A Global Path Planning Approach Based on Particle Swarm Optimization for a Mobile Robot

QIAORONG ZHANG SHUHONG LI
Institute of Information
Henan University of Finance and Economics
No.80,Wenhua Road,ZhengZhou, Henan,P.R.. China,450002

Abstract: - A new global path planning approach based on particle swarm optimization (PSO) for a mobile robot in a static environment is presented. Consider path planning as an optimization problem with constraints. The constraints are the path can not pass by the obstacles. The optimization target is the path is shortest. The obstacles in the robot's environment are described as polygons and the vertexes of obstacles are numbered from 1 to n. The particle swarm optimization is used to get a global optimized path and the particle is defined as a set which is composed of zero or nozero numbers from 1 to n. Simulation results are provided to verify the effectiveness and practicability of this approach.

Key-Words: - Particle swarm optimization (PSO), Swarm intelligence, Global path planning, Mobile robot, Static known environment

1 Introduction

Global path planning is one of the key technologies of mobile robots, and it represents the intelligence level of mobile robots in some ways. The task of the global path planning is to find a path from the start position to the destination under a static environment where the obstacles are known.

Up to now, many different approaches have been used to solve the path planning problem. *Artificial potential field* approaches[1][2], based on following the gradient of generated potential field lines in an environment, may run into local minima which lead to no optimal solutions or may even trap the robot. *Free space networks* create a directed graph based on some property of the environment. The graph is then searched with one of many possible heuristic algorithms. But this approach lacks flexibility. *Cell decomposition* approaches consist of subdividing an environment into discrete cells of a predefined shape and size and then searching an undirected graph based on the adjacency relationships [3]. But this method has the contradiction between the precision and the memory consumption.

The path planning of a mobile robot can be considered as an optimization problem because the path can be described by some evaluation functions such as the shortest distance and avoiding the collision with the obstacles. Evolutionary computational methods such as genetic algorithm (GA) are used to solve the optimization of path planning [4]-[7]. Inspired by social behavior in the nature of ants or birds are developed to solve optimization problems and path planning. The particle swarm optimization (PSO) was developed in 1995 and now it becomes a very important method for solving optimization problems including path planning [8]-[12].

In this paper a new path planning approach based on the particle swarm optimization (PSO) is presented. Firstly, the obstacles in the robot's environment are described as polygons. Then the model of the vertex of obstacles is built to describe the workspace of the mobile robot and the particle swarm optimization is used to get a global optimized path.

The rest of the paper is organized as follows. Section 2 provides the statement of the problem. An overview of the path planning approach is given in section 3. The details of the approach are found in section 4. Section 5 provides simulation results and analysis. Conclusions are found in section 6.

2 Problem statement

Without loss of generality, it is supposed that the robot moves in the workspace as described below:

- The robot moves in a limited two-dimensional space;
- The robot can be considered as a particle if the boundary of each obstacle is extended by the half size of the robot's maximal dimension in length or width direction;

- The obstacles in the robot's workspace are described as polygons;
- The vertexes of each obstacle in the workspace are numbered from 1 to n (n is the amount of the obstacle's vertexes in the workspace).

Fig.1 shows a representation of a sample environment. The obstacles are described as rectangles in the map for simplicity and clarity. In fact the obstacles can be any polygon shape. The vertexes are numbered from 1 to 16.

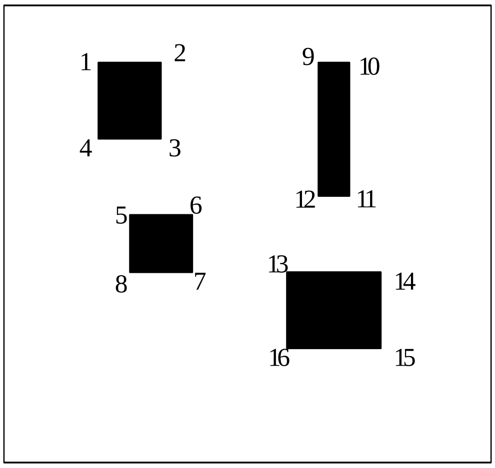

Fig.1. A representation of a sample environment

Now the target of the path planning is to find a set of vertexes which can connect the start position and the destination position. So the path planning can be converted to a optimization problem with restriction which is to find a set P={S, P1, P2….G}. Here S is the start position, Pi is the vertex of the obstacle and G is the destination position. The target is to minimize the sum of the distances between the points of the set. The restriction is when the robot moves along the path, it can avoid collisions with any obstacles.

