2019 IEEE Symposium Series on Computational Intelligence (SSCI)
December 6-9 2019, Xiamen, China

Robot Path Planning Based on Freeman Direction Chain Code

Meng Zhao
School of Electronic and Information Engineering
Beihang University
Beijing, P. R. China
meng960416@163.com

Hui Lu
School of Electronic and Information Engineering
Beihang University
Beijing, P. R. China
mluhui@buaa.edu.cn

***Abstract*—Path planning is one of the key technologies for robots to achieve autonomous movement, which is critical to realize intelligent operation. When facing complex environment, in order to plan the path as soon as possible, we propose a Freeman direction chain code path planning (FDCC-PP) algorithm for single and multi-robot systems. The FDCC-PP algorithm consists of two processes. The first process is map processing, in which the Freeman direction chain code is adopted to extract information about the position and direction of the obstacles in the map. The second process of the FDCC-PP algorithm is path planning, which includes moving toward the target point in free space and making a fast detour according to the Freeman direction chain code near the obstacle. In addition, in multi-robot path planning, we introduce the grid-dominant principle to handle collisions when the robots move along their planned paths. The comparison with other algorithms shows that the introduction of Freeman direction chain code can reduce the search redundancy in space and shorten the path planning time without sacrificing the safety of the robot, which makes fast path planning possible in complex environment.**

Keywords—path planning; boundary extraction; Freeman direction chain code; grid-dominant principle.

I. INTRODUCTION

In order to improve production efficiency, more and more factories employ mobile robots in single-mode jobs such as cargo transportation and device package. These applications promote the development of research on autonomous control of robots. Among most robotics technologies, path planning is the premise of robot autonomous mobile, which attracts extensive attention. In fixed working scenarios, the robots usually constantly adjust their pre-planned paths to complete their tasks on the basis of sensor information in their moving processes. Therefore, whether a robot can complete the task safely usually depends on the planned path, which shows the importance of research on the global path planning. However, the difficulty of robot path planning increases with the increasing number of robots in the workshop, especially in the environment where the number of robots changes at any time, which also poses a certain challenge to the real-time of path planning. Hence in order to improve the practicability of the path planning algorithm, in this paper, we make the best of prior map information and greatly reduce the time required for path planning by introducing a process of map processing before the planning process.

Considering that the previous research on global path planning has hardly involved pre-processing of obstacles in the map. In this paper, we propose a Freeman direction chain code path planning (FDCC-PP) algorithm by employing a boundary coding method to the obstacles in maps and apply it to the path planning for different numbers of robots. The FDCC-PP algorithm for a single robot path planning can be roughly divided into three steps. First, in order to ensure the safety of the robot, the obstacles in the map are dilated at a fixed safe distance. Then the boundaries of obstacles are encoded by Freeman direction chain code to obtain direction information of the static obstacle avoidance. Finally, the robot moves from the start point to the target point. When the next point on the line between the start point and the target point is an obstacle, it moves along the edge of the obstacle according to the direction information until it reaches the destination. As for the multi-robot system, another step is introduced into the path planning process, in which they have to determine whether to move according to the grid-dominant principle.

Our contributions of the proposed algorithm lie in the following aspects. To begin with, path planning by the FDCC-PP algorithm, the robot only needs to move towards the target point, and it is not necessary to independently search for the best path during the entire path planning process. Moreover, since the direction information of the edge of the obstacle has been obtained in the map processing, the robot only needs to query the chain code of the obstacle and make a fast detour in the vicinity of the obstacle, which can save a quantity of planning time. In addition, this algorithm also provides the possibility for real-time adjustment of multi-robot paths when the number of robots is not fixed in the environment.

The path planning results based on the grid maps show that the FDCC-PP algorithm can be used for both a single robot and multi-robot system. In addition, it can be found that the search redundancy in the process of path planning is reduced, the time required for path planning is decreased, and the real-time performance of the path planning algorithm is enhanced compared with the A* algorithm and the Dijkstra algorithm.

The rest of this paper is organized as follows. Section II gives a brief overview of the existing algorithms of robot path

978-1-7281-2485-8/19/$31.00 ©2019 IEEE

planning. Section III presents the Freeman direction chain code, which is the core of the FDCC-PP algorithm. The proposed algorithm is described in detail in Section IV. Section V lays out the experiment result and a brief analysis. Section VI summarizes the entire article and discusses the implication of the findings to future research into this area.

