key role in the process of building an intelligent warehouse. As everyone knows, the picking is very critical to the operation of most of the warehouse, which accounted for about 55% of the warehouse operation cost [1]. It also directly determines the downstream customer service experience. Therefore, to solve the problem of how to realize the further improvement has very important realistic significance to the picking efficiency by logistics robot, but the key to solve the problem lies in the efficient path planning in view of logistics robot picking in different situations.

The routing planning of the moving robot in warehousing and logistics is a key technology to realize the autonomous navigation in the process of the sorting operation of the warehousing and logistics robots. Using the good path planning technology can not only save logistics robot operation time, reduce wear and reduce energy consumption, but also can reduce the production cost of the moving robot, reduce capital investment,

improve warehousing and logistics robots to warehouse environment adapting ability lay a solid the basis for realizing intelligent warehouse, automatic management [2].

2   State of the art

Warehousing and logistics robot is a new type of moving robot applied in the field of warehousing and logistics technology. It can independently accomplish the tasks of goods location, cargo transportation and even goods picking in warehouses, and it is the leading role of future intelligent warehouse. At present, foreign warehousing and logistics robots are mainly divided into two categories. One is the Kiva robot system represented by Amazon Company and the other is two kinds of robots with goods taking and delivering function represented by Fetch Company. Domestic storage and logistics robot technology research started relatively late, but the speed of development is faster. At present, the "Cao Cao" robot, similar to Fetch Company's delivery robot developed by Alibaba, has been applied to the warehouse of Alibaba, which mainly has the function of independently locating goods and transporting goods. The main drawback of this type of robot is that the picking process of the goods should be aided by the robot, and it cannot reduce the human cost well [3]. The path planning algorithm studied in this paper is mainly for this type of robot or the robot that will be able to pick up goods and transport goods in the future.

At present, there are relatively few professional and systematic researches on the picking path planning of warehousing and logistics robots at home and abroad. Most of the literatures are about the path planning of moving robots and the planning of artificial picking paths. The centralized multi-robot path planning method is more suitable for the known conditions of the global environment. The distributed multi-robot path planning method is more suitable for the known conditions of the local environment [4]. Considering that the working environment of multi warehouse logistics robot is fixed and generally does not change significantly, the centralized approach is applied to the path planning of multi-warehouse logistics robot.

3   Methodology

3.1   Warehouse environment modelling and picking task description

The layout of the warehouse channel in this paper is a decussation channel with multi-lateral channels, as shown in Figure 1. Multi-lateral channel can effectively reduce the traffic jam of multi-warehouse logistics robot.

In order to facilitate the research of the picking path planning for the warehousing logistics robot, this paper makes some assumptions about the order and the arrangement of the goods on the shelf. Each order has only one type of commodity, and the quantity of the goods for each order is changed. The types of goods for each order in batch are not conflicting. The number of orders in each batch is changed. Assume that there is only one type of goods on each shelf. Based on the above assumptions and the

warehouse model established in this article, the task flow of the sorting operation is roughly as shown in Figure 2.
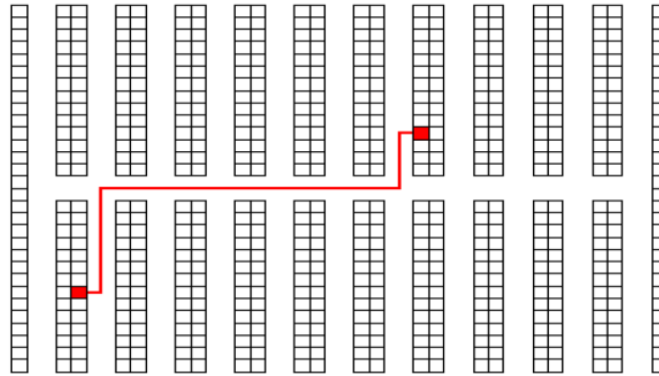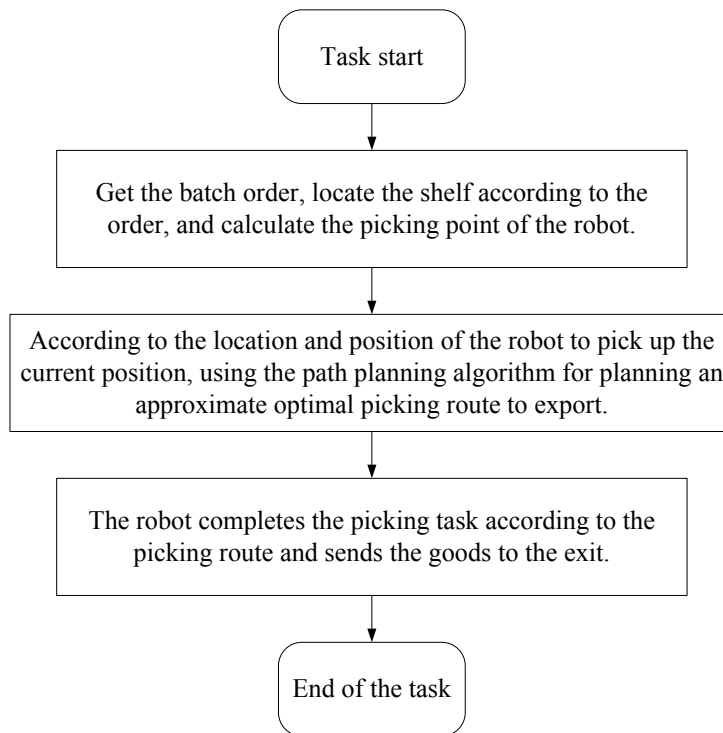

