Multi-goal Path Planning of Mobile Robot Based on Improved Particle Swarm Algorithm

Junjie Li [a], Guolun Guan [b] and Xiaofeng Zheng [c]

College of Automation, Chongqing University of Posts and Telecommunications, Chongqing 400065, P.R. China;

[a]lijunjiejie166@126.com, [b]jianyoutianxia@163.com, [c]380424865@qq.com

Abstract

Mobile robot multi-goal path planning is a new challenge in field of Robotics. In order to resolve the problem, a hybrid algorithm combining Guo Tao(GT) algorithm with particle swarm optimization (PSO) is proposed. The hybrid algorithm introduces the Inver-over operator and the clonal mechanism to PSO. The probability of the learning selection, the iterative threshold and the ratio of local optimal subgroup are used to keep the diversity of swarm. The data experiment demonstrates the performance of PSO is improved. In order to evaluate the practicability of GTPSO in path planning, the GTPSO is compared with IAPSO by pioneer3 robot, the experimental results show that GTPSO can plan a optimal route in different environment better than IAPSO. The trajectory is the shorter and the time consumption is fewest.

Keywords

Multi-goal path planning, PSO, Guo Tao algorithm, Inver-over operator, Clonal mechanism, Robot Operation System.

1. Introduction

Mobile robot path planning is the key part in the robot development Therefore, there were many scholars studied it and there are many common algorithms proposed, such as artificial potential field method [2], A* algorithm [3] and grid method [4]. However, there are some disadvantages for common algorithms, such as it will take too much time for A* algorithm to sustain the open list when the map is very large, and due to the effect of repulsion, it is difficult for artificial potential field method to arrive the goal. Swarm intelligent algorithms could finish efficient global and local search, keeping the diversity of particles. Hence, the Swarm intelligent algorithms will be selected to realize mobile robot multi-goal path planning.

Generally, mobile robot path planning is to find a suitable collision-free route from a start point to a destination in an environment fulfilled with obstacles. Essentially, mobile robot path planning can be regard as the problem of optimization, therefore some intelligent algorithms play the important effect on solving the problem of mobile robot path planning, such as genetic algorithm (GA), ant colony algorithm (ACA), PSO, etc [5]. Li and Feng [6] study mobile robot path planning based on the adaptive GA. Zhang and Ma [7] propose an improved ACA applied to mobile robot path planning in the static environment. The improved ACA adds information of the local path in the environment to the initialization of pheromone and probabilities of path selection, it enhances the convergent speed and avoid mature. Li and Hu [8] proposes a new mobile robot path planning algorithm based on quantum-behaved particle swarm optimization (QPSO) and polar coordinate conversion. Zhang and Gong[9]design robot path planning based on multi-objective particle.

According to above information, we know that the most research of mobile robot path planning focus on the point-to-point path planning currently. But there are many mobile robots need multi-goal path planning in real life, such as cleaning robot, hospital-ward robot and factory-inspecting robot, etc. Mobile robot multi-goal path planning aims to find a collision-free path from the starting point to starting point passing a sequence of goals with the shortest total route and the less time. Multi-goal path planning for mobile robot has many problems now, such as how to keep the balance of the distance and the waste time. To solve the problem, a hybrid algorithm combining GT [10] with PSO (GTPSO) is proposed. The hybrid algorithm avoid particles trapping into local optimum, and improved the performance of searching optima. Comparing with other improved PSO by MATLAB, the result shows the convergent speed of GTPSO is enhanced. Furthermore, the path planning experiment by pioneer3 robot shows that GTPSO can conduct robot path planning effectively in the different environment.

This paper is organized as follows. Section 1 presents the standard Particle Swarm Optimization. The idea of the improved algorithm is introduced in section 2. Condition of improved algorithm verification and performance analysis is proposed in section 3. Section 4 describes the experiment of pioneer3 robot based on GTPSO. Finally, section 5 gives the conclusion and the open problems.