II. Related Work

Robot path planning is that a robot figures out a path from the start point to the target point and avoids collisions in the environment with obstacles. It is an important aspect of artificial intelligence in robotics area because it is the foundation of robot autonomous control and navigation. Therefore, a considerable amount of algorithms on global path planning have been proposed in the past few decades, which can be roughly classified into three types: layer-by-layer search algorithm, heuristic algorithm and evolutionary algorithm.

Layer-by-layer search algorithm centers on the start point and extends to the target point layer by layer. Since almost all nodes between the start point and the target point are traversed during the search, planning by this algorithm usually can obtain the global shortest path but costs a lot of time. There are two classic algorithms based on this plan principle, namely Dijkstra algorithm and Floyd algorithm [1], [2]. To improve the efficiency of path planning, many scholars further optimized the process of path searching. Yuan Y and Wang D W combined Dijkstra algorithm with ant colony algorithm, which made rapid path planning possible in emergency logistics management [3]. O.V. Gnana Swathika et al. proposed a Prims aided Floyd algorithm, where the Prims algorithm output active nodes in the current topology of the network to aid the Floyd algorithm in identifying the shortest path from the node closer to the fault to the utility grid or point of common coupling in the event of fault occurrence [4].

In order to reduce the path planning time, scholars have introduced heuristics in the study of global path planning and obtained many research achievements. For example, A* algorithm has been widely used since it was proposed [5]. It considers the cost of moving from the current position to the start point and the target point, which reduces the planning time greatly while obtaining the optimal or sub-optimal path. To improve the practicability of the algorithm and apply it to the path planning in dynamic environment, Stentz A et al. improved A* algorithm and proposed D* algorithm [6], [7]. In addition to this, an improved bidirectional time-efficient A* algorithm was proposed in the paper of Mindong G et al., which decreased the time of the algorithm by 76. 8% [8]. However, although A* algorithm takes the traction of the target point into account, it is unavoidable to compare the moving cost of the sub-nodes many times when planning by this algorithm, which leads to the problem of search redundancy in the maps with more free space.

In the global path planning process, constraints such as path length, mobile security, and path smoothness need to be comprehensively considered. Therefore, global path planning is a multi-objective optimization problem. Evolutionary algorithms have natural advantages in solving multi-objective problems. Therefore, evolutionary algorithm is widely used in the research on robot path planning. For example, in the algorithm proposed by Marco A. Contreras-Cruz et al [9], they found a feasible path in the free space by artificial bee colony algorithm and optimized the path length and smoothness by evolutionary programming. The simulation result showed that the path obtained by this algorithm was better than the PRM algorithm. In addition, some scholars applied PSO algorithm [10], ant colony algorithm [11] and genetic algorithm [12] in the robot path planning. However, evolutionary algorithms usually require a lot of iterations, and the implementation of the algorithm requires many parameters, which may affect the effect of path planning.

In view of all that have been mentioned so far, it can be found that most of the studies on path planning focused on the adjacency relationship between path nodes, ignoring the use of obstacle information provided in the map. Consequently, although there are a lot of methods, global path planning based on known maps still remains challenging and unsolved in terms of the following problems and obstacles. First, the excessive pursuit of traversal search to obtain the global shortest path may make the planning time too long, which is not suitable for tasks with high real-time requirements. Second, a large amount of search redundancy is caused by multiple comparisons of the moving cost of adjacent nodes in maps with more free space. Third, the complex modeling and training process make the generalization ability of the algorithm poor. As a result, the existing path planning algorithms are still not satisfactory so far. In this paper, we will focus on making best use of map information to reduce search redundancy and improve the efficiency of path planning algorithm, so as to achieve fast path planning for robots.

III. Freeman Direction Chain Code

Freeman direction chain code (also known as chain code) is used to represent a boundary by a connected sequence of straight-line segments of specified length and direction. Typically, this representation is based on 4- or 8-connectivity of the segments, as shown below.

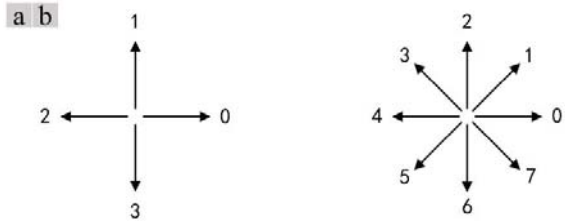

Fig. 1. Direction numbers for (a) 4-directional chain code, and (b) 8-directional chain code.

Because digital images usually are acquired and processed in a grid format with equal spacing, the Freeman direction chain code can be easily generated by following a boundary in a clockwise direction and assigning a direction to the segments connecting every pair of pixels [13]. Freeman direction chain code is widely used in digital image processing. In order to avoid the chain code being too long and taking into account the pixel characteristics of the image, the common method used in image boundary tracing is resampling the boundary by selecting a larger grid spacing, as shown in Fig. 2.