Fig. 1. Decussation channel

Fig. 2. Sorting task flow chart

3.2 Single picking path planning for warehousing and logistics robots

The path planning of a single mobile robot is to solve an optimal path from the starting point to the end point. Aiming at single warehouse logistics robot single order path planning algorithm design, we only consider time complexity, space complexity, moving rule, path length and moving time. In terms of a path planning algorithm for warehousing logistics robot by single selection, the A* algorithm is used in this paper. Because the A* algorithm has a small time complexity and space complexity. It also has a strong adaptability and it can meet the need for multi-robot scheduling [5].

The A* algorithm is a heuristic search algorithm, which uses the valuation function to find the optimal path from the starting point to the end point. The general form of the valuation function is:

$$f(n) = g(n) + h(n) \tag{1}$$

g(n) represents the actual distance from the starting point startNode to the current node n, and h(n) refers to the estimated distance between the current node n and the end point endNode. When h(n)≤h*(n) and it has the optimal distance from the current node n to the end point endNode, it can ensure the A* algorithm to get the optimal path [6].

According to the moving rules of warehousing and logistics robot, we should choose four-neighbourhood searching in search mode, and at the same time, we should use Manhattan distance for unconstrained distance estimation function, and its formula can be expressed as:

$$h'(n) = abs(nx - endNode.x) + abs(nx - endNode.\text{y}) \tag{2}$$

In addition, due to the different time consumptions and energy consumptions of the warehousing logistics robot, we should add the steering cost ε0 to h(n), so the form of heuristic function is:

$$h(n) = h'(n) + \varepsilon_0 \tag{3}$$

ε0 represents the turning cost and the value is:

$$\varepsilon_0 = \begin{cases} 0 & dis(l,n) = 2 \\ 1 & dis(l,n) = \sqrt{2} \\ 2 & dis(l,n) = 0, and \quad m = pickNode \end{cases} \tag{4}$$

The dis(l, n) represents the Euclidean distance of the precursor node 1 between the current node n and the node m with the smallest f value, which is mainly used to judge the robot rotation. When dis(l, n) =2 (Figure 3), the robot turns 90 degrees and then the turning price is ε0=1; when dis(l,n)=2, the robot does not turn, then the turning price is ε0=0; when dis(l,n)=0, it shows that the robot may conduct a 180 degree turn. This case will only appear when the robot needs to return to the original starting point after it

reaches the picking point of pickNode and completes the picking task [7]. It is important to note that every time we use A* algorithm to plan the pickNode path of picking point, we need to remember the precursor node of picking point pickNode to determine the posture and position of the robot.
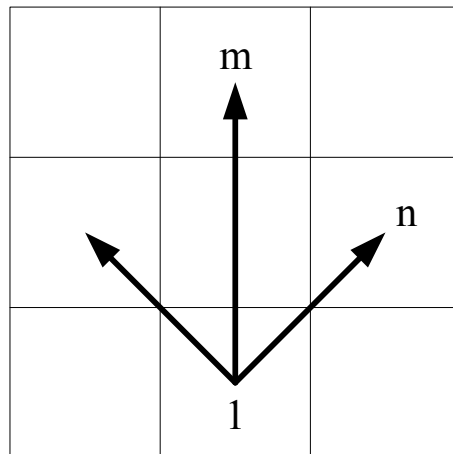

Fig. 3. Steering judgment

In this paper, the priority queue based on two-fork heap is used to manage the OPEN table. The time complexity of each optimization, insertion and deletion operation is O(logn). Therefore, the overall time complexity of A* algorithm is O(nlogn). The priority queue source used in this article is found on the NuGet platform. NuGet is an open source project under the.NET platform, which is an extension of VisualStudio [8]. In the process of using VisualStudio to write this program, the open source priority queue program found in Nuget is added to the project and can be directly quoted in the project.

