HTN-based multi-robot path planning

Suying Zeng[1], Yuancheng Zhu[2], Chao Qi[3*]

1. Institute of Systems Engineering, School of automation, Huazhong University of Science and Technology, Wuhan 430074, China
E-mail: zengsuying9960@163.com

2. School of automation, Huazhong University of Science and Technology, Wuhan 430074, China
E-mail: 784787775@qq.com

3. Institute of Systems Engineering, School of automation, Huazhong University of Science and Technology, Wuhan 430074, China
*Corresponding author, E-mail: qichao@mail.hust.edu.cn

Abstract: In this paper, Hierarchical Task Network (HTN) method is applied to multi-robot path planning. A conflict resolution mechanism and time constraint method are also proposed for searching an optimal or approximate optimal collision-free path from start state to target state. Experiment indicates that HTN planning can manage the conflict situations and time constrains well, and provides more optimal route comparing with the traditional A * algorithm.

Key Words: Multi-robot path planning; Hierarchical task network; Collision-free; Time constraint

1 INTRODUCTION

With the advancement of robot technology, the research of multi-robot system become one of the hot spots. As the main research direction, the multi robot path planning is the key to task completion, performance improvement and range extension, which is also the basis of the deeper research in the field of mobile robot. At present, robot path planning algorithms can be divided into two parts, the traditional algorithms and intelligent algorithms. Traditional path search algorithm is based on graph theory or basic mathematical theory, which consists Rapidly-exploring Random Tree (RRT)[1], artificial potential field[2], [3], A * algorithm, etc. As a basic algorithm in the field of artificial intelligence, A * algorithm, which reduce the search space and thus improve search efficiency by heuristic function, has caused a lot of attention of scholars home and abroad and a plenty of research appeared. Intelligent algorithm, inspired by natural phenomenon, consists neural network, genetic algorithm [4], ant colony algorithm, the fuzzy logic control method [5], etc. Hierarchical Task Network (HTN) planning is a popular and widely used intelligent planning method. Compared with the classical planning method, the advantages of HTN planning mainly lies in its ability to reason and efficient representation of domain knowledge. With a great capability of expression, HTN can represent and solve various non-classical planning problems, which can make full use of the domain knowledge to search with high efficiency and handle problems on a large scale, conforming to the characteristics of the path planning problem of mobile robot. However, the application of HTN method in robotic mainly focuses on the single robot navigation [6] [7] and task planning [8], there are few studies on multi-robot path planning.

Therefore, this paper adopts the HTN planning technology to solve the problem of multi-robot path planning, overcoming the challenges of conflict resolution and time constraints. Numerical experiments are conducted to compare with A * algorithm, which prove the effectiveness and availability of HTN planning for robot path planning problem.

2 PROBLEM DESCRIPTION

Robot path planning comprises two parts: space expression and path search. The space expression is to abstract the practical physical spaces of the robot into abstract spaces which can handled by algorithm. The common environment modeling methods includes grid method, visualization method and topological method, free space method, etc.

In this paper, the work space is expressed by grid method, which is easy for storage, handling and display. Mobile robot work environment is represented as a series of binary information grid cell. The space after the division is shown in figure 1.

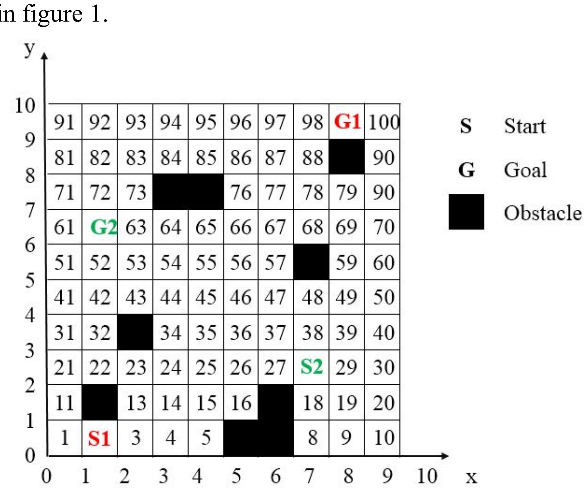

Figure 1 path planning environment in grid map

In this grid map, multi-robot path planning problem can be regarded as: each robot should reach the target point by move and rotate from the starting point, avoiding the obstacles in the environment and other dynamic mobile robots in the process. Make the following assumptions:

(1) Each robot begins at the same time with the same speed.

