Print ISSN: 1738-3110 / Online ISSN 2093-7717
http://dx.doi.org/10.15722/jds.13.6.201506.11

[Field Research]

An Optimized Random Tree and Particle Swarm Algorithm For Distribution Environments

Zhou Feng*, Un-Kon Lee**

Received: February 19, 2015. Revised: May 10, 2015. Accepted: June 14, 2015.

Abstract

Purpose – Robot path planning, a constrained optimization problem, has been an active research area with many methods developed to tackle it. This study proposes the use of a Rapidly-exploring Random Tree and Particle Swarm Optimizer algorithm for path planning.

Research design, data, and methodology – The grid method is built to describe the working space of the mobile robot, then the Rapidly-exploring Random Tree algorithm is applied to obtain the global navigation path and the Particle Swarm Optimizer algorithm is adopted to obtain the best path.

Results – Computer experiment results demonstrate that this novel algorithm can rapidly plan an optimal path in a cluttered environment. Successful obstacle avoidance is achieved, the model is robust, and performs reliably. The effectiveness and efficiency of the proposed algorithm is demonstrated through simulation studies.

Conclusions – The findings could provide insights to the validity and practicability of the method. This method makes it is easy to build a model and meet real-time demand for mobile robot navigation with a simple algorithm, which results in a certain practical value for distribution environments.

Keywords: Robot, Path Planning, Rapidly-exploring, Random Tree Particle, Swarm Optimizer.

JEL Classifications: D39.

* First Author, College of Electronic Information, Shandong Institute of Commerce and Technology, Ji'Nan, 250103, Shandong China. Doctoral Student in Economic and Business Administration, The University of Suwon, Hwaseong, Korea, Tel: +86-1-50-5318-5007. E-mail: zhoufengkey@163.com.

** Corresponding Author, Professor, School of Economic and Business Administration, The University of Suwon, Hwaseong, Korea. E-mail: snkon@suwon.ac.kr.

1. Introduction

Robot path planning means in the working space, a path is found with the robot starting from a point ,rounding barriers and arriving to the destination. Commonly, there are many paths for robot to accomplish the task, but in fact the best path is selected according to some guide line. These guide lines are :shortest path, least energy consuming or shortest time. So, the robot path planning is a constrained optimization problem. Robot path planning has been an active research area, and many methods have been developed to tackle this problem, such as rolling plan (Hu and Yang, 2004), RRT methods (Guo et al., 2006), neural networks approaches(Qin et al., 2008), GA methods (Sun and Chen, 2005) and ACO algorithm (Zhu, 2006) so on. These works have made some innovative achievements. but the common shortage is the solving time is too long, efficiency is not high. Although Guo et al. (2007) explain that tthe algorithm can achieve rapid optimization, but because a robot go dogleg path in free zone, the path may not be optimal. Grid method is one of the commonly used modeling method, but it has a defect: a robot go dogleg path in free zone, The dogleg is suboptimal paths. Robot arrival time increase.

In this paper, A based on Rapidly-exploring Random Tree(RRT) and Particle Swarm Optimizer (PSO) for path planning of the robot is proposed. First the grid method is built to describe the working space of the mobile robot, then the Rapidly-exploring Random Tree algorithm is used to obtain the global navigation path, and the Particle Swarm Optimizer algorithm is adopted to get the better path. The effectiveness and efficiency of the proposed algorithm is demonstrated by simulation studies.

2. Description of Environment

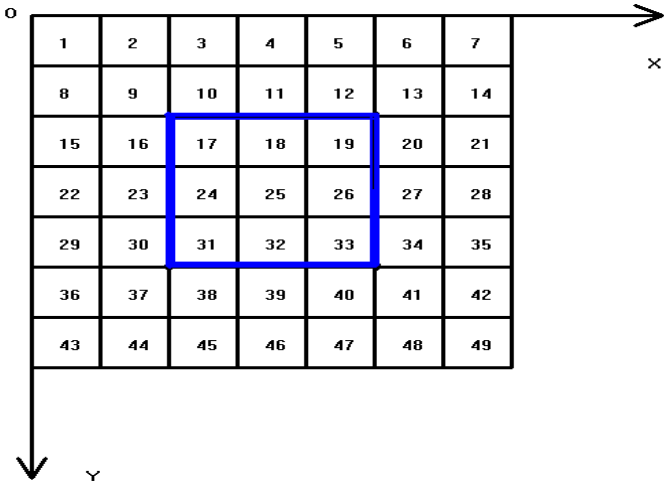
To ensure that the path is not too close to any of the obstacles, the dimension of the robot is represented by a point, and the boundaries of the obstacles are expanded according to the plus of the maximum distance occupied by robot's cross.