3.3   Batch selection path planning for warehousing and logistics robots

In order to solve the problem of batch picking path planning for warehousing and logistics robots, a suitable mathematical model should be established firstly. Under the premise of considering the distance between point and point as the Manhattan distance, according to whether the starting point of robot is at the outlet, the problem of batch picking path planning can be divided into two types. One is that when the starting point of the robot is at the exit, the problem can be regarded as a classic TSP problem. The other is that when the starting point is not at the exit but in other positions, the problem can be considered as the open loop TSP problem with fixed starting and end point (hereinafter referred to as ST-TSP) [9]. The mathematical model of the classic TSP problem can be described as the formula:

$$\min Z\left(P_\beta\right) = \sum_{i=1}^{k-1} d\left(v_{\beta_i}, v_{\beta_{i+1}}\right) + d\left(v_{\beta_i}, v_{\beta_k}\right)$$
(5)

In view of the TS-TSP problem, s is recorded as the starting point of the robot, and e is the end of the robot, which meets the flowing formula.

$$\min Z\left(P_\beta\right) = \sum_{i=1}^{k-1} d\left(v_{\beta_i}, v_{\beta_{i+1}}\right) + d\left(s, v_{\beta_1}\right) + d\left(e, v_{\beta_k}\right)$$
(6)

In order to solve the TSP problem, we use the nearest neighbour algorithm in the genetic algorithm to generate the initial population. Combined with the characteristics of the batch picking task and the variable order number, we propose a new adaptive parameter design plan based on the size of the problem scale α.

First of all, on the whole, when the problem scale is small, the quality of the initial population is generally higher, the greater crossover probability δ and smaller mutation probability φ can be used to get the optimal solution as far as possible; when the problem scale α is large and the initial population is of poor quality, we can use the smaller crossover probability and greater mutation probability φ to protect the gene structure as far as possible, guarantee the quality of the solution and avoid local optimum. At the same time, in the iterative process, by considering the number of iterations η and the fitness value f changes, we further improve the solution quality and convergence speed [10].

The range of cross probability δ value in this paper is 0.80-0.95, δmax=0.95 and δ values are shown as follows.

$$\delta_{\min} = \begin{cases} 0.90 & 2 \leq \alpha \leq 20 \\ 0.85 & 20 < \alpha \leq 30 \\ 0.80 & 30 < \alpha \end{cases}$$
(7)

$$\delta = \begin{cases} \delta_{\max} - \left(\delta_{\max} - \delta_{\min}\right)\left(\dfrac{\eta'}{2\eta} + \dfrac{f_{\max} - f_i}{2\left(f_{\max} - \overline{f}\right)}\right), f_i \geq \overline{f} \\ \delta_{\max}, f_i < \overline{f} \end{cases}$$
(8)

Similarly, the value of mutation probability φ is taken, which is generally small. In this paper, the range of the value of φ is taken in 0.01-0.05, where φmax=0.05, and the formula of the φ value is shown below [11].

$$\varphi_{\min} = \begin{cases} 0.01 & 2 \le \alpha \le 20 \\ 0.03 & 20 < \alpha \le 30 \\ 0.05 & 30 < \alpha \end{cases}$$

(9)

$$\varphi = \begin{cases} \varphi_{\max} - \left(\varphi_{\max} - \varphi_{\min}\right)\left(\dfrac{\eta'}{2\eta} + \dfrac{f_{\max} - f_i}{2\left(f_{\max} - \overline{f}\right)} \right), f_i \ge \overline{f} \\ \varphi_{\max}, f_i < \overline{f} \end{cases}$$

(10)

There are six main steps in the implementation of the batch selection path planning algorithm. Firstly, input the problem scale α, population size β, the maximum crossover probability δmax, the maximum variation probability φmax, and the maximum number of iterations η. Then, output the picking path P. The first step is to make use of the nearest neighbour algorithm to produce the initial population with scale of β. The second step is to calculate the fitness fi of the individual (the reciprocal of the path length), the average fitness $\overline{f}$ and the maximum fitness fmax. The third step is to calculate the cumulative probability λ of the individual in the population and the selected probability λi and choose individual to generate new population by means of roulette. The formulas (7) and (8) are used to determine the cross probability δ, and generate a new individual. The mutation probability is determined by the formulas (9) and (10), and the mutant individual is produced. The fourth step is to decide whether it satisfies the iteration termination condition. If satisfies, end the loop and save the order L of access node, and move to the fifth step; otherwise it will go to the second step. The sixth step is to use the improved A* algorithm to give the path of adjacent nodes in L and merge into the complete path P and output [12].