3 Path Planning Approach Based on Particle Swarm Optimization

Particle Swarm Optimization (PSO) was developed in 1995 by James Kennedy and Russ Eberhart and emerged from earlier experiments with algorithms that modeled the "flocking behavior" seen in many species of birds. It has been applied successfully to a wide variety of search and optimization problems.

The Particle Swarm Optimization (PSO) can be described as follows.

Assume that the dimension of swarm space is D, and a swarm includes N particles. The location of particle i is defined as $X_i=(x_{i1},x_{i2},x_{i3}\ldots x_{id})$. The best location that the particle i has searched is defined as $pBest=(p_{i1},p_{i2},p_{i3}\ldots p_{id})$. The best location that the particle swarm has searched is defined as $gBest=(g_{i1},g_{i2},g_{i3}\ldots.g_{id})$. The updating principle for individual particle is defined

$$v_{id}(t+1)=v_{id}(t)+c1\times rand()\times(p_{id}(t)-x_{id}(t))$$
$$+c2\times rand()\times(g_d(t)-x_{id}(t)). \quad (1)$$

$$x_{id}(t+1)=x_{id}(t)+v_{id}(t+1) \quad . \quad (2)$$

where c1 and c2 are positive constants; rand() is random variable; p_{id} reprents the best position that particle i has found so far; g_d reprents the best position that the particle swarm has found so far. The initial position and velocity of the particle swarm are generated randomly and updated using Eq. (1) and Eq. (2) until the best solution is obtained.

The path planning approach based on PSO is listed as:

Step 1: Initialize each particle and its velocity randomly.

Step 2: Calculate the fitness value of each particle.If the fitness value is better than the best fitness value (pBest) in history, set current value as the new pBest.

Step 3: Choose the particle with the best fitness value of all the particles as the gBest.

Step 4: Calculate particle velocity according equation (1).

Step 5: Update particle position according equation (2).

Step 6: Go step 2 until the maximum iterations is attained or the fitness is small enough (convergence).

4 Key Technologies

4.1 Particle representation

The direct encoding scheme is applied to encode the individual particle. Firstly, the obstacles in the environment are described as polygons. Secondly, the vertexes is numbered from 1 to n. Here n is the amount of the vertexes of the obstacles. Then the dimension of each particle is set as equal to n. Each element in the dimension is either "0" or a nonzero number in [1,n]. The nonzero number in an individual particle indicates the vertex that the path passes by. The sequence that the nonzero number appears represents the sequence that the vertex the path passes by. To avoid the path passes one vertex repeatedly, the nonzero number in the individual particle is not permitted repeatedly. For example, a particle's current position is (0,0,5,0,0,0,3,0,0,9,0,0,0,0,0,0). It represents the path: start→vertex 5→vertex 3→vertex 9→destination, which is shown in Fig. (2).

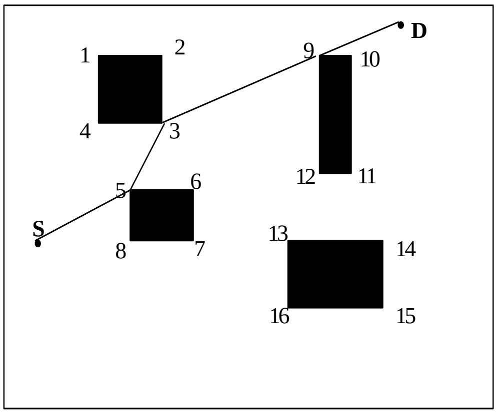

Fig.2. An example of the particle representation

4.2 Initialization

The initialization process is described as follows.