2. Particle swarm oprimization

2.1 Standard PSO.

PSO is a swarm intelligent algorithm, proposed by Eberhard and Kennedy in 1995 [11]. PSO is deduced by observing the foraging behavior of birds, and every bird is regarded as a particle. Particles change flying speed and position by exchanging the information with other particles, finally the optimal solution is found. In N-dimensional space, the particles gradually come to the global best position by adjusting their own position and velocity learning from their neighbors and themselves. The position of the particle is defined with a vector $x_i = (x_{i1}, x_{i2}, \cdots, x_{in})$ and the vector $v_i = (v_{i1}, v_{i2}, \cdots, v_{in})$ is its velocity. In each iteration, the i th particle updates its own position and velocity according to the following formula.

$$v_i = \omega v_i + c_1 r_1 (p_{best} - x_i) + c_2 r_2 (g_{best} - x_i)\,. \tag{1}$$

$$x_i = x_i + v_i\,. \tag{2}$$

Where ω is the inertia weight of the velocity. c_1 and c_2 are the learning factors. r_1 and r_2 are the random number of [0, 1]. p_{best} is the local optimal value. g_{best} is the global optimal value. The range of position and velocity of all particles are limited in $x_i \in [x_{\min}, x_{\max}]$ and $v_i \in [v_{\min}, v_{\max}]$. When the velocity of the i th particle beyond the $v_{\max}$, the velocity of the i th particle is limited to $v_{\max}$.

2.2 Improved PSO

From equation (1) and (2) we can conclude that the best position of the particles after the movement represents the optimal solution of the issue we study. Particle velocity v_i determines the speed of the particles movement. But in an iterative process, particle with the fast velocity do not necessarily mean that it will come close to the optimal position, and it is possible to reverse the direction of movement of optimal position, which will disperse the particles. Therefore, in order to reduce the degree of computational complexity of the algorithm and eliminate the adverse effects caused by speed, the author introduces the idea of displacement calculation formula $s = vt$ to simplify formula (1) and (2) as the following formula.

$$x_i = \omega x_i + c_1 r_1 (p_{best} - x_i) + c_2 r_2 \left(g_{best} - x_i\right). \tag{3}$$

The formula (3) eliminates the speed parameter, and the remaining parameters are the same as the parameters of formula (1) and (2). Formula (3) reduces the order of particles updating formula from second-order equation to first-order equation and simplifies the iterative process of particle. Hu Wang [12] proves by theory and experiment that the process of particle swarm optimization has nothing to do with the speed, and after eliminating velocity parameters, PSO becomes simpler and easier to control.

In the process of particles finding the optimal position, along with the particles come close to the global optimal position, inertia weight ω should gradually decrease so that the particles are more likely to get the global optimal position in later iterations, and obtain the optimal solution of issue we study. So the linearly decreasing inertia weight update formula is introduced as follows.

$$\omega = \omega_{\min} + \left(\omega_{\max} - \omega_{\min}\right) \times \left(\frac{T_{\max} - t}{T_{\max}}\right)^k . \tag{4}$$

Where $\omega \in [0.4, 0.9]$, $\omega_{\max}$ is the maximum inertia weight and $\omega_{\min}$ is the minimum inertia weight. $T_{\max}$ is the maximum number of iteration. t Represents the number of the current iteration. k Indicates the decreasing degree of ω along with the process of iteration. Shi and Eberhard [13] proposed an improved algorithm of linearly inertia weight alteration where k = 1 and the convergent rate of PSO in this paper has been improved.

With the iterative process, there will be some particles falling into local optimum, which will decrease the ability of particles optimization in the latter part of iterative process. In order to ensure the quality and diversity of the population of particles, we launched clonal mechanism. Clone simulates the principle of asexual reproduction, which completely copies the father's information to produce a certain amount of next generations. Clonal operation of particle x_i of PSO: $C\left(x_i\right) = \left\{x_i^1, x_i^2, \cdots, x_i^n\right\}$, q_i is the clonal size of x_i. The expression of q_i is expressed as follows.

$$q_i = \operatorname{int}\left(N_c \cdot \frac{f\left(x_i\right)}{\sum_{i=1}^{n} f\left(x_i\right)}\right) \quad i = 1, 2, \cdots, M . \tag{5}$$

int() indicates the integer function; N_c is the upper limit of a clone; $f(x_i)$ is the fitness value of particles. From equation (5), we can conclude that the number of clonal particles is determined by fitness value of particles.