In addition, due to the periodicity of the chain code, the

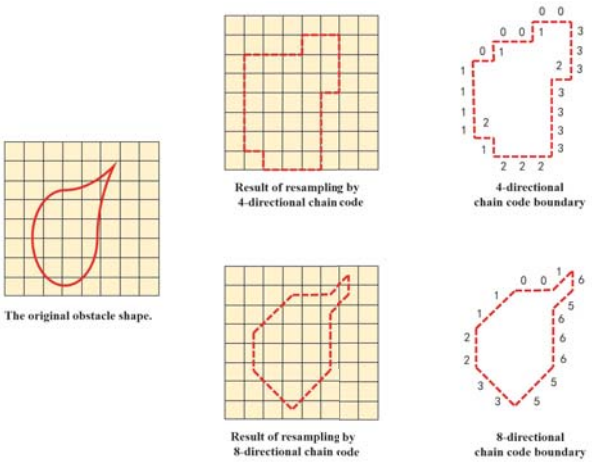

Fig. 2. The resampling process of the obstacle boundary.

vertexes of the graphics can be gained by introducing a differential chain code, which helps the robot to determine the temporary target point when avoiding obstacles. The differential chain code is obtained by counting the number of direction changes (in a counterclockwise direction in Fig. 2) that separate two adjacent elements of the code. For instance, the differential chain code of the 4-direction chain code 111000333222 is 300300300300, the starting coordinate point corresponding to the non-zero values of this code sequence are the vertexes of the graph, and the specific process is as follows.

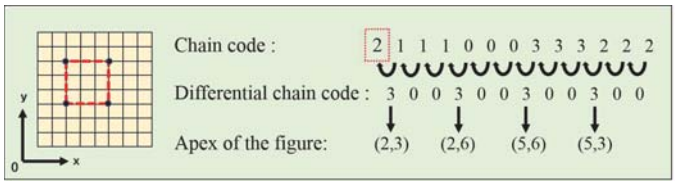

Fig. 3. The process of calculating the vertexes of a graph by the chain code.

IV. FDCC-PP Algorithm

The FDCC-PP algorithm can not only achieve the fast path planning of a single robot in the working environment where the map is known, but also can be used for multi-robot path planning by introducing the grid-dominance principle, and the planning efficiency is higher than that by other algorithms. It can be divided into two processes: map processing and path planning. In this section, we will introduce the FDCC-PP algorithm in detail, and show the pseudo codes of this algorithm used for a single robot and multi-robot respectively.

A. Map Processing

Map processing is the core part of the FDCC-PP algorithm, which consists of four steps: binary processing, dilation processing, obstacle boundary extracting and Freeman direction chain code editing. The design of these steps not only guarantee the safety of the robot, but also extract boundary information of obstacles in the map, which paves the way for the follow-up path planning. The changes of the map in map processing are shown in Fig. 4.

1) Binary processing

First, considering the different forms of map storage and

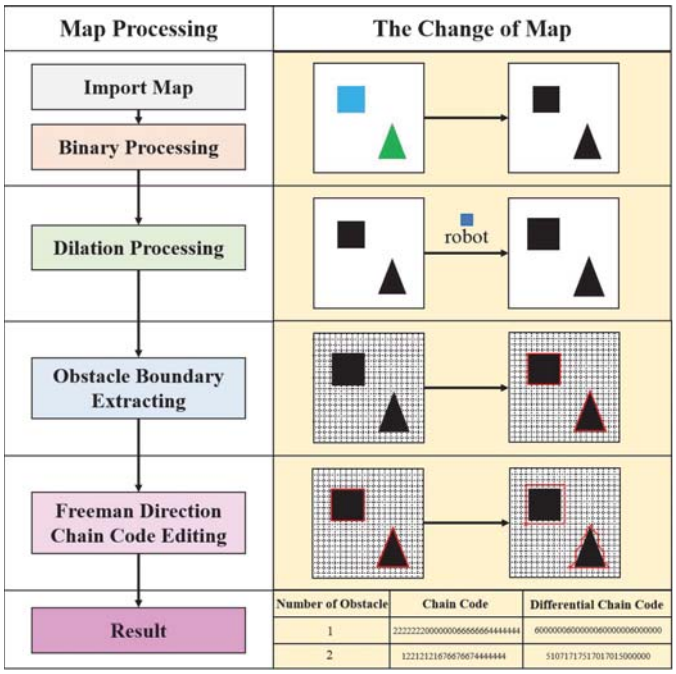

Fig. 4. Map Processing Based on 8-directional Freeman chain code.

the diversity of obstacles, we need to binarize the map to identify and extract obstacles in the map to facilitate subsequent processing. There are only two kinds of pixel values in the processed map, 0 and 1, which represent free space and obstacle space respectively.