Step 1: Initialize each particle's velocity randomly.

Step 2: Let each element in a particle equal zero.

Step 3: Decide each element is zero or nonzero at the probability 0.5. If the element is decided to be nonzero, then choose a number in [1,n] randomly. To avoid a vertex appears in the path repeatedly, the chosen number must be different from the other nonzero numbers exist in the individual particle already.

4.3 Fitness function

The fitness function is an important factor for the convergence and the stability of particle swarm optimization (PSO). The collision avoidance and the shortest distance should be considered in path planning. The sum of each evaluation function weight is a typical method to construct the fitness function, but it is prone to be instable and its weight coefficients are difficult to be tuned and to be determined because they are changing when the path and the obstacles change. So when the fitness function is constructed, the number of evaluation functions is as small as possible. On the other hand, the two evaluation functions, the collision avoidance and the shortest distance, must be integrated into a fitness function.

Collision avoidance is essential to path planning and makes the mobile robot travel in the workspace safely. Because the path is composed of the vertexes of the obstacles, the start position and the destination position, it must be guaranteed that the line connecting every two adjacent points must not pass by the obstacles. So a chastening function which is used to ensure that the path does not pass by any obstacle is defined as

$$f1 = 1/N \quad . \tag{3}$$

where N is the number of the lines which connect the adjacent vertexes in the individual particle and do not pass by the obstacles. Eq. (3) implies that the more is the number of the lines which connect the adjacent vertexes in the individual particle and do not pass by the obstacles the smaller is the fitness value.

In addition to collision avoidance, the path can be optimized for minimum distance whose fitness function can be described by

$$f2 = (1 + 1/\sqrt{n-1})D$$
$$D = \sum_{i=1}^{d} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \quad . \tag{4}$$

where n is the number of the vertexes in the individual particle, D is the sum of the direction distance between the vertexes in the individual particle, and (x_i, y_i) is the coordinate of the vertex i in the individual particle.

Eq. (4) implies that the path can be optimized for minimum distance. But if the path is optimized only by the distance D, the individual (source, goal) whose fitness value is the shortest and such individual will increase after several iterative computations. So $1/\sqrt{n-1}$ is used in the fitness function to eliminate the individual such as {source, goal}.

Thus the final fitness function is constructed as shown in (5).

$$f = (f1 \times f2)/n \quad . \tag{5}$$

where n is the number of the vertexes in the individual particle. n is used to ensure not to much vertexes in the path because the more is the vertexes in the path the bigger is the possibility the path passes by the obstacles. When the fitness function reaches the minimum, the global optimal path is found. This not only makes computation simple but also overcomes the disadvantage of the instability from the summation of evaluation function weights.

4.4 Intersection estimating algorithm

The intersection estimating algorithm is to find whether the lines connecting every two adjacent vertexes intersect with the obstacles.

The line connecting two adjacent vertexes in the path can be defined as

$$a_{ij}x + b_{ij}y + c_{ij} = 0 \quad . \tag{6}$$

where the values a_{ij},b_{ij} and c_{ij} can be computed by

$$\begin{aligned} a_{ij} &= ver_i.y - ver_{i+1}.y \\ b_{ij} &= ver_{i+1}.x - ver_i.y \\ c_{ij} &= ver_i.y * ver_{i+1}.x - ver_{i+1}.y * ver_i.x \end{aligned} \quad . \quad (7)$$

where ver_i and ver_{i+1} represent the vertex i and the vertex i+1 in the path. x and y means the coordinates of the vertexes.

The intersection estimating algorithm is defined as follows.