(2) One grid can only contain one robot;

(3) Mobile robot can move to adjacent eight grids. The attitude angle of the eight grid starts from 0 ° to 360 ° with 45° increase gradually;

(4) The initial direction of robot is direction of increasing x, and the rotation angle is between the next-grid attitude angle and current direction. Robots should rotate in the current grid, then move to the next grid.

3 HTN-BASED MULTI-ROBOT PATH PLANNING

This section describes the problem and domain in HTN Planning, emphatically elaborates the task decomposition methods and operators. In addition, two challenges, the design of conflict resolution mechanism and time constraints, are respectively introduced.

3.1 Problem and Domain in HTN Planning

The planning hierarchy of methods and operators in domain files expresses the planning process clearly, as shown in figure 2. Problem files records the initial task and state, which consists the start and target position of each robot and the coordinate information of grid environment and obstacles. When path planning starts, the priority of robots would configure according to distance length form the start and target location. Then, path planning for robots in priority order until all robots reach the target point. Specifically:

(1) Choose robot: In this method (: method choose-robot). the for-all statement would match the current position of all robots with their corresponding target location. If all robots have arrived the goal location, the program will exit with plan output; otherwise, the robot, with the longest distance from the goal position, will be chose by sort-by statement to plan route by schedule next target grid.

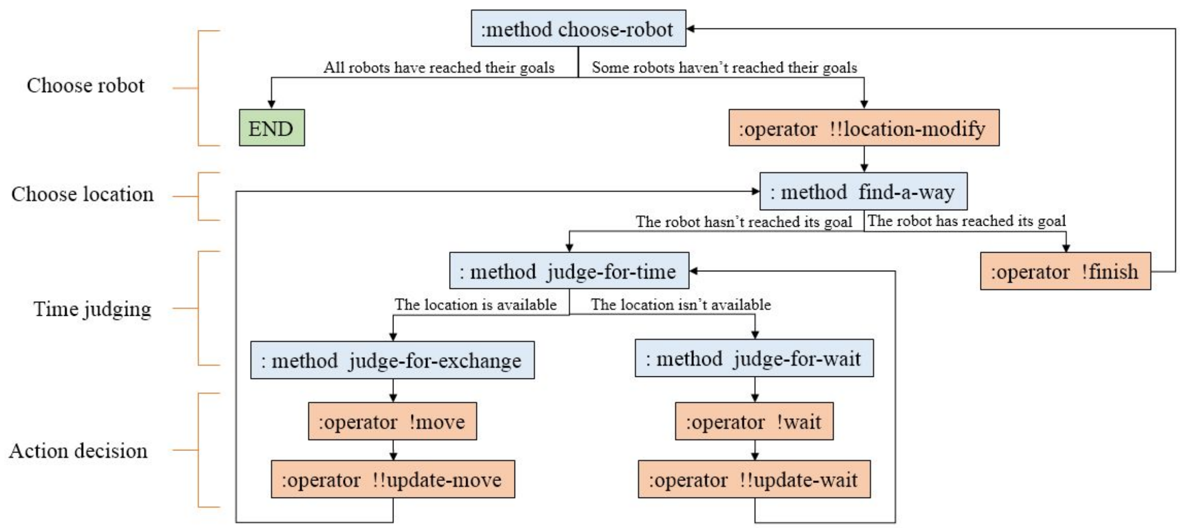

Figure 2. The hierarchy of methods and operators

(2) Choose location: In this method (: method find-a-way? robot), through the coordinate information of grid environment in problem file, the eight grid around the robot could be found. Traverse the eight grids, and mark the grids which obtain obstacles and have been explored by the same robot before. Among the grids which satisfy the two hard constraints, the grid with the smallest heuristic values is the next moving grid. The heuristic information here includes: energy consumption, the distance with the target grid and the path cost.

