

A New Method for Mobile Robot Path Planning based on Particle Swarm Optimization algorithm

Ning Qiang, Jie Gao*

School of Physics and Information Technology, Shaanxi Normal University, Xi'an 710119, China

qn315@snnu.edu.cn, gaojie@snnu.edu.cn(corresponding author)

Keywords: robot path planning, particle swarm optimization, cubic spline, initialization strategy

Abstract: Particle swarm optimization (PSO) algorithm has been widely applied to the mobile robot path planning problems, but it still has some shortcomings such as high coding dimension and premature convergence. This paper proposes a new PSO algorithm combined with cubic spline functions. Firstly, several path nodes are set up in the environment, and then use the cubic spline functions to interpolate on the path of the starting point, path nodes and target point, thus obtain a full path which formed by connecting all interpolation points. The particle coding is just the coordinates of the path nodes, which significantly reduces the coding dimension of the particle. In order to further improve the efficiency of this algorithm, a new initialization strategy and penalty function based fitness are designed. The proposed method shows excellent performance in computational experiments in terms of solving quality, stability and speed.

Introduction

Robot global path planning method mainly involves two key problems: map modeling and search strategy^[1]. At present, the main global path planning methods include visibility graph, A* algorithm, grid method and search algorithm based on intelligent optimization^[2]. Intelligent optimization algorithms have become a research hotspot in solving robot path planning problems. Multiple intelligent optimization algorithms have been successfully applied in search strategy of path planning, including genetic algorithm (GA), colony algorithm (CA) and particle swarm optimization (PSO). PSO has drawn extensive concern by virtue of advantages like high convergence rate and less regulating parameters^[3].

Literature [4] firstly proposed path planning method based on PSO algorithm and cubic spline function. The particles coding is parameter of Ferguson spline function, and the solution path consists of several Ferguson curves. As each spline curve has 4 parameters, coding dimension of this method is still high. Path planning method designed in this paper, on basis of PSO algorithm and cubic spline interpolation, takes coordinates of path nodes in the environment as particle coding, which significantly lowered coding dimension, because in most cases only 3 or 4 path modes are needed to avoid obstacles. Simulation results indicated effectiveness of the algorithm in this paper.

Proposed path planning method

The proposed algorithm (named PSO-SP) in this paper take path node as particle coding. The number of path nodes represents number of cubic spline curves. As spline curve is first-order continuous and is second-order continuous at path nodes, number of path nodes represent maximum times for path to turn. Even under extremely complicated environment, all obstacles could be bypassed through turning for 3 to 5 times, so particle coding dimension would not increase by a large margin as environmental complexity enlarged. A series of path points were obtained by conducting interpolation of starting point, path nodes and terminal point through cubic spline function, and one path is obtained by connecting these path points.

Illegal solutions would be generated during PSO algorithm iteration process, namely paths passing through obstacles. Traditional methods mainly used correction method or penalty function to eliminate illegal paths when disposing illegal paths, which actually reduced efficiency of PSO

algorithm. If there were too many illegal paths, PSO algorithm would seriously degrade, because there were not enough feasible solutions to maintain evolutionary swarm. Even if illegal solution was corrected in each iteration, the algorithm could not continuously conduct effective optimizing, as new illegal solution would be generated after each iteration. In practical problems, one illegal path could approximate optimal path. For example, one part of path only contacted with obstacle in a small range, that mean only slight adjustment was needed to obtain legal high-quality path. That indicated even illegal paths could be evolved into legal optimal solutions. Hence, “degree” of path illegality was defined in this paper, fitness function based on penalty function was designed and it reserved those paths with good fitness value and low illegality degree, thus reserving more particles in evolutionary swarm and improving operating efficiency of the algorithm.

Particle coding method. Particle coding is about coordinate position of several path nodes. Number of path nodes decide maximum times for cubic spline curve to bend. Under general circumstance, one path could bypass all obstacles by bending for 3-5 times. Position of path nodes could be randomly selected or selected according to certain strategies. The following would firstly introduce particle coding design. It’s assumed that coordinates of m path nodes were determined, respectively being $r1=(xr1,yr1), \dots, rm=(xrm,yrm)$, coordinates of starting point and terminal point of path were respectively $(x0,y0)$ and $(xn+1,yn+1)$. Coordinates $(x1,y1), \dots, (xn,yn)$ of n interpolating points were obtained through cubic spline interpolation on intervals $[x0,xr1, \dots, xrm, xn+1]$ and $[y0, yr1, \dots, yrm, yn+1]$. Path corresponding to particle coding consisting of m path nodes was a ligature consisting of interpolating points $(x0,y0), (x1,y1), \dots, (xn,yn), (xn+1,yn+1)$.