3.4 Batch picking path planning for multi warehouse logistics robot

In this paper, the minimum time cost is chosen as the optimization goal of the batch picking path planning model of multi warehouse logistics robot. The time spent for picking a single robot is mainly composed of the time of robot picking up the goods (PT), the time of robot moving (RT) and the waiting time (WT) of the robot [13]. On the basis of single robot batch picking mathematical model, we can get the mathematical model of multiple warehouse logistics robot batch picking path planning based on the appropriate modification and the model is shown in the following formula.

$$\min T\left(P_{\beta}\right) = \left(\sum_{i=1}^{k-1} d\left(v_{\beta_i}, v_{\beta_{i+1}}\right) + d\left(v_{\beta_1}, v_{\beta_k}\right) \right) * t + RRT + WT$$

(11)

$$\min T\left(P_{\beta}\right)=\left(\sum_{i=1}^{k-1}d\left(v_{\beta_i},v_{\beta_{i+1}}\right)+d\left(s,v_{\beta_1}\right)+d\left(e,v_{\beta_k}\right)\right)*t+RRT+WT$$
(12)

When multiple warehousing and logistics robots batch pick in warehouse, the path of robots may be in conflict. When the conflict of multiple robots is more, it is easy to cause channel blockage, which affects the efficiency of robot executing batch picking task [14].

From Figure 4, we can see that when robot 1 and robot 2 arrive the same point with the same distance at the same time with the same speed, there will inevitably be a conflict of paths, that is, two robots will appear at the same position at the same time.
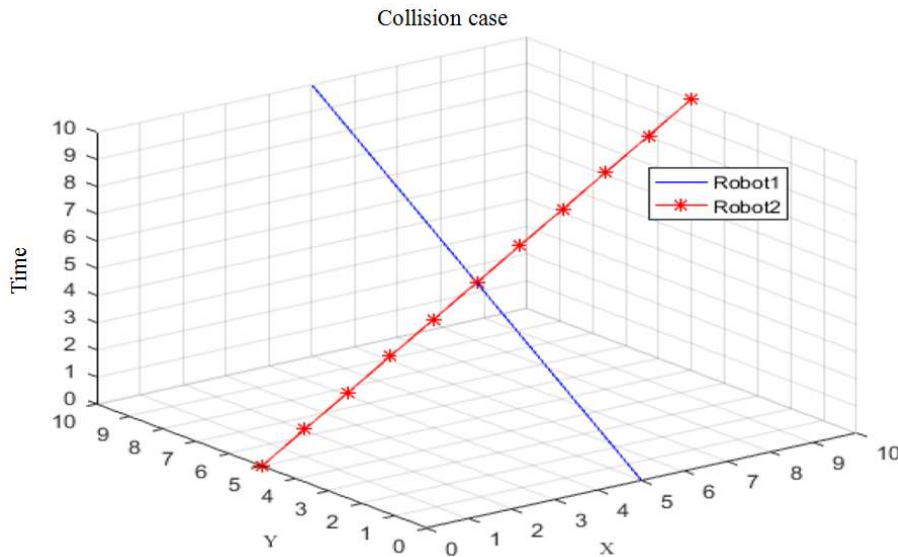

Fig. 4. The situation of path conflict

For obstacle avoidance planning, we first of all need to determine the priority of two robots with path conflict, and the one low priority needs to adjust the temporal and spatial information of the path. Secondly, the adjustment of time and space information of the robot with low priority should be based on the type of conflict to determine [15]. The strategy adopted in this paper allows the robot to enter the channel to wait and give the corresponding conflict resolution strategy according to the specific type of conflict. The result of the obstacle avoidance is shown in Figure 5.
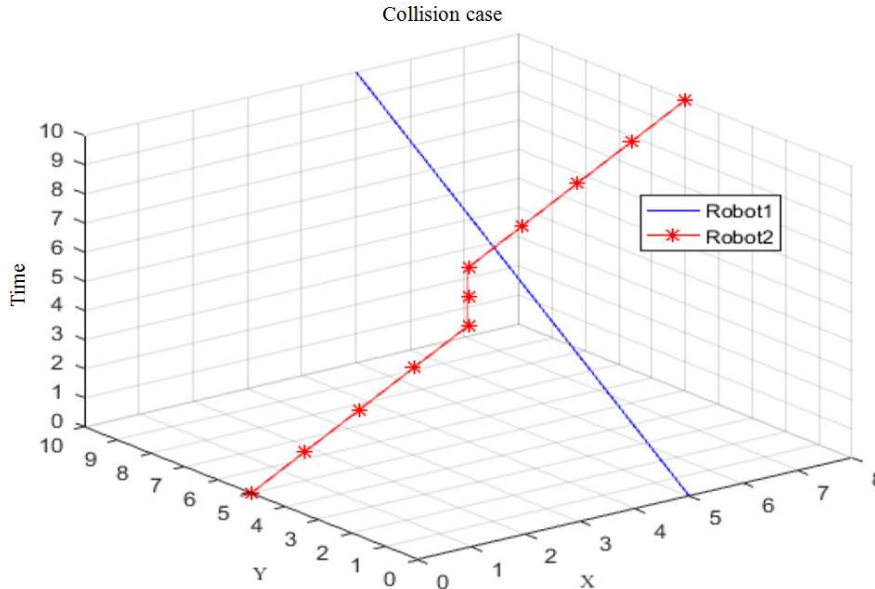
Fig. 5. Path conflict resolution