(3) Time judging. (: method judge-for-time). According to the coordinate information of target grid, the next rotation angle, rotating and moving time should be calculated by the current and the next location. If the next grid is available in the next period of time, the swap position judgment between two robots (: method judge-for-exchange) should be carried on. If not available, the schedule of wait (: method judge-for-wait) should be executed.

(4) Action decision. The operator for wait calculates the optimal wait time for robot in current l0cation. An upper limit for wait is set to avoid waiting too long. The operator for move will update the robot information and resource usage. After moving to next grid, the method for find a path will be carried on.

3.2 The Design and Implementation of Conflict Resolution

In this paper, the conflict problems in multi-robot path planning include the conflict between robot and environment and the conflicts between robots. The flow chart of conflict resolution is shown in figure 3. Figure 3 shows two sections in conflict resolution.

Conflict resolution I handles the collisions between robot and obstacle, which obtain robots reached their target location. In the information list (?location ?x ?y ?z), the variable ?z has three value. When ?z equals 0, the location is free for robot. When ?z equals 1, there is a obstacle in the grid. When ?z equal to a number of robot ?robot, the location is the target of the robot and it has reached. By judging the value of variable ?z, the conflict in this section can be solves.

Conflict resolution I handles the collisions between robot and obstacle, which obtain robots reached their target location. In the information list (?location ?x ?y ?z), the variable ?z is the grid mark, which has three kinds of values. When the grid mark equals 0, the location is free for robot. When the grid mark equals 1, there is an obstacle in the grid. When the grid mark equal to a number of robot, the location is the target of the robot and it has reached. By judging the value of variable the grid mark, the conflict in this section can be solves.

Conflict resolution II handles the collisions between robot and other dynamic robots. The conflicts here mainly include the two cases: swap positions and two robots aiming at the same grid at the same time. In this paper, the priority method and speed regulating method are both used to realize the conflict resolution. As mentioned before, the priority method chooses the robot, which is most remote to its target grid, as the planning object. That is to say, in the planning process, the robot under planning must be the one with lowest priority. Then, the paths of higher priority robots stored in the problem file could all be helpful to the planning. To solve the collision of exchange locations, the operator for move will add sentences to state, representing the leaving time from current grid and the arriving time to next grid respectively. To solve the situation that two robots aim at the same grid in same time, the operator for wait is brought in. The operator for wait will schedule an appropriate waiting time for the robot. After the higher priority robot gone, the current robot could move to the optimal grid. If the wait time is beyond the upper limit, the wait action should be given up to find an available subprime grid.

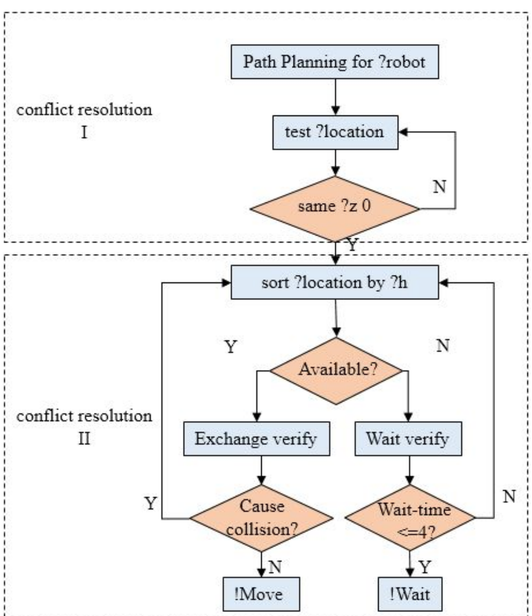

Figure 3. The flow chart of conflict resolution

3.3 Time Constraint Handling

The temporal processing ability of most HTN planners is relatively weak, and the preprocessing technique Multi-Timeline Preprocessing (MTP) is widely used in SHOP2 planner. For each property p that changes over time, MTP modifies the operators to keep track, within the current state, of the times at which the property changes and the times at which various preconditions depend on the property. The idea is that for each dynamic property, the current state will contain two time-stamps: read-time (p), which is the last time that any action read the value of p, and write-time (p), which is the last time that any action modified the value of p. Thus, instead of a single “global time,” the current state will contain many “local times,” namely a read-time and a write-time for each dynamic property. However, this mechanism, which only record the last moments, does not apply to multi-robot path planning problem. Because the MTP processing mechanism will push the whole timeline to next point after a robot action finished, thus, the path planning for other robots will not be able to execute.