Initialization strategy. Random initialization strategy is generally used in particle swarm optimization. But for the proposed algorithm, if path nodes are randomly selected, it would probably result in path illegality. It is easy to imagine that path nodes are inside the obstacles, which would certainly result in path illegality. So a new initialization strategy is designed in this paper which could not only guarantee path legality but also could obtain optimal initialization path. The initialization strategy is as below:

Step 1: Connect starting point and terminal point and record it as line segment L .

Step 2: Take m points uniformly on line segment L , and coordinates of each point were definite and recorded as:

$$x_i = \frac{x_{n+1} - x_0}{m+1} i, y_i = \frac{y_{n+1} - y_0}{m+1} i, i = 1, 2, \dots, m \quad (1)$$

Step 3: Judge whether line segment L intersected with obstacle. If they intersected with each other, then solve the intersection interval between line segment L and obstacle.

Step 4: Judge whether m points were in the intersection interval between line segment L and obstacle. If not, values of path nodes are randomly taken nearby equational nodes, namely:

$$xr_i = Rand[0.5 \frac{x_{n+1} - x_0}{m+1} i, 1.5 \frac{x_{n+1} - x_0}{m+1} i], i = 1, 2, \dots, m \quad (2)$$

$$yr_i = Rand[0.5 \frac{y_{n+1} - y_0}{m+1} i, 1.5 \frac{y_{n+1} - y_0}{m+1} i], i = 1, 2, \dots, m \quad (3)$$

If yes, then draw a vertical line to line segment L by passing a node on intersection interval between line segment L and obstacle. Next, find a point which is nearest to the node on this vertical line and not on the obstacle, namely intersection point between the vertical line and obstacle, and randomly select a point nearby this point as path node. It’s assumed that for linear equation $y = k_1 x + b_1$ decided by starting point and terminal point, slope k_1 and constant b_1 could be solved by substituting coordinates of starting point and terminal point. It’s assumed that an equational point (x_i, y_i) fell on intersection interval between line segment L and obstacle, then a vertical line passing line segment L was drawn, then vertical line equation was $y = k_2 x + b_2$, then $k_1 \cdot k_2 = -1$, and vertical line could be obtained by

substituting point (x_i, y_i) , coordinates of this vertical line and obstacle were solved and recorded as (x'_i, y'_i) . At last, coordinates of path node could be selected according to the following formula:

$$x'_i = \text{Rand}[x'_i - 0.5 \frac{x_{n+1} - x_0}{m+1}, x'_i + 0.5 \frac{x_{n+1} - x_0}{m+1}], i = 1, 2, \dots, m \quad (4)$$

$$y'_i = \text{Rand}[y'_i - 0.5 \frac{y_{n+1} - y_0}{m+1}, y'_i + 0.5 \frac{y_{n+1} - y_0}{m+1}], i = 1, 2, \dots, m \quad (5)$$

From the above steps, path nodes which were near to line segment L as much as possible and under uniform distribution could be obtained. But as range of random selection was large, path nodes would probably fell inside the obstacle, thus resulting in path illegality. If path nodes were inside the obstacle, then path nodes would be randomly re-selected within corresponding scope until all path nodes were not inside the obstacle.

Fitness based on penalty function. During evolution process, particles continuously sought for route with the shortest length and weed out paths intersecting with the obstacle. To simplify calculation, it's only necessary to judge whether all path points namely interpolating points were inside the obstacle. In reality, the situation in which two endpoints of line segment were outside the obstacle but line segment intersected with the obstacle might occur. If there was a large number of interpolating points, then cubic spline curve would be smoother, while each line segment on the path would be shorter, so illegal paths under this situation could be neglected. Fitness value and illegality degree of each path could be defined according to number of inside interpolating points inside the obstacle:

$$F = L(1+V)(1+0.1t) \quad (6)$$

$$V = \frac{n_o}{n+2} \quad (7)$$

Formula (6) is fitness function with penalty function, whereby F represented fitness value, L represented path length, V is illegality degree of path, and t is number of iterations for illegality continuing to enlarge. If illegality degree lowered, then t experienced zero clearing. It could be seen that if illegality degree is low, penalty value added to illegal path is small, illegal paths have the opportunity to evolve into optimal legal paths. But if illegal degree of illegal paths didn't lower as number of iterations increased, then fitness value would increase by 10% after each iteration. If penalty value added to a path is large, then this path would be automatically weeded out during algorithm evolution process. Formula (7) is definition of illegality degree, n_o represented number of interpolating points falling inside the obstacle, and $n+2$ was number of interpolating points including starting point and terminal point.