Let AS be the finite walking area of Rob in a two-dimension plane and AS is a protruding polygon, in which there are finite numbers of static obstacles b1 ,b2 ,....., bn. The task is to plan a much shorter path for the robot to walk from g_{begin} to g_{end} collision-freely. And Σ_0 is the right-angled coordinate system in AS whose origin is the left and upper corner of AS and its landscape orientation is X-axes, its portrait is Y-axes. The maximal values of x and y are $x_{\max}$ and $y_{\max}$, respectively. Then x and y can be divided usingδas an unit, as a result, all grids can be formed one by one (fig.1). The numbers of each row and column are $N_x = x_{\max} / R_a$ and $N_y = y_{\max} / R_a$, respectively, (if AS is considered as the arbitrary shape, some grid-obstacles can be filled on the boundary of AS to make it square or rectangle), where $b_i (i = 1, 2, \cdots, n)$ holds one or more gratings, when it is not a full one, they are considered as the full one.

The mobile robot environment is represented by orderly numbered grids （C={1,2,3,... , ,M}）, each of which represents a location in the environment. g(x ,y) also represents a location in the environment, x is the row number, y is the columns number.

In the grid based robot environment, the number i and g(x ,y)denote grid and environment coordinate, respectively, thus

xi=((i-1) mod Nx)+1, yi=(int)((i-1)/Nx)+1 (1)

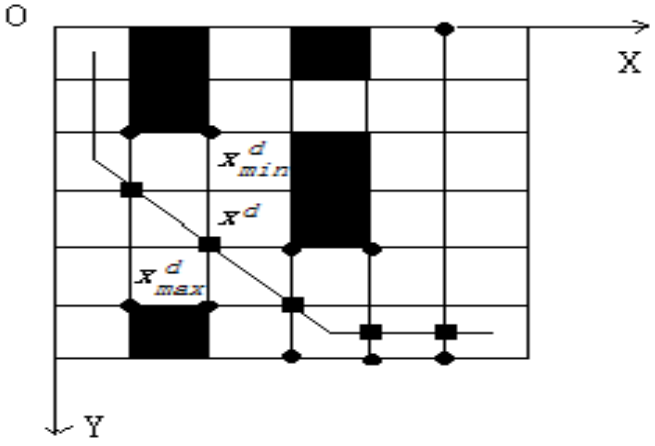


<**Figure 1**> Relationship between grid coordinates and sequence number

3. RRT-PSO

3.1. Algorithm Related Definition

Grid method is simple common modeling methods. But its drawback is that partition size is difficult to control, the grid size is smaller, obstacles are said to be more precise, but at the same time it will take up a lot of storage space; the grid size is too large, the planned path is not accurate. Therefore when planning the precise path, the grid method cannot be used. To resolve the shortage in this paper the Rapidly-exploring Random Tree algorithm is used to obtain the global navigation path, and the Particle Swarm Optimizer algorithm is adopted to adjustment the intersection point of each initial path and the boundary of the grid. Each dimension of each dimension represents a point of intersection, The connection which from low dimension to high dimension will constitute a new path, After several iterations the approximate optimal global path will be finded. The optimization task is to adjust the position of xd to shorten the length of path and get the optimized (or acceptable) path in the planning space. The adjust process of xd is shown in Figure 2. Any xd can slide freely along the free link that it lies on . Path Coding Method new path formed by the slide of xd on line xd min、xd max will not intersect with the obstacles. After processing every xd, the new path nodes sequence forms a new collision free path for the robot.