When part of the particles fall into the local optimum, in order to make the rest of the particles have a better fitness value be cloned, we should set conditions to launch the clonal mechanism so that particle swarm could have a higher optimized capability. When the average distance D is lower than the threshold value we set, PSO launches clonal mechanism to clone some particles with better fitness value.

$$D = \frac{1}{n * sum\left(g_{best} - x_i\right)} . \tag{6}$$

Where, n is the size of space. g_{best} is the global best position.

3. GT algorithm with particle swarm optimization algorithm(GTPSO)

3.1 Guo Tao algorithm

Guo Tao algorithm is proposed based on the idea of GA. Guo Tao algorithm is realized by keeping particle subsequence adaptive inversion. The operation constantly breaks the original sequence, retaining better sequence and weeding out poor sequence, consequently Guo Tao algorithm forms a new and more rational sequence [14].

There are three important parameters in the GT: p , r and T_g . p Represents the probability of learning selection, which decided the learning object of particles. T_g Denotes iterative threshold. r is the ratio of local optimal subgroup, which is used to determine the size of local optimal subgroup. For example, if the size of particles is 100 and $r = 10\%$, the size of local optimal sub-group of is 10.

In GT, a random number r_m in the range (0, 1) will be raised. Afterwards, r_m is compared with p . If $r_m < p$, and the current number of iterations $t < T_g$, the learning object of current particle is selected randomly from the swarm. If $r_m < p$ and $t > T_g$, the global best particle is selected as the learning object. If $r_m > p$, the learning object of current particle will be selected from the local optimal subgroup.

3.2 GTPSO algorithm

In PSO, particles trap into local optima easily in the running time, which results in the diversity of particles and the capacity of searching optima decreasing. To resolve the problem of PSO, The idea of GT algorithm is applied to PSO (GTPSO).The detail procedure of GTPSO are listed as follows:

Step 1 Initial swarm, including inertia weight, learning factors, velocity and position of particles, the maximal iterations.

Step 2 Calculating the fitness value of each particle.

Step 3 setting the ratio of local optimal subgroup r , calculating the size of local optimal subgroup.

Step 4 r_m decides the learning object of particles.

Step 5 Updating the position of particles according to Equation (3).

Step 6 Ending.

4. Experimental Result Analysis

4.1 Test function

In order to test the performance of GTPSO, we select four fitness functions. In the paper, we use GTPSO to find out the minimum of four test functions. Function name, function equation and best value are listed in Table 1.

Table 1. Test function and parameter setting

Function	Equation	Best value
Sphere	$f(x) = \sum_{i=1}^{n} x_i^2$	0
Rastrigin	$f(x) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)$	0
Grewank	$f(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i - \prod_{i=1}^{n}\cos(\frac{x_i}{i}) + 1$	0
Schaffer	$f(x) = 0.5 + \frac{\sin^2\sqrt{x_1^2 + x_2^2} - 0.5}{1.0 + 0.001(x_1^2 + x_2^2)}$	0

Sphere function is a simple multi-mode function. Rasrigin function and Grewank function have a global optimum value and multiple local optima. Schaffer function is a two-dimensional multi-mode function.

4.2 Evaluation criterion

In order to evaluate the performance of GTPSO, there are two criteria presented. They are listed as follows.

1) Global best value: it reflects the convergence accuracy of the improved algorithm;

2) Standard deviation: it reflects the stability of the improved algorithm. If the standard deviation is smaller, the performance of improved algorithm is better. Standard deviation calculation formula is expressed as follows.

$$\overline{f} = \frac{1}{S}\sum_{i=1}^{s} f(i). \tag{6}$$

$$\delta^2 = \frac{\sum_{i=1}^{s}(f(i)-\overline{f})}{S-1}. \tag{7}$$

$\overline{f}$ is the average value of particle fitness function; S represents the size of particle population; $f(i)$ is expressed as i th particle fitness value.