4   Results and discussion

4.1   Algorithm implementation of warehouse logistics robot based on single picking path planning

According to the single picking path planning process of logistics robot, first of all, start from the starting point to plan the path to picking point; secondly, from the starting point, plan the path to the original starting point. A* algorithm is called for two times in the process and the core idea is to use A* algorithm to segment routing. First, input the warehouse grid map G, starting point startNode, end point endNode, and output the optimal path from the starting point startNode to the end endNode. The first step is to set the initial state and insert the starting point startNode into the OPEN table. The second step is to determine whether the OPEN table is empty, and if it is not empty, then turn to the third step; otherwise end the loop and path finding fails. The third step is to extract the node m with the minimum f value in the OPEN table and determine if it is the end point e. If it is not the end point, it will go to the fourth step; otherwise, end the loop and return to the optimal path. The fourth step is to traverse the neighbour node of m node, used as the current node n and calculate f(n). If the current node has not been inserted into the OPEN table, then it is inserted into the OPEN table; otherwise, determine whether the g(n) value of current node n is less than the original g(n) value. If it is less than that, then change the parent node and update the OPEN table; otherwise, calculate the f value of the next neighbour node m. The fifth step is, after traversing the neighbour node of the m node, to return to the second step.
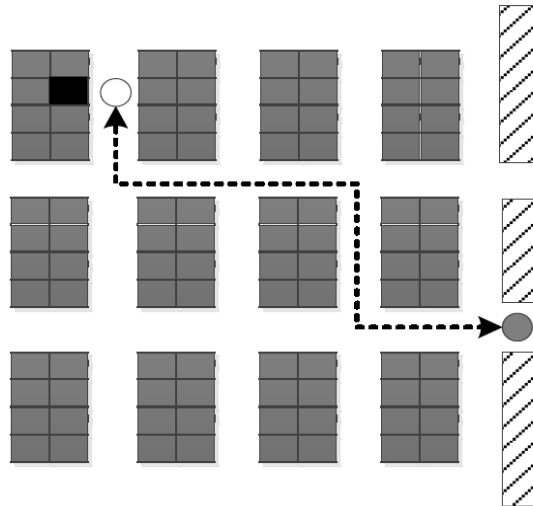
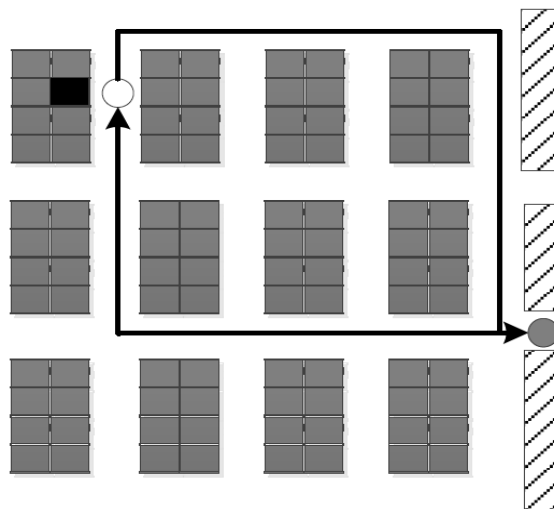**Fig. 6.** Implementation effect of traditional single picking path planning algorithm

Fig. 7. Implementation effect of improved single picking path planning algorithm

The implementation effect of algorithms is shown in Figure 6 and Figure 7. Figure 6 is the path of traditional A* algorithm without adding steering cost (the dotted line represents the coincided path). According to the route for walking, the robot needs to have 7 turns, the right angle turning for 6 times and 180 degrees for 1 time. However, the A* algorithm with the steering cost added (Figure 7) requires only 4 times right angle turning. Therefore, the A* algorithm, which is added to the steering cost, can effectively meet the needs of robot path planning.

4.2 Comparison of batch picking path planning algorithms for warehouse logistics robots

The alleyway algorithm (AA), GA algorithm (traditional genetic algorithm), IAGA algorithm (improved adaptive genetic algorithm) and branch and bound method (BB) are combined with A* algorithm, respectively, and the average path length of the algorithm is compared. Alleyway algorithm is a path picking planning method for lateral multi - channel and picking points. The principle is that the robot starts from a point (usually at the entrance) and enters the picking alleyway, completes the sorting of goods in the picking channel and then directly turns to the next alleyway needing to conduct the picking, and finally returns to the starting point. Branch and bound algorithm is a traditional algorithm to solve the TSP problem. The solution is the best solution, but the algorithm has high time complexity and space complexity. When the scale is large, it is difficult for the computer to get effective results.