2) Dilation processing

In order to ensure the practicability of global path planning algorithm, it is necessary to take the size of robot into account in the map processing. Therefore, we choose the size of the robot as the structural element to dilate the obstacles in the map. This process realizes the crack connection in the narrow space where the robot can not pass, and enables the robot to move freely on the planned path.

3) Obstacle boundary extracting

Since the imported map has been binarized, the pixel values in the map are 0 and 1. In this step, we select the connected regions of obstacles according to the pixel value at first. Then the grid coordinates at the boundary of the pixel values 0 and 1 can be got by edge detection method. Up to now, we have obtained the coordinate information of the boundaries of the obstacles in the map.

4) Freeman direction chain code editing

After the above three steps, the boundary information of the obstacles in the map has already been got. Therefore, in the last process, we introduced the Freeman direction chain code to code the boundary of obstacles, so that the robot can make a detour quickly. Based on the direction indication information provided by the chain code, the robot can judge which path to bypass the obstacle is shorter. Afterwards, it moves along the shortest path to avoid the obstacle and continue to move to the target point. Apart from that, the first difference of the chain code can provide the robot with a temporary moving target in the process of detouring.

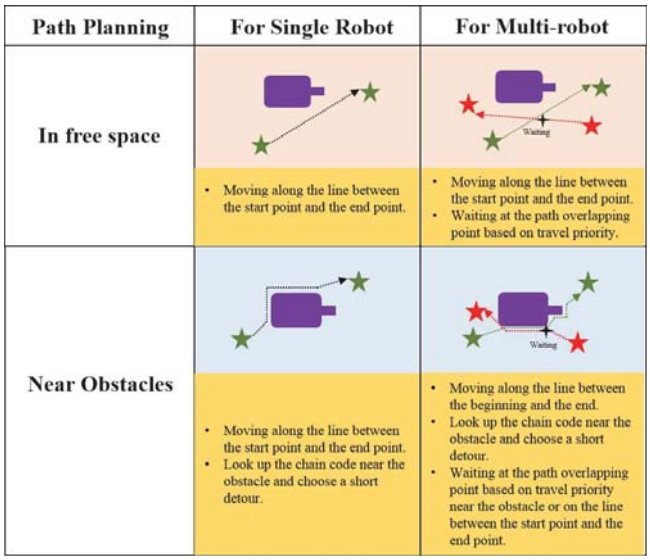

Fig. 5. The comparison between the single robot and multi-robot when planning a path by the FDCC-PP algorithm.

After the above processing, the boundary and direction information of obstacles can be extracted from the map. This information is the premise of robot path planning, which can help the robot to reach the destination faster and safer.

B. Path Planning

Path planning is the second process of the FDCC-PP algorithm. In order to minimize the path and improve the efficiency of path searching, the realization of this process is mainly based on the principle of human path planning in unknown environment. This process can be divided into two steps, moving towards the target point in the free space and moving around the obstacle in the vicinity of obstacles. However, because there is no feedback of sensor information in the simulation verification phase of the algorithm, we judge whether the next moving point of the robot is an obstacle or not by the obstacle information obtained from map processing. If the robot is in the free space, it continue to move towards the target point. If the robot enters the obstacle space at the next moment, it would make a quick detour based on the direction guidance of the obstacle.

Different from other algorithms, the coordinate information and direction information of obstacle boundaries has already been obtained in the map processing in the FDCC-PP algorithm. Therefore, when the robot is near an obstacle, it only needs to query the direction chain code of the corresponding obstacle and chooses the shortest detour, instead of moving around the obstacle to find the best detour path before moving towards the destination again. Considering that there are some differences in the path planning for a single robot and multi-robot, in this part, we will describe the path planning for a single robot and multi-robot respectively, and compare them in Fig. 5.