So, a feasible time constraint handing mechanism is proposed. Specifically:

(1) At first, a timeline (idle 0 end) is added to every location in the initial state;

(2) In the process of task decomposition, the precondition of occupied the period O[t3,t4] of a grid is, there is a timeline (idle t1 t2) which contains the period O[t3,t4];

(3) If any operators causes the occupation of period O[t3,t4] in that gird, the time line (idle t1 t2)should be deleted and the time line(idle t1 t3) and (idle t4 t2)should all be added to the state.

(4) When robot moves to its next grid, how long will the robot occupy the grid is unknown, because its next rotating time should be calculated next step. So a default rotating time is added for path planning. Then, after operator for wait. The occupied time in current grid should be reset, and after operator for move, the occupied time in current grid should be released and the occupied time in next grid should be added.

4 EXPERIMENTAL ANALYSIS

To compare the HTN method and A * algorithm in the performance of multi-robot path planning problem, a set of experiments are designed. In those eight experiments, there are three variables: the size of grid map, the number of robots and the proportion of obstacle. The parameter setting about time is shown as table 1.

Table1. Parameters Setting about Time

Rotation angle	Rotation time	Direction of movement	Move time
45	0.25	vertical direction	1
90	0.5		
135	0.75	oblique direction	1.5
180	1		

Experiment in each group repeated 10 times randomly, comparing the completion time, total path length and planning time in eight different environment. The results were applied comparative t-test to perform statistical analysis, the confidence level is 95%. The experimental results are shown in table 2.

Table2. The Experimental Results

Size of Map	Proportion of Obstacle	Number of Robots	Completion Time			Total Path Length			Planning Time	
			HTN	A*	Improvement Percentage	HTN	A*	Improvement Percentage	HTN	A*
10*10	5%	5	10.325	12.575	17.893*	28.60856	33.17808	13.773	0.4133	0.0196
		10	11.925	13.45	11.338*	55.2328	62.19067	11.188	0.8813	0.0442
	10%	5	11.5	13.65	15.751	33.54336	36.98234	9.299	0.3862	0.048
		10	13.05	13.5	3.333*	61.24701	66.51199	7.916	0.6986	0.0155
50*50	5%	5	49.5	68.525	27.764	133.2153	154.2022	13.610	24.9921	0.0513
		10	52.6	75.875	30.675	264.0035	268.8889	1.817	57.9506	0.1133
	10%	5	46.75	69.55	32.782	131.0024	134.0925	2.304	22.1716	0.042
		10	50.425	79.225	36.352*	304.2498	317.4267	4.151	64.4955	0.1505

Note：The numbers with * mean that T test results are not significant

The results show that:

(1) for the completion time, the results of 50% experiments shows HTN planning algorithm is significantly superior to A *.

(2) for the total path length, the results of whole eight tests show HTN planning algorithm is significantly superior to A *. To solve the conflict in the same environment, HTN prefer to wait while A * would like choose an alternate path, which caused the increase of the path length. The contrast diagram of path under the same experimental environment is shown in figure 4.

(3) for planning Time due to the different data structure, which A * algorithm is extended by data structure with more flexible, the experimental results show that planning time of HTN is much longer than A* at the same test. And with the increase in grid environment, this disadvantage is more apparent.