Each problem scale has 10 different picking point distributions, 10 experiments are conducted in each case, and the experimental results are shown in Table 1. It can be seen from Table 1 that, when the problem scale is small, the average path length of IAGA+A* algorithm is similar to that of GA+A* and BB+A* algorithm, significantly less than that of AA+A*; when the scale of the problem is in the middle, the average path length of IAGA+A* algorithm is smaller compared with that of AA+A* algorithm and GA+A* algorithm, while the BB+A* algorithm has been unable to obtain the calculation results; when the scale of the problem is large, the average path length of IAGA+A* algorithm is similar to or greater than that of AA+A* algorithm, but significantly less than that of GA+A* algorithm. Generally speaking, the IAGA+A* algorithm can well adapt to the path planning of small and medium scale problems. When the size of the problem is over 45, it is suitable for path planning by channel join algorithm. In addition, during the experiment, the IAGA+A* algorithm does not run more than 0.15 seconds each time.

Table 1. Genetic algorithm parameters

α	AA+A*	GA+A*	IAGA+A*	BB+A*
5	115	107.1	107.1	107.1
15	227	169.93	161.97	149.1
25	359.3	280.67	264.97	-
35	428	421.97	387.91	-
45	530.5	597.03	530.27	-
55	613.5	805.73	666.77	-

Overall, IAGA+A* algorithm, in different scale problems, its convergence speed, average path length and running time of the algorithm could achieve better results, especially small and medium scale problemsd can well meet the needs of the robot for effective and real-time path planning in the process of completing tasks in batch picking.

4.3 Comparison of batch picking path planning for multi warehouse logistics robot

In this paper, the designed simulation system is used to verify the solution of the multi warehouse logistics robot scheduling. The experiment is mainly to compare the active obstacle avoidance and passive obstacle avoidance by using the current algorithm. Active obstacle avoidance is to choose the optimal path from the global comprehensive time cost. Passive obstacle avoidance only considers the shortest path, and the strategy is only waiting in terms of the path conflict. During the experiment, the scale of the robot $M \in \{5,10,15,20,25,30\}$, the size of each robot uses the same order, and 10 experiments are carried out. Finally, from three aspects of the average path length, average conflict node number and average picking time, active avoidance and passive avoidance strategies are analyzed and compared.

From the comparison of the average path length, as shown in Figure 8, the overall path length of active avoidance all more than the passive avoidance and with the increase in the number of robot, the gap between the two is expanding, which is mainly because that the active avoidance strategy will choose a relatively long path length. But the path with low comprehensive cost and the increasing number of robots will result in the increase of conflict nodes. In addition, the instability of the genetic algorithm can cause the fluctuation of the selection path length.
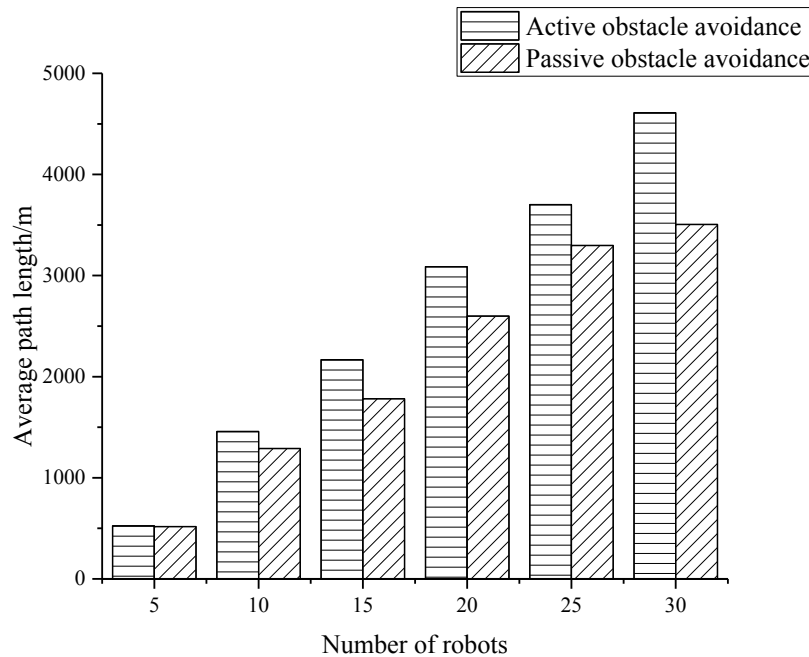