1) Path planning for a single robot

In order to minimize the path of the robot, we adopt the principle of moving towards the target point, moving along the line between the start point and the target point in the free space. If the next position that the robot is going to move is an obstacle, where the robot may collide. We would take three steps to bypass the obstacle. First of all, we need to find the index of the obstacle, which can be gained by querying the boundary information of obstacles according to the coordinates of the next position. Then, the vertexes of the obstacle can be obtained by calculating the differential chain code of the obstacle. Then the robot can get the current destination round the obstacle by comparing the Euclidean distance from each vertexes of the obstacle to the target point. In order to bypass the obstacle as soon as possible and move to the target point again, the vertex of the obstacle nearest to the target point is selected as the robot's current destination in the FDCC-PP algorithm. Finally, the path with fewer moving steps is acquired by comparing the number of steps needed for clockwise and counterclockwise bypass the obstacle, and the corresponding chain code is recorded as the direction indication for the robot movement.

The pseudo code of the FDCC-PP algorithm used for a single robot path planning is given below. Where ***cp***, ***np***, ***S*** and ***E*** are the current position, the next position, the start point and the target point of the robot in the path planning, respectively.

Algorithm 1: FDCC-PP Algorithm for a Single Robot

```
Import the map of the workshop as a picture
Initialize S, E, cp, np, route = []
Map processing
while cp ~= E                    //path planning
  calculate np
    if map (np) == 1
      find the obstacle number where np is
      calculate the current destination round the obstacle
      compare the path lengths of two detours
      store the shortest detour in route
      cp = current destination
    else
      store np in route
      cp = np
    end if
end while
Output route as the planned path for the robot
```

2) Path planning for multi-robot

There is no continuous updating of the relationship between nodes in the whole process of path planning in the FDCC-PP algorithm, it is only based on the Freeman direction chain code querying mechanism of obstacles. So this algorithm can also be applied to the global path planning for multi-robot through simple improvement.

Different from planning for a single robot, the collision risk between robots should be taken into account when the FDCC-PP algorithm used for multi-robot. Therefore, we introduce the a grid-dominance principle to ensure the safety of robots. The detailed planning steps are as follows. First, the priority of robots is ranked according to the importance of their work in

the workshop. Second, the robot with the highest priority plan the first step of its path, then, the robot with the second priority plan its first step. Rested on this sequence, the path points of each robot are planned one by one until all robots reach their respective destinations. Based on this grid-dominance principle, every robot needs to make sure that the robot with higher priority will not appear in the position at the same time before it store this position in its own planned path. For example, assuming that robot A has higher priority than Robot B, if a position is already the path point of robot A and the timestamp is the same as that of robot B, then robot B would choose to wait a step in the position of the previous step to avoid collision with robot A according to the grid-dominance principle. Apart from that, the rest process of path planning is the same as that for a single robot.

The pseudo code of the FDCC-PP algorithm used for multi-robot path planning is given below. We assume that there are n robots in the workspace, and cp_i, np_i, S_i and E_i are the current position, the next position, the start point and the target point of the ith robot in the path planning, respectively.

Algorithm 2: FDCC-PP Algorithm for Multi-robot

```
Import the map of the workshop as a picture
Initialize S, E, cp, np, route_i = []  for each robot
Map processing
while any ( cp_i ~= E_i )                    //path planning
    for i = 1 : n
        calculate np_i
        if map ( np_i ) == 1
            find the obstacle number where np_i is
            calculate the current destination
            compare the path lengths of two detours
            if i ~= 1
                insert the waiting point at the overlapping point
                store the new detour path in route_i
            else
                store the shorter detour path in route_i
            end if
            cp_i = current destination
        else
            if i ~= 1
                insert the waiting point at the overlapping point
                store waiting point and np_i in route_i
            else
                store np_i in route_i
            end if
            cp_i = np_i
        end if
    end for
end while
Output route_i for the ith robot
```

C. Summary

The FDCC-PP algorithm employs the Freeman direction chain code to extract information about the position and direction of the obstacles during map processing. In order to plan the path as short as possible, the robots are allowed to move to the target point in the free space and choose the shortest detour with the help of obstacle information in the proximity of the obstacle. Apart from that, because the FDCC-PP algorithm makes full use of obstacle information gained from the map, the relationship between nodes in the map is fixed. This algorithm can be applied in the path planning for different number of robots. Different from a single robot path planning, a grid-dominant principle is used to deal with collisions in the multi-robot path planning. The figure below shows the detailed comparison between the FDCC-PP algorithm used for single and multi-robot systems.

V. Experiments

The FDCC-PP algorithm extracts the obstacle information provided in the map by adopting the Freeman direction chain code in the map processing, so that the robot can obtain the shortest detour by querying the corresponding obstacle chain code in the process of path planning. This planning strategy greatly improves the efficiency of path planning. Besides, due to the introduction of the grid-dominant principle, this algorithm can also complete the path planning for multi-robot system in a short time, which further demonstrates the practicability of FDCC-PP.