4.3 Numerical experimental results and analysis

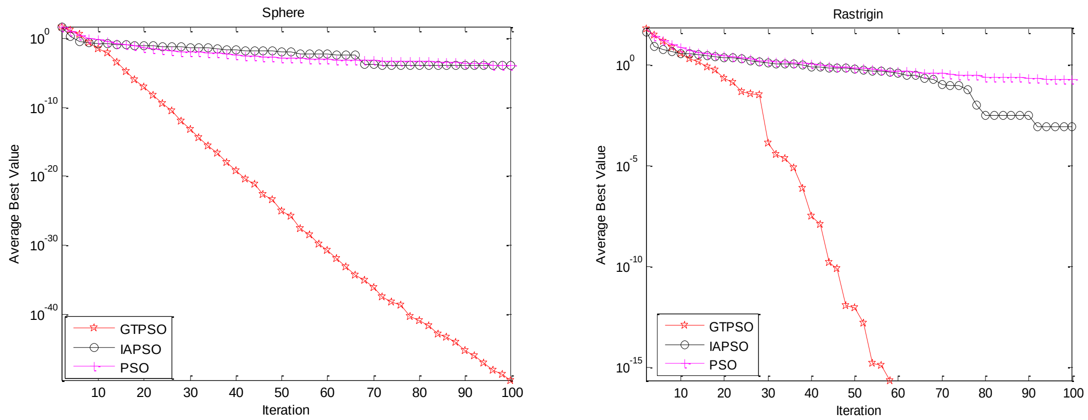
To test the advantages of GTPSO, we compare GTPSO with inertia weight adaptive particle swarm optimization (IAPSO) [15] and standard PSO. Some parameters of GTPSO are set as follows: $S = 40, T_{max} = 100, p = 0.3, T_g = 0.8 \times T_{max}, r = 20\%, k = 3$. Space dimension is 2. Learning factors of GTPSO are: $c_{11} = c_{12} = 1.4$, and learning factors of IAPSO and PSO are: $c_{21} = c_{22} = 2$.

In order to evaluate the performance of both GTPSO, IAPSO and PSO, the experiment is run 50 times, and the average value of 50 time experiment is regard as the average value of the algorithm. The result is displayed in Fig. 1 and the data is listed in Table 2 and Table 3.

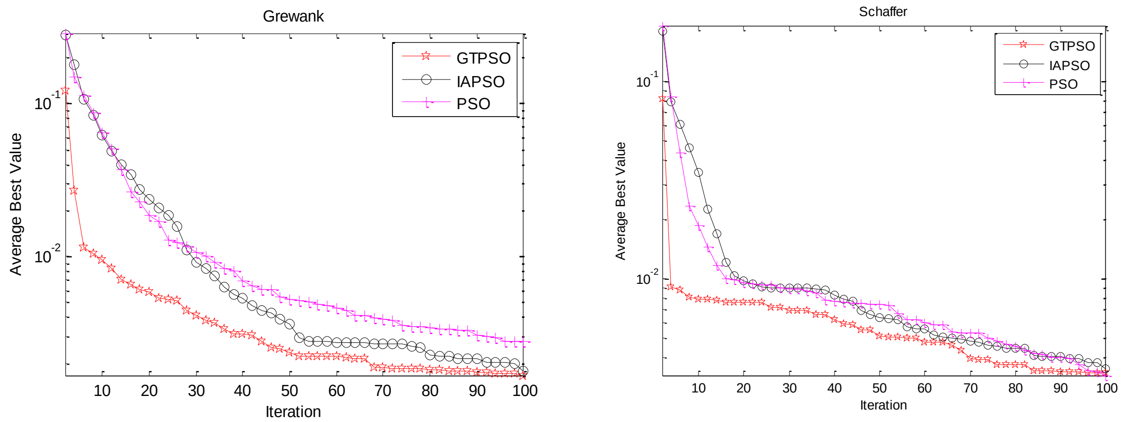
The searching process of 3 algorithms is shown in Fig. 1. From Fig. 1, it is known that GTPSO has a fastest searching speed in 3 algorithms, especially in Fig. 1(b), GTPSO searching the best value of Rastrigin function at the 60th iteration approximately. In the later stages of iteration, GTPSO keep a strongest searching capacity. The experiment demonstrates that the searching speed of GTPSO is enhanced.