<**Figure 2**> Path coding method

For conveniently describe, We make the following definition:

Definition 1. The distance between any two grid cells gi and gh (or corresponding points Pi and Ph) is the length of the line between the center points of the two grids and is denoted by d(gi, gh)or d(Pi , Ph). i,h∈C.

$$d(g_i, g_h) = \sqrt{(x_i - x_h)^2 + (y_i - y_h)^2} \quad (2)$$

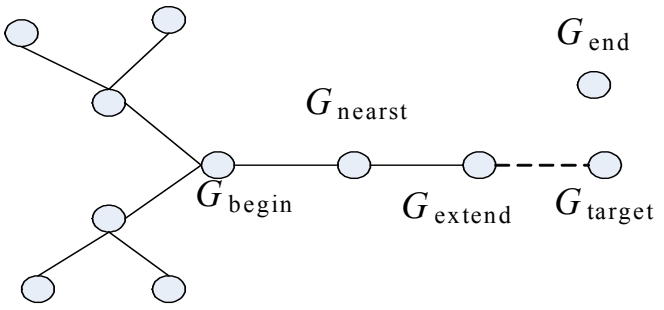
Definition 2. The number of i which is free grid make up a set which was called the feasible region, Marked as F. F∪S=A., i⊆F.

Definition 3. For all nodes in the RRT corresponding grid serial number i consisting of a collection called RRT node set, Marked as P(RRT).

Definition 4. If g is a grid cell, the set NEIBi is called the neighborhood of gi. The broad lines marked the neighborhood of g(4,4) in figure 1.

Definition 5. Collection which is composed by the serial number that the robot is not searched called Q(RRT)

3.2. The Basic Idea and Steps of The Algorithm



<**Figure 3**> The RRT construction

The rapidly-exploring random tree planner is an incremental search algorithm that provides benefits over conventional road-map planners due to the inherent feasibility of the solutions generated. Initially, the Gbegin is the only node of Gend. For each iteration, a random state Grand is chosen, and Gnearst is selected as the nearest state to Grand according to a metric function p. For Gnearst a best input Gbest is chosen to generate a new state Gextend which is closest to Grand of all states generated by applying one step control from Grand. If Grand satisfies the global constraints, Grand will be added to Gend.

Step1: Gbegin as the start point, Gend as the goal point, The grid serial number of Gbegin as RRT root node, initializes relevant parameters.

Step2: Let Gnearst= Groot ,Gneast is the closest node in the P(RRT) to the Gend.

Step3: if Gnearst= Gend, then goto Step6;else goto Step4.

Step4: p which obeys uniform distribution is a random number (p∈¡ô[0,1]). If p is less than pg, then let Gtarge equals Gend. If p is greater than pg, let Gtarget is blank grid which is randomly selected individuals from Q(RRT). pg is known as the constants.

Step5: Find a node Gn($Gn \subset$ P(RRT)),and Gn is the closest to the Gtarget. And then find a node Gextend ($Gextend \subset NEIB_{Gn}$, $Gextend \notin P(RRT)$)which is the closest to the Gn. If Gextend can be find ,it will be added to P(RRT).If d (Gextend, Gend) is less than d(Gnearst, Gend), then let Gnears equals Gextend. Otherwise, the extension failed, go to Step3;

Step6: Return to the formation of the RRT ,a path from Gend to Gbegincan be made by traversing up the tree until the root node is reached.

Step7: Select the best navigation path P={P0P1...¡¦...¡¦Pu+1} , P0 is the start point, Pu+1 is the goal point.

Step8: Calculated the number(Marked as Pnum) of intersection points the navigation path and the vertical boundary of the grid, that is each particle has Pnum dimension, Calculate the sliding range of the first d-dimensional from xd min to xd max.

Step9: Initialize particle c (randomly initialize each dimension of the particle's position $x_C(0)$ and velocity in the solution space $v_C(0)$), each particle's historic optimal position pc is itself. initializes relevant parameters: $\omega_{\max}$ 、 $\omega_{\min}$ 、 c_1 、 c_2 $P\max$ (the largest number of iterations)、n(iteration counter).