In this part, we designed some experiments from two aspects in order to verify the performance of the FDCC-PP algorithm. First, we drew two types of grid maps, on which we compared the results of the FDCC-PP algorithm with the A* algorithm and the Dijkstra algorithm base on the same path planning task. Second, in order to illustrate the FDCC-PP algorithm can be used in the path planning for multi-robot, we designed a working scene of three robots, and compared the moving timestamps of each robot at the overlapping points of paths to verify that the robots are safe when they are moving.

A. Path Planning for A Single Robot

We designed two different types of maps that include most of the elements in the workplace to verify the stability of the FDCC-PP algorithm used in robot path planning. The sizes of these maps are 400×400. In the first map, there are more obstacles and less free space, and fewer obstacles and more free space exist in the second map. The results of path planning in these maps based on the A* algorithm, the Dijkstra algorithm and the FDCC-PP algorithm are shown in Fig. 6, Fig. 7 and Fig. 8, respectively, where the yellow area is the area searched during the path planning process, the blue curve is the boundary of search area, and the magenta curve is the final planned path. In addition, the robot needs to move from point (5, 5) to point (375, 270) in both maps.

Since the searching strategies of the A* algorithm and the Dijkstra algorithm are both based on the comparison of node moving cost and the renewal of relation between nodes, there are large search areas in Fig. 6 and Fig. 7. That results in a slew of search redundancy in an environment with fewer obstacles, which can be clearly seen in Fig. 7(b). However,

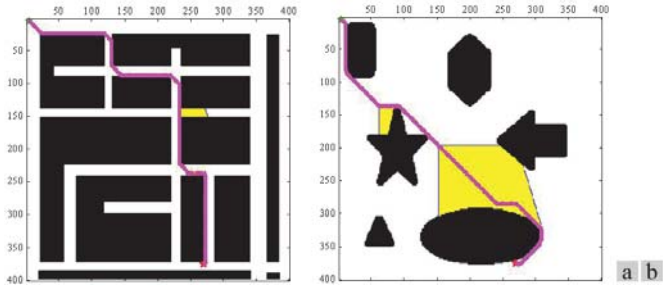

Fig. 6. The path planning results of the robot by using A* algorithm.

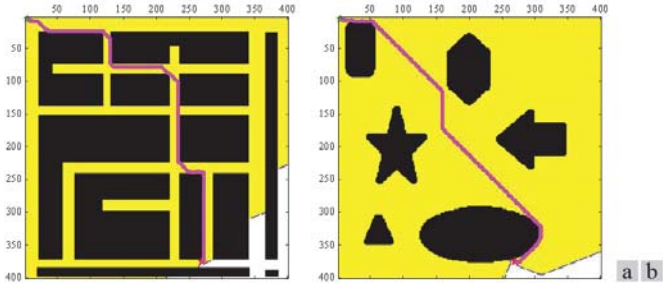

Fig. 7. The path planning results of the robot by using Dijkstra algorithm.

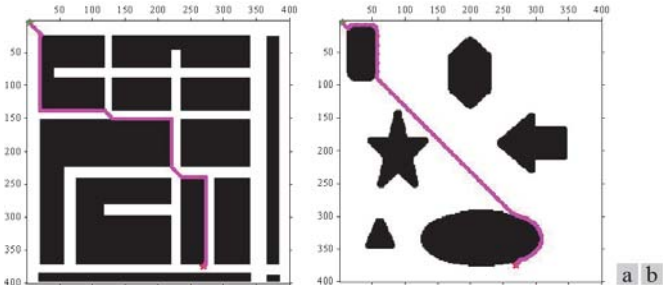

Fig. 8. The path planning results of the robot by using the FDCC-PP algorithm.

according to the paths shown in Fig. 8, we can see that the robot moves towards the target point in the free space and selects the shorter path near the obstacle to the apex of the obstacle closest to the target point. There is no complicated comparison process around obstacles in the whole path planning process, which is completely based on the boundary direction chain code of the obstacle. Moreover, in order to further demonstrate the efficiency of this path planning strategy, we compared the superiority of the FDCC-PP algorithm with the A* algorithm and the Dijkstra algorithm based on these two maps in the Table I and Table II, respectively.

The comparisons shown in Table I and Table II illustrate that due to the use of Freeman direction chain code, the path planning efficiency of the robot can be increased by 76.14% compared with the A* algorithm in the map with more free space, and be even increased by 96.67% compared with Dijkstra algorithm. Even if there are more obstacles in the map, the path planning efficiency of FDCC-PP can be improved by 92.71% compared with the Dijkstra algorithm, which is similar to the A* algorithm with heuristic search. The improvement of planning efficiency makes the FDCC-PP algorithm more useful in robot path planning.