1) If all the vertex coordinates of the obstacle satisfy $a_{ij}x + b_{ij}y + c_{ij} < 0$, then the obstacle is on the left of the line.
2) If all the vertex coordinates of the obstacle dissatisfy $a_{ij}x + b_{ij}y + c_{ij} > 0$, then the obstacle is on the right of the line.
3) If some vertex coordinates of the obstacle satisfy $a_{ij}x + b_{ij}y + c_{ij} < 0$, and some vertex coordinates of the obstacle satisfy $a_{ij}x + b_{ij}y + c_{ij} > 0$, then the obstacle intersect with the line.

Fig.3 shows three examples corresponding to the three conditions mentioned above. L1 is a line and polygon a, b, c are three obstacles. Obstacle a that is on the left of the line satisfies condition 1) mentioned above. Obstacle b satisfies condition 2) mentioned above and it is on the right of the line c satisfies condition 3) mentioned above and it intersects with the line.

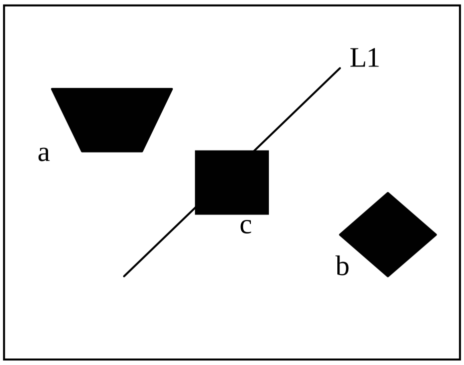

Fig.3. Examples of Intersection estimating algorithm

5 Simulation results

The proposed algorithm has been implemented on a Intel Pentium 4, 3.00 GHZ computer. The simulation environment is described as follows.

- For simplicity and clarity, the obstacles in the robot's workspace are rectangles. In fact, the shape of the obstacles can be random polygons.
- The boundary of each obstacle is extended by the half size of the robot's maximal dimension in length or width direction. So the robot can be considered as a point.
- The swarm used in PSO has 40 particles. The parameters used in PSO are defined as:c1=c2=2, w=0.7968, v is the random velocity which is limited within [-n,n]. Here n is the number of the vertexes of the obstacles.
- The iterations for path planning duration are identically 500 iterations.

The simulation results are shown in Fig. 4 and Fig. 5.

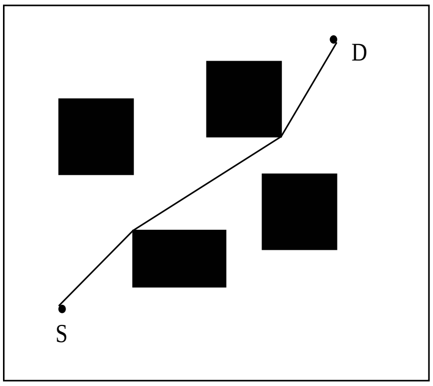

Fig.4. Simulation result of workspace with sparce obstacles

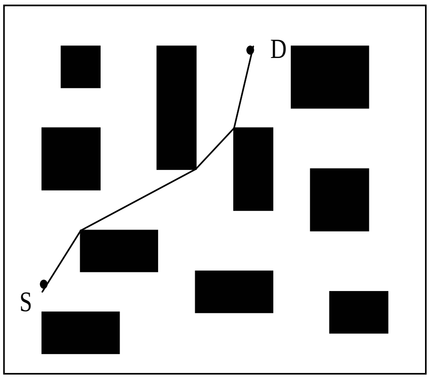

Fig.5. Simulation result of workspace with dense obstacles

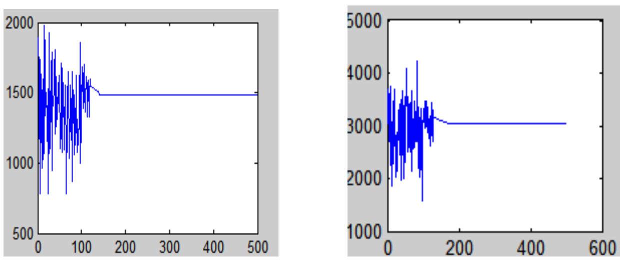

Fig. 6. Evolution of the path planning process

Fig.6 shows the evolutionary process about the two results mentioned above and obviously every optimization process is convergent. This means in every time of path planning all particles of the swarm converge to an optimal value which guarantee a safety path avoiding obstacle and connecting the start and the destination. Because the value of the fitness can not be optimized to zero, the path may be not the global best path.