Computational experiments

Under a complex testing environment, PSO-SP algorithm and PSO algorithm in Literatures [4] were tested. Particle coding dimension of PSO-SP algorithm was set as 3. Number of interpolating points was set as 50. Particle swarm scale N was set as 80, inertia factor and learning factor were set as: $w = [0.9, 0.4]$, $c_1 = 1.5$, $c_2 = 1.5$. Maximum value of particle speed was limited to $v_{\max} = 6$. Particle coding Dimension of PSO in Literatures [4] was set as 16, so number of spline curves was identical with that of PSO-SP algorithm, but coding dimension of PSO-SP algorithm was lower. Fig 1 shows the best 3 paths generated by PSO-SP and PSO[4] for 30 times of operation respectively. It could be seen from Fig 1 that path acquitted by the two algorithms are very smooth, and this is characteristic of cubic spline interpolation curve. It could also be seen from the figure that PSO-SP found the global optimal path with path length being 141.7652, while PSO[4] only found 3 local optimal paths with short distance. Table 2 displayed statistical results of PSO[4] and PSO-SP for 30 times of operation. It is obvious from the table that PSO-SP had superior performance in aspects of convergence speed,

convergence property and stability.

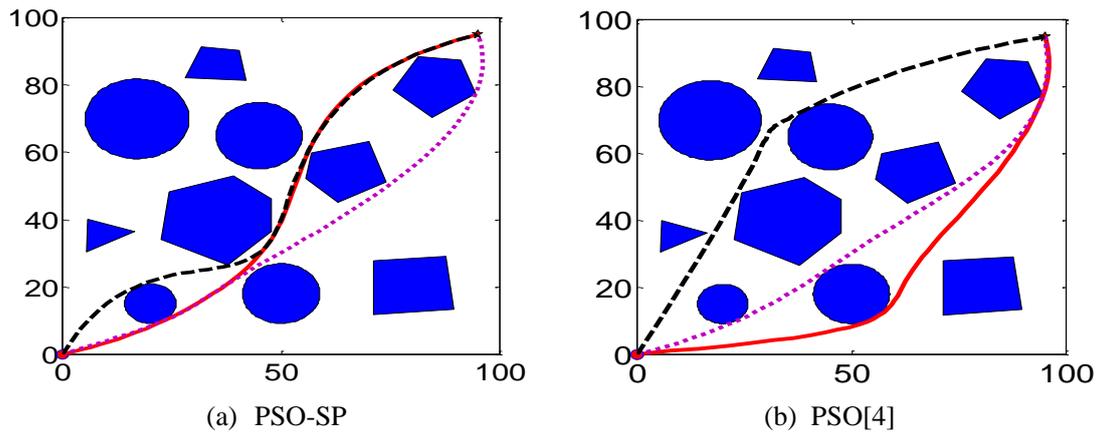


Figure 1 The best 3 paths generated by PSO-SP and PSO[4] respectively

Table 1 Statistical comparisons of two algorithms

| | PSO[4] | PSO-SP |
|------------------------|----------|----------|
| Best solution (length) | 141.7845 | 143.5578 |
| Average length | 148.0574 | 154.2678 |
| Average time (second) | 23.57s | 15.32s |

Conclusions

A new mobile robot path planning method based on PSO and cubic spline function was proposed in this paper. The proposed algorithm took coordinates of path codes as particle coding, which significantly lowered particle coding dimension. Designed penalty function strategy gave opportunity to solutions with low illegality degree to evolve into optimal and legal solutions. In order to prevent the algorithm from falling into local optimum, special initialization strategy was designed to set initial path nodes. Compared with traditional path planning method based on PSO, operating efficiency of the method in this paper was higher. Therefore, it might be applied to dynamic and real-time path planning problems in the future.

Acknowledgements

This work was financially supported by the National Natural Science Foundation of China (Grant. No. 11404205).

References

- [1] Takahashi O, Schilling R J. Motion planning in a plane using generalized Voronoi diagrams[J]. IEEE Trans on Robotics and Automation, 1989, 5(2): 143-150.
- [2] Avneesh S, Erik A, Sean C, et al. Real-time path planning in dynamic virtual environment using multiagent navigation graphs[J]. IEEE Trans on Visualization and Computer Graphics, 2008, 14(3): 526-538.
- [3] Shi C, Wang Y, Yang J. A local obstacle avoidance method for mobile robots in partially known environment [J]. Robotics and Autonomous Systems, 2010, 58(5): 425-434.
- [4] Saska M, Macas M, Preucil L, et al. Robot path planning using particle swarm optimization of Ferguson splines[C]. Proceedings of Emerging Technologies and Factory Automation, Prague, Czech, 2006: 833-839.