Step10: Calculate each article's fitness value according to Eq. (3) and label the particle with the minimum fitness value as Pg;

$$F_c(x(t)) = d(G_{begin}, x_{c,1}) + \sum_{d=1}^{Pnum-1} d(x_{c,d+1}(t), x_{c,d}(t)) + d(G_{end}, x_{c,Pnum}(t)) \quad (3)$$

Step11: Update particle's velocity according to Eq. (4). if $v_{c,d} < -v_{\max}^d$, set $v_{c,d} = -v_{\max}^d$, if $v_{c,d} > v_{\max}^d$, set $v_{c,d} = v_{\max}^d$ r1 and r2 are random number between 0 and 1.

$$v_{c,d}(t+1) = \omega V_{c,d}(t) + c_1 r_1 [p_{c,d}(t) - x_{c,d}(t)] + c_2 r_2 [p_{g,d}(t) - x_{c,d}(t)] \quad (4)$$

Update particle's position according to Eq. (5), if $x_{c,d} < x_{\min}^d$,set $x_{c,d} = x_{\min}^d$. if $x_{c,d} > x_{\max}^d$, set $x_{c,d} = x_{\max}^d$

$$x_{c,d}(t+1) = x_{c,d}(t) + v_{c,d}(t+1) \quad (5)$$

Step12: Calculate each article's fitness value(Fc(x(t+1))) according to Eq. (3). Update pc、Update pg、n=n+1, $\omega = \omega_{\max} - (\omega_{\max} - \omega_{\min}) \times n / p_{\max}$.

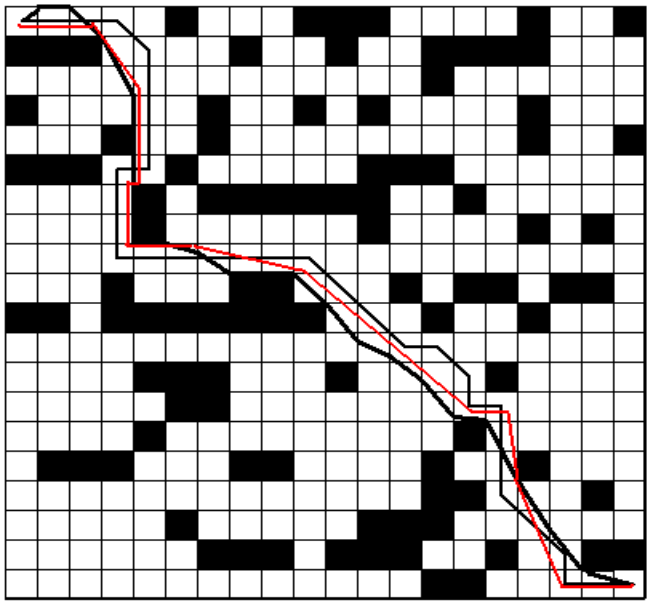
Step13: if n<pmax, turn to step 11 and iterate; if n=pmax turn to step 14.

Step14: find the optimal path.

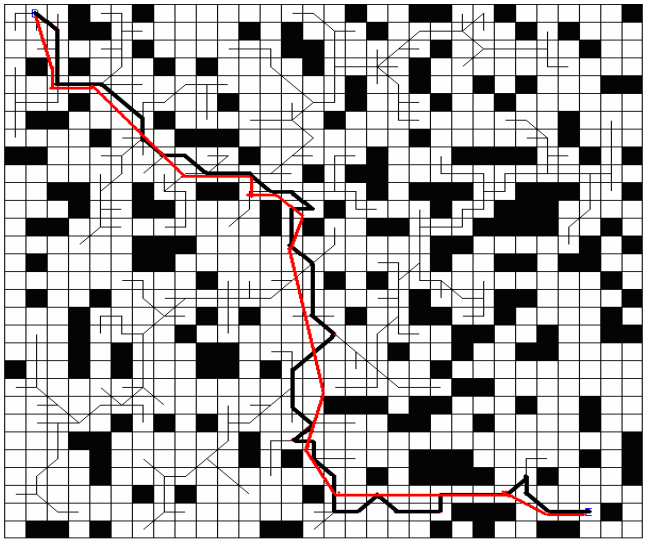
4. Simulation Result

To investigate the effect of the algorithm proposed in this paper, many simulation experiments were conducted. As will be shown in the next sections, the results were quite satisfactory and compare favorably to results using other algorithms. A Pentium desktop computer with Intel Dual Core CPU, 2.33GHz, and 2GB memory was used. The software is written in MS Visual Studio 6.0 C. In order to results were comparable, setting the same experimental environment. Under the same conditions and with the environment of Figure. 4.We compare our results with the results in (Guo et al., 2007). The fine line which length is 32.14 stands for the path of ACO. The bold line which length is 30.12 stands for the path of the algorithm in Guo et al.(2009). The red line which length is 29.57 stands for the path of our algorithm. The grid is used as length unit in this drawing.