Table 2 is the mean best value of 4 functions, which demonstrate that the searching optima capacity of GTPSO is stronger than other algorithms. Table 3 is the standard deviation, which proved that the stability of GTPSO is superior to other algorithms.

According to the above analysis, it shows that the performance of GTPSO is best than other algorithms.



(a) (b)



(c) (d)

Fig. 1 Average value of algorithm comparison

Table 2. Comparison of average testing value of 50 times

Algorithm	Sphere	Rastrigin	Grewank	Schaffer
GTPSO	0	0	0.0016630317111	0.0030750896693
IAPSO	0.0757547734410	0.0144453488886	0.0017595283271	0.003516573461
PSO	0.1449288687412	0.2482552331681	0.0027705494651	0.003241071019

Table 3. Comparison of Standard deviation of 50 times experiment

Algorithm	Sphere	Rastrigin	Grewank	Schaffer
GTPSO	0	0	0.001033477448271	0.003704172056235
IAPSO	0.40795193990718	0.077790584462007	0.003128094986865	0.004300035599509
PSO	0.23947374017565	0.239473740175653	0.00349049050969	0.004128861239793

5. Application of GTPSO

In order to further test the validity and practicability of GTPSO, the author tests GTPSO on Pioneer3 robot equipped with ROS (Robot Operation System). Experimental platform is shown in Fig 2.

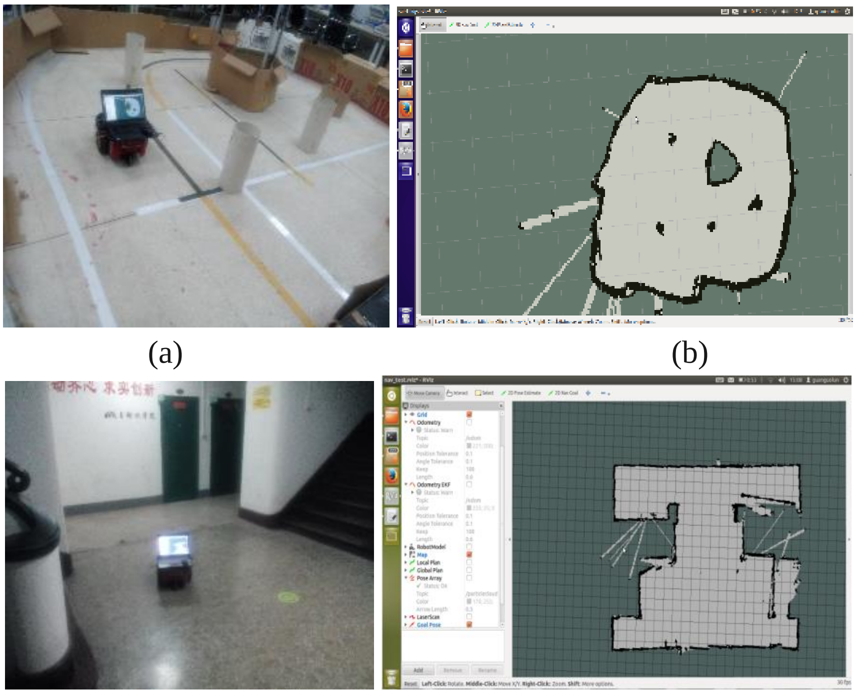
Fig. 2 Hardware platform

5.1 ROS (Robot Operation System)

ROS is an open-source robot operating system designed by Willow Garage [16]. ROS provides services that is similar to operation system, including abstract description of hardware, the bottom driver management, sharing features, messaging deliver between programs and the of management program distribution package. ROS also provides a number of tools and a collection of packages to help software developers to create a robot applications. ROS is a distributed process framework so that he program can be performed independently. ROS has some advantages, including point to point design and the independent programming language, open source, etc [17-19].