Fig. 8. Comparison of average path length

From the comparison of the average number of nodes of the conflict, as shown in Figure 9, with the increase of the number of robots in scale, the number of average conflict nodes in the active and passive avoidance increases. However, the average number of conflict nodes of active avoidance is significantly less than the number of nodes in the passive avoidance, and with the increasing number of robot scale, the gap between the two is gradually widening. This fully demonstrates the effectiveness of the proposed strategy to solve the path conflict of multi-robot.
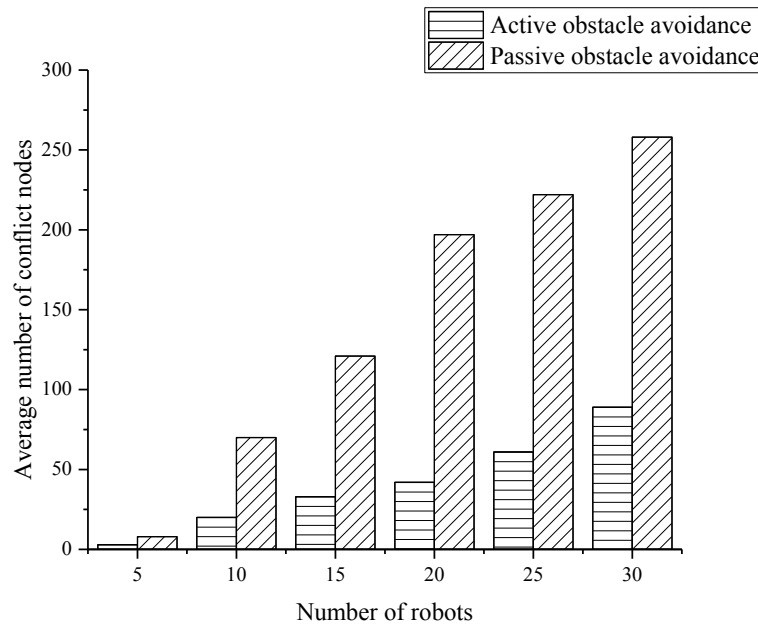

Fig. 9. Comparison of average number of conflict nodes

From the comparison of average picking time, as shown in Figure 10, with the increase of robot number, the average picking time of active obstacle avoidance and average obstacle avoidance is increasing and the gap is gradually widening. When the number of robots is small, the difference between average picking time and passive obstacle avoidance is not significant. This is mainly due to the fewer number of nodes in the path conflict when the number of robots is small, so the time saved by obstacle avoidance strategy is relatively short. Besides, the active obstacle avoidance strategy may choose a longer path than the passive obstacle avoidance strategy, which also reduces the time saved by the active obstacle avoidance strategy to a certain extent.
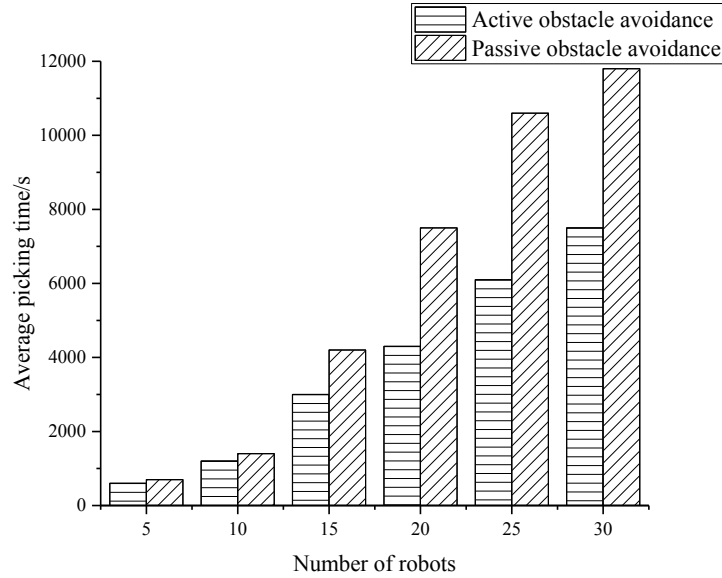
Fig. 10. Comparison of average picking time

Overall, the average path length has poor effect on active avoidance strategy than passive avoidance strategy, while the average number of conflict nodes and the average conflict picking time obtain good results and the effect becomes more obvious with the expansion of problem scale, which fully demonstrates the effectiveness of the proposed strategy.

4.4 Realization of the simulation system of path planning for warehousing and logistics robot

In order to verify the reliability of the proposed warehouse logistics robot picking path planning algorithm, we use the C# programming language to build a visual simulation system in the simulation tool Visualstudio2015, and the system operation flow chart is shown in Figure 11. The user interface of the visual experimental platform is Windows form based on the.NET framework. This simulation system is mainly used to verify single batch warehouse logistics robot batch picking path planning algorithm and multiple warehouse logistics robot batch picking path planning algorithm. The implementation of the system is shown in Figure 12.
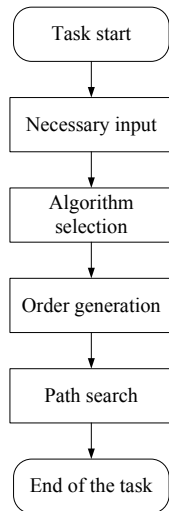
Fig. 11. System operation flow chart

Fig. 12. Schematic diagram of simulation system running

5 Conclusion

This paper made a systematic study on the routing planning of the storage and logistics robot. Firstly, a path planning method for single warehouse logistics robot single

picking - A* algorithm was given. And then the characteristics of logistics robot batch picking path planning were studied, as the TSP problem and the corresponding mathematical model was established. The improved adaptive genetic algorithm was used to get the picking access sequence and experimental simulation was conducted. In view of the possible various path conflicts in the multi-warehouse logistics robot, the method of obstacle avoidance was proposed. Finally, the simulation system of the path planning of the warehousing logistics robot was realized. The research results showed that the proposed logistics robot batch picking path planning based on modified A* algorithm could well meet the needs of the robot for efficient real-time path planning in the process to complete the batch picking tasks. It lays a solid foundation for the realization of intelligent warehouse and full automation management.