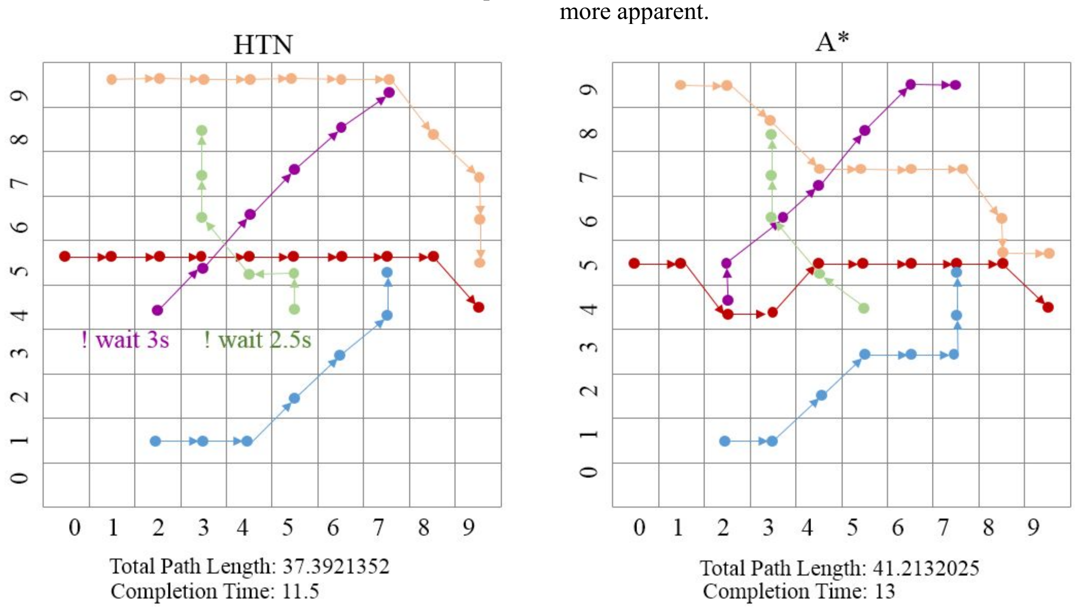

Figure 4. Path contrast between HTN and A*

5 CONCLUSION

Aiming at the multi-robot path planning problem, a new method based on HTN planning is proposed. The planning domain, problem, the hierarchy of method and operators are described briefly. Meanwhile, a set of example were carried on to verify the effectiveness of the proposed method with the comparison of A * algorithm. The work of this paper can be used as the preliminary exploration of the HTN planning to solve the problem of multi-robot path planning. In the future work, the actual situation of more complex, such as dynamic path planning under unknown environment, expanding to a targeted planning method, should be considerate to improve the HTN planning practicality in the field of robot path planning.

ACKNOWLEDGMENT

This work is supported by National Natural Science Foundation of China Grant No. 71371079, Fund of the Hubei Collaborative Innovation Center for Early Warning and Emergency Response Technology No. JD20150103 and Huazhong University of Science and Technology National Independent Innovation Fund Grant No. 201523.

REFERENCES

[1] Yu H, Sharma R, Beard R W, et al. Observability-based local path planning and obstacle avoidance using bearing-only measurements[J]. Robotics & Autonomous Systems, 2013, 61(12):1392–1405.

[2] Montiel O, Orozco-Rosas U, Sepúlveda R. Path planning for mobile robots using Bacterial Potential Field for avoiding static and dynamic obstacles[J]. Expert Systems with Applications, 2015, 42(12):5177-5191.

[3] Serpen G, Dou C. Automated robotic parking systems: real-time, concurrent and multi-robot path planning in dynamic environments[J]. Applied Intelligence, 2015, 42(2):231-251.

[4] Hassanzadeh I, Madani K, Badamchizadeh M A. Mobile robot path planning based on shuffled frog leaping optimization algorithm[J]. Automation Science & Engineering IEEE Conference on, 2010:680-685.

[5] Su K H, Phan T P. Robot path planning and smoothing based on fuzzy inference[C]// System Science and Engineering (ICSSE), 2014 IEEE International Conference onIEEE, 2014:64-68.

[6] Belker T, Hammel M, Hertzberg J. Learning to Optimize Mobile Robot Navigation Based on HTN Plans[J]. Proceedings - IEEE International Conference on Robotics and Automation, 2003, 3(3):4136 - 4141.

[7] Zhang J, Zeng Y, Liu H, et al. MBHP: A memory-based method on robot planning under uncertainties[C]//Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on. IEEE, 2011: 366-371.

[8] Weser M, Off D, Zhang J. HTN robot planning in partially observable dynamic environments[C]//Robotics and Automation (ICRA), 2010 IEEE International Conference on. IEEE, 2010: 1505-1510.