TABLE I. THE COMPARISON BETWEEN FDCC-PP ALGORITHM AND TWO CLASSICAL ALGORITHMS BASED ON MAP1

Algorithm	Average Planning Time	Path length
A* algorithm	0.5974s	563
Dijkstra algorithm	8.1084s	563
FDCC-PP algorithm	0.5908s	600

TABLE II. THE COMPARISON BETWEEN FDCC-PP ALGORITHM AND TWO CLASSICAL ALGORITHMS BASED ON MAP2

Algorithm	Average Planning Time	Path length
A* algorithm	2.5024s	437
Dijkstra algorithm	17.9044s	421
FDCC-PP algorithm	0.5970s	511

However, in spite of the search efficiency is greatly improved, the length of the path planned by the FDCC-PP algorithm is longer than that planned by the A* algorithm and the Dijkstra algorithm. The reason for this result is that the planning strategy of the FDCC-PP algorithm is to move to the target point in the path planning process, which causes the robot to have multiple bypass processes during the movement. However, as for the A* algorithm and the Dijkstra algorithm, the planning strategy of which involves the comparison of the moving costs between nodes, the continuous updating of the relationship between nodes helps the robot avoid bypassing many obstacles, thus reducing the path length to some extent. Based on these two different types of maps, the length of the path planned by the FDCC-PP algorithm is 6.57% and 21.38% longer than the shortest path planned by the Dijkstra algorithm, respectively. Compared with the significant reduction of the time cost of path planning, the length of the planned path is within acceptable range when the real-time requirement for robot path planning is high.

The path planning result of the single robot further demonstrates that the FDCC-PP algorithm can be applied for robot path planning in workshops where the environment changes in real time, especially when there are fewer obstacles in the environment.

B. Path Planning for Multi-robot

In order to illustrate that the FDCC-PP algorithm can be used for multi-robot system, we planned the paths for three robots in the first map, where there are more obstacles and less free space. We assume that the priority of robots from high to low is Robot1, Robot2 and Robot3. As the paths of each robot shown in Fig. 9, it can be seen that there are two overlapping points in the map. However, in the actual movement, Robot2 waits for Robot1 at point (130,149) and Robot3 waits for Robot1 at point (227,230) due to the use of grid-dominant principle, which is explained by their timestamps in Fig. 10, The timestamps of Robot1, Robot2, and Robot3 are labeled with green, blue, and magenta in Fig. 10, respectively.

In Fig 10 (a), according to the path planning strategy for a single robot in FDCC-PP algorithm, both Robot1 and Robot2 will walk to the point (130, 149) at the 70th step, which may cause a collision. However, because of the restriction of the principle of grid-dominance, and the priority of Robot1 is higher than Robot2, Robot2 would wait one step at point (129,

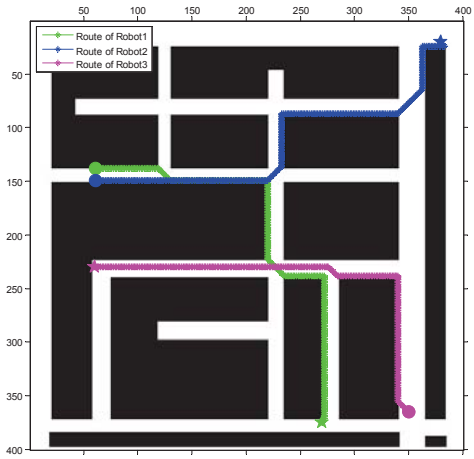

Fig. 9. The result of path planning for three robots.

149) and let Robot1 go first, which ensures the safety of robots. Fig 10 (b) also demonstrates the benefits of using the grid-dominant principle in the FDCC-PP algorithm when it is used for multi-robot path planning.

However, because the path planning processes based on A* algorithm and Dijkstra algorithm both have the comparison of the moving cost from the current position to the start point or the target point, and the parent-child relationship between nodes needs to be updated constantly according to the comparison results. If these algorithms are applied directly to the multi-robot system, not only dose the priority of the robots need to be taken into account, but also we need to consider the real-time adjustment of the moving timestamp of each robot after the parent node updates. Therefore, to be used for multi-robot path planning, the planning strategies of these algorithms need to be adjusted in a complex way, instead of only employing a dominant principle like the FDCC-PP algorithm.