6   References

[1] Gao, W.F., Liu, S.Y., & Huang, L.L. (2013). A novel artificial bee colony algorithm based on modified search equation and orthogonal learning: IEEE Transactions on Cybernetics, 43 (3) : 1011. https://doi.org/10.1109/TSMCB.2012.2222373

[2] Yılmaz, S., & Küçüksille, E.U. (2015). A new modification approach on bat algorithm for solving optimization problems: Applied Soft Computing, 28 (5) : 259-275. https://doi.org/10.1016/j.asoc.2014.11.029

[3] Gonglin, Y., & Maojun, Z. (2013). A Modified Hestenes-Stiefel Conjugate Gradient Algorithm for Large-Scale Optimization: Numerical Functional Analysis & Optimiza-tion, 34 (8) : 914-937. https://doi.org/10.1080/01630563.2013.777350

[4] Chang, C.F., Lin, T.Y., & Tai, C.L., et al. (2013). Advanced Information of Parity Bits for Decoding Short Linear Block Codes Using the A* Algorithm: IEEE Transactions on Communications, 61 (4) : 1201-1211. https://doi.org/10.1109/TCOMM.2013.020813.120216

[5] Nascimento, T.P.D., Costa, P., & Costa, P.G., et al. (2013). A set of novel modifications to improve algorithms from the A* family applied in mobile robotics: Journal of the Brazilian Computer Society, 19 (2) : 167-179. https://doi.org/10.1007/s13173-012-0091-5

[6] Takeshi, N. (2013). Modification of ABC Algorithm for Adaptation to Time-Varying Functions: Electronics & Communications in Japan, 96 (11): 44-53. https://doi.org/10.1002/ecj.11479

[7] Adnanshiltagh, N., & Dalawr Jalal, L. (2014). A Comparative Study: Modified Parti-cle Swarm Optimization and Modified Genetic Algorithm for Global Mobile Robot Navigation: International Journal of Computer Applications, 89 (9) : 32-46.

[8] Mezura-Montes, E., Portilla-Flores, E.A., & Herna Ndez-Ocaa, B. (2014). Optimum synthesis of a four-bar mechanism using the modified bacterial foraging algorithm: In-ternational Journal of Systems Science, 45 (5) : 1080-1100.

[9] Zhan, W., Wang, W., & Chen, N., et al. (2015). Path planning strategies for UAV based on improved A* algorithm: Geomatics & Information Science of Wuhan University, 40 (3) : 315-320.

[10] Mahajan, M., & Preet Singh, T. (2014). The Modified Gaussian Function based RED (MGF-RED) Algorithm for Congestion Avoidance in Mobile Ad Hoc Networks: In-ternational Journal of Computer Applications, 91 (6) : 39-44.

[11] Liu, T., Xie, J.B., & Yan, W., et al. (2013). An algorithm for finger-vein segmentation based on modified repeated line tracking: Journal of Photographic Science , 61 (6) : 491-502. https://doi.org/10.1179/1743131X12Y.0000000013

[12] Zhou, D., & Zhou, H. (2015). A modified strategy of fuzzy clustering algorithm for im-age segmentation: Soft Computing, 19 (11) ：3261-3272. https://doi.org/10.1007/s00500-014-1481-8

[13] Azizi, R., Sedghi, H., & Shoja, H., et al. (2015). A Novel Energy Aware Node Cluster-ing Algorithm for Wireless Sensor Networks Using a Modified Artificial Fish Swarm Algo-rithm: International Journal of Computer Networks & Communications, 7 (3) ： 13. https://doi.org/10.5121/ijcnc.2015.7307

[14] Roberts, M.K., & Jayabalan, R. (2015). A Modified Optimally Quantized Offset Min-Sum Decoding Algorithm for Low-Complexity LDPC Decoder: Wireless Personal Communica-tions, 80 (2) ： 561-570. https://doi.org/10.1007/s11277-014-2026-2

[15] Defoort, M., & Djemaï, M. (2013). A Lyapunov-based design of a modified super-twisting algorithm for the Heisenberg system: Ima Journal of Mathematical Control & Information, 30 (2) : 185-204. https://doi.org/10.1093/imamci/dns020

7 Author

Li-Mei Duan is with the Kunming Metallurgy College, Yunnan Kunming, China (dlimeimetal00@126.com).