5.2 Experimental Environment

There are two environments in this experiment. (1) The laboratory environment. (2)The campus office environment. Fig. 3 are the two real environments and the environment map. In these environments including the walls, corners, still barriers, etc. The black parts represent obstacles in Fig. 3(b). Fig. 3(c) and Fig (d) are the office floor environment of College of Automation Chongqing University of Posts and Telecommunications (obstacles are not put yet).



(a) (b)

(c) (d)

Fig. 3 Experiment environment

5.3 Experimental Results

Fig. 4 displays the trajectory in the laboratory environment. Firstly, pioneer3 robot begins to move from the starting point to point1. After resting a while at point1, poineer3 robot moves to point2 in accordance with the positional information of the next goal. During the process, if a new obstacle suddenly appears at the point 3, the robot will relocate and find a new path to the point 2.

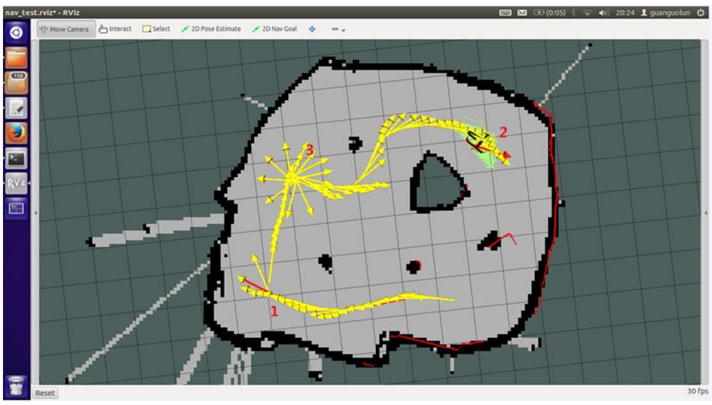


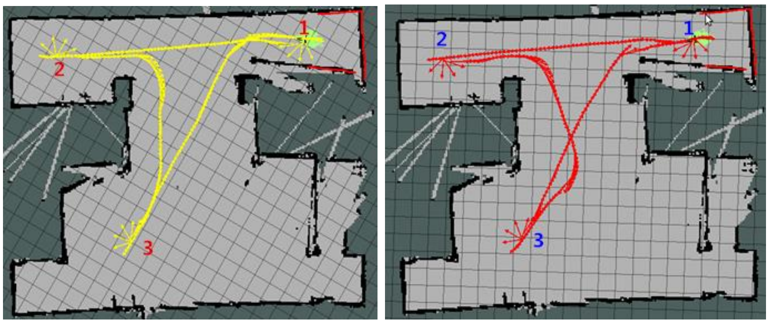
Fig. 4 Laboratory environment path trajectory

Fig. 5 shows the trajectory of pioneer3 robot in the environment without obstacles. Pioneer3 robot moves from point1 to point2, then to point3, and finally gets back to point1. Fig. 5(a) is the trajectory of GTPSO. Fig 5(b) is the trajectory of IAPSO. The robot walks against the wall after a turn, and it moves smoothly without divergence from 3 to point 1 in Fig. 5(a), but in Fig. 5(b) the robot walks away from the wall after a turn, though it walks smoothly from point3 to point1, the path crosses.

Fig. 6 shows the trajectory of pioneer3 robot in the environment filling the obstacles. Fig. 6(a) is the trajectory of GTPSO. Fig 5(b) is the trajectory of IAPSO. With the increase of obstacles, the mobile robot moves from the same starting position and then returns to starting point. .