According to the above experiment results, it can be found that the FDCC-PP algorithm is much more efficient than the A* algorithm and Dijkstra algorithm in the path planning for a single robot because of the application of Freeman direction chain code. In addition, it can be improved by only adding the grid-dominant principle for multi-robot path planning, which neither A* algorithm nor Dijkstra can achieve up to now.

VI. Conclusion

In this paper, we propose a Freeman direction chain code path planning (FDCC-PP) algorithm and a grid-dominant principle, which can achieve fast global path planning where the number of robots is not fixed. Through encoding the obstacles in the map by Freeman direction chain code, the robot can query the corresponding obstacle chain code near the obstacle to make a fast detour and continue moving towards the target point, which reduces the search redundancy greatly. Apart from that, the use of the grid-dominant principle makes fast multi-robot path planning possible based on the FDCC-PP algorithm. The statistical results of different experiments illustrate the much better performance of the FDCC-PP algorithm in terms of planning efficiency and practicability when solving the global path planning problem of robots.

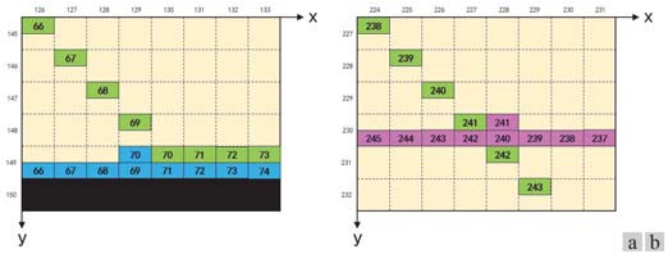

Fig. 10. The timestamps of the robots in their planned paths.

However, it can also be seen that the FDCC-PP algorithm does not hold an advantage from the perspective of path length. Therefore, in future research, we will focus on optimizing the length of the path and further improve the practicability of the FDCC-PP algorithm. In addition, the application of this algorithm to the path planning for robots in practical workshops will also be one of our research directions.

Acknowledgment

This research is supported by Shaanxi Key Laboratory of Integrated and Intelligent Navigation under Grant No. SKLIIN-20180204.

References

[1] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numerische Mathematik*, vol. 1, pp. 269–271, Dec 1959.

[2] R. W. Floyd, “Algorithm 97: shortest path,” *Comm Acm*, vol. 5, no. 6, pp. 345, 1969.

[3] Y. Yuan and D. Wang, “Path selection model and algorithm for emergency logistics management,” *Computers & Industrial Engineering*, vol. 56, no. 3, pp. 1081–1094, 2009.

[4] O. V. Gnana Swathika and S. Hemamalini, “Prims aided flfloyd warshall algorithm for shortest path identifification in microgrid,” in *Emerging Trends in Electrical, Communications and Information Technologies,* pp. 283–291, Springer Singapore, 2017.

[5] Hart P E, Nilsson N J, Raphael B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths [J]. *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100-107, 1968,.

[6] A. Stentz, “Optimal and effifficient path planning for partially known environments,” in *IEEE International Conference on Robotics & Automation*, 1994.

[7] A. Stentz, “The focussed d* algorithm for real-time replanning,” *in Proceedings of the 14th International Joint Conference on Artifificial Intelligence* , vol. 2, pp. 1652–1659, 1995.

[8] G. Mindong, Z. Yani, and Z. Lingyun, “Bidirectional time-effifficient a* algorithm for robot path planning,” *Application Research of Computers*, 2019.

[9] M. A. Contreras-Cruz, V. Ayala-Ramirez, and U. H. Hernandez Belmonte, “Mobile robot path planning using artifificial bee colony and evolutionary programming,” *Applied Soft Computing*, vol. 30, pp. 319–328, 2015.

[10] Y. Fu, M. Ding, and C. Zhou, “Phase angle-encoded and quantum behaved particle swarm optimization applied to three-dimensional route planning for uav,” *IEEE Transactions on Systems Man & Cybernetics Part A*, vol. 42, no. 2, pp. 511–526, 2012.

[11] C. T. Yen and M. F. Cheng, “A study of fuzzy control with ant colony algorithm used in mobile robot for shortest path planning and obstacle avoidance,” *Microsystem Technologies*, vol. 24, no. 1, pp. 1–11, 2016.

[12] M. Nazarahari, E. Khanmirza, and S. Doostie, “Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm,” *Expert Systems with Applications*, vol. 115, pp. 106–120, 2018.

[13] R. C. Gonzalez and P. Wintz, *Digital image processing*. Publishing House of Electronics Industry, 2007.