6 Conclusions

In this paper, a PSO-Based global path planning approach for a mobile robot in a static environment is presented. Firstly, The obstacles in the robot's environment are described as polygons and the vertex of obstacles are numbered from 1 to n , the number of the obstacles's vertexes, to describe the workspace of the mobile robot. Secondly, the particle swarm optimization is used to get a global optimized path. From the Simulation results , it can be concluded that this approach is effective and practicable. In the future work, the parameters used in the PSO can be adjusted and the crossover and mutation operation can be used to improve the convergent speed and avoid the local minimum problems.

Acknowledge

This work is supported by Science Technology Project of Henan Province of China under Grant No.0624260019, No.072102210001 and Natural Science Foundation of Department of Education of Henan Province of China under Grant No.2004922081.

References:

[1] Yoshifumi Kitamura, 3-D Path Planning in a Dynamic Environment Using an Octree and an Artificial Potential Field, *Proc. 1995 IEEE International Conference on Intelligent Robots and Systems*, pp. 474-481

[2] M.G. Park and M.C.Lee, Experimental Evaluation of Robot Path Planning by Artificial Potential Field Approach with Simulated Annealing, *Proc. of the 41st SICE Annual Conference*, Vol. 4. Osaka, Japan, August 2002,pp. 2190-2195

[3] Brooks R.A, Lazano-Perez, A Subdivision Algorithm in Configuration Space for Finding Path with Rotation, *8th International Conference on Artificial Intelligence*, 1983, pp. 799-806

[4] Liu Yanfei, Robot Path Planning Based on Genetic Algorithms with Two-Layer Encoding, *Control Theory and Applications*, Vol. 17, No. 3, 2000,pp.429-432

[5] DU Xin, CHEN Hua-hua, GU Wei-kang, Neural network and genetic algorithm based global path planning in a static environment, *Journal of Zhejiang University Science,* Vol. 6, No. 6,2005, pp. 549-554

[6] Y.hu and S.X.Yang, A Knowledge Based Genetic algorithm for Path Planning of a Mobile Robot, *Proc. of IEEE Int. Conf. on Robotics and Automation*, New Orleans, LA, Vol. 5, 2004,pp.4350-4355

[7] W.Tao, M. Zhang, and T.J. Tarn, A Genetic Algorithm Based Area Coverage Approach for Controlled Drug Delivery Using Micro-Robots, *Proc. of the IEEE Int. Conf. on Robotics and Automati*on, New Orleans, LA, Vol. 5, 2004,pp.2086-2091

[8] R.C.Eberhart and J.Kennedy, A New Optimizer Using Particle Swarm Theory, *Proc. of the 6th Int. Symp. On Micro Machine and Human Science*,1995,pp.39-43

[9] J.Kennedy and R.C.Eberhart, Particle Swarm Optimization, *Proc. of IEEE Int. Conf. on Neural Network*, 1995, pp.1942-1948

[10] Y.Q. Qin, D.B. Sun, M. Li and Y.G. Cen, Path Planning for Mobile Robot Using the Particle Swarm Optimization with Mutation Operator, *Proc. of Int. Conf. on Machine Learning and Cybernetic*s, Vol. 4, 2004,pp.2473-2478

[11] Y.Li and X. Chen, Mobile Robot Navigation Using Particle Swarm Optimization and Adaptive NN, *Advances in Natural Computation*, 2005,pp.475-482

[12] S.Doctor and G.K.Venayagamoorthy, Unmanned Vehicle Navigation Using Swarm Intelligence, *Proc. of Int. Conf. on Intelligent Sensing and Information Processing*, 2004,pp.249-253