In complex environments the Grobot navigation is very successful using our algorithm. We compare our results with the results in Guo et al. (2007). The black line which length is 32.13 stands for the path of the algorithm in Guo(2007). The red line which length is 31.25 stands for the path of our algorithm.



<Figure 4> The path in a simple environment



<Figure 5> The path in a complex environment

In order to further validate the algorithm, We compared our algorithm with others in the 30 * 30 environment of reference (Zhang et al., 2009).In the table 1 are listed the average length of path which obtained in five tests.

<Table 1> Performance Contrast

environment	A*	GA	ACO - grid	RRT-grid[7]	reference [9]	our algorithm
30*30[8]	N/A	85.491	65.0	61.2	45.0	44.5

5. Conclusion

In this paper, the authors propose a novel method to solve the global path-planning problem for the mobile robot. First the grid method is built to describe the working space of the mobile robot, then the Rapidly-exploring Random Tree algorithm is used to obtain the global navigation path, and the Particle Swarm Optimizer algorithm is adopted to get the better path. Experiment results show the validity and practicability of the method. By this method, it is easy to build a model and meet the real-time demand for mobile robot's navigation with the simple algorithm, which results in a certain practical value in distribution environments.

The results also shows that combine the Rapidly-exploring Random Tree algorithm and the Particle Swarm Optimizer algorithm ,with its high efficiency and flexibility, not only handles a single mobile manipulator well in dynamic environments with various obstacles of unknown motions in addition to static obstacles, but can also readily and effectively plan motions for each mobile manipulator in the distributive environment shared by multiple mobile manipulators and other moving obstacles.

References

Cai, Wenbin, & Zhu, Qingbao (2009). Rolling Path Planning of Robot Based on Rapidly Exploring Random Tree in an Unknown Environment[J]. *Journal of Nanjing Normal University: Engineering and Technology Edition,* 6(2), 79-83.

Guo, Haitao, Zhu, Qingbao, Xu, Shoujiang, & Zhou, Feng (2007). Rapid-exploring random tree algorithm for path planning of robot based on grid method[J]. *Journal of Nanjing Normal University: Engineering and Technology Edition,* 7(2), 58-61.

Guo, Haitao, Zhu, Qingbao, & Si, Yingtao (2009). Novel Path Planning for Robots Syncretizing Ant Algorithm and Genetic Algorithm[J]. *Journal of Chinese Computer Systems,* 29(10), 1838-1841.

Guo, Tongying, Qu, Daokui, & Dong, Zaili (2006). Research of Path Planning for Polishing Robot Based on Improved Genetic Algorithm[C]. Proceedings of the 2004 IEEE International Conference on Robotics and Biomimetics, 334-338

Hu, Yanrong, and Yang, Simon X. (2004). A Knowledge Based Genetic A1gorithm for Path Planning of a Mobile Robot[C]. Proceedings of the 2004 IEEE international Conference on Robotics & Automation NewOrleans,200, 4350-4355.

Qin, Yuan-Qing, Sun, De-Bao, & Zhou, Feng (2008). Path Planning For Mobile Robot Using The Particle Swarm Optimization With Mutation Operator[C]. Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai, 26-29 August 2008, 2473-2478.

Sun, Bo, & Chen, W. (2005). Particle Swarm Optimization Based Global Path Planning for Mobile Robots[J]. *Control and Decision,* 20(9), 1052-1060.

Zhang, Meiyu, Huang, Han, & Hao, Zhifeng (2008). Motion Planning Of Autonomous Mobile Robot Based On Ant Colony Algorithm[J]. *Computer Engineering and Applications,* 41(9), 34-37.

Zhu, Qing-Bao (2006). Ant Algorithm for Navigation of Multi-Robot Movement in Unknown Environment[J]. *Journal of Software,* 17(9),1890-1898.