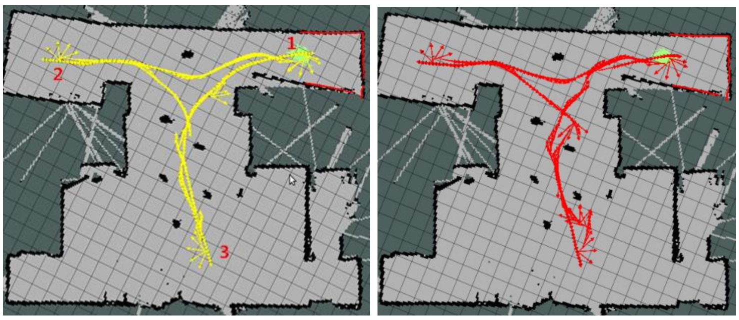
In comparison with these two pictures in Fig. 6, we can see that with the increasing complexity of experimental environment, two algorithms both accomplish multi-goal path planning. But GTPSO is better than IAPSO in view of walking distance and the time consuming (in Table 4).

Fig. 7 shows the trajectory of 5 goals. Fig. 7(a) is the trajectory of GTPSO. Fig 7(b) is the trajectory of IAPSO. 2 algorithms finish the 5 goal path planning, but from the Fig. 7, it is obvious that the trajectory of GTPSO is safer than IAPSO.



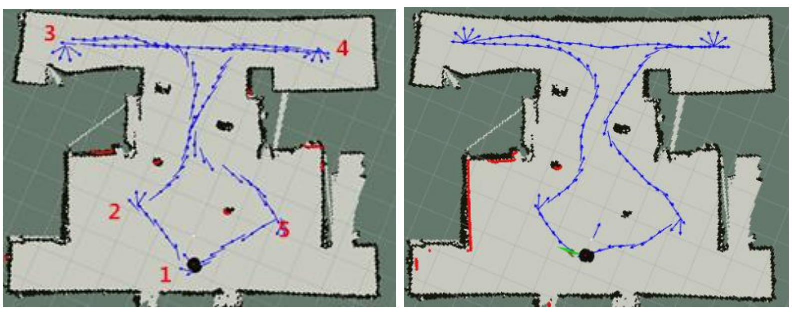
(b)

Fig. 5 Environment without obstacle



(a) (b)

Fig. 6 Environment with obstacles



(a) (b)

Fig. 7 5 goals experiment

In order to get the data of walking distance and time consuming above three kinds of experiments, we do 10 times experiments to get the average value. Now the time consuming and the moving distance of robot is compared in Table 4.

Table 4. Performance evaluation of GTPSO and IAPSO

Experiment		Distance(m)	Time(s)
Fig. 5	GTPSO	20.1	114
	IAPSO	20.1	144
Fig. 6	GTPSO	19.5	186
	IAPSO	20	216
Fig. 7	GTPSO	23.9	220
	IAPSO	24.5	245

From Table 4 we conclude that with the increasing complexity of environments, the expending time of GTPSO and IAPSO on path planning is also increasing. However, in the barrier-free environment and under the condition of the same walking distance, GTPSO uses less time than IAPSO; in multi-obstacle environment, GTPSO takes less time and walks the shortest distance. When the starting point is changed, GTPSO has obvious advantages over IAPSO in terms of time consuming and walking distance.

Overall, GTPSO and IAPSO can accomplish the same task under the same environment. But with the change of the environment, GTPSO algorithm is superior to IAPSO in walking distance and the time spending. Thus, comprehensively speaking, the performance of GTPSO is better than IAPSO.

6. Conclusion

In this paper, a hybridization algorithm of GT-PSO was presented to realize mobile robot multi-goal path planning. In GTPSO, there are three parameters, and to keep the diversity of particle, the data experiment shows that the GTPSO has a robust stability and searching capacity. In order to evaluate the practicability of GTPSO in path planning, the GTPSO is used to mobile robot multi-goal path planning that is equipped with ROS. The GTPSO is compared with IAPSO by pioneer3 robot, the experimental demonstrated that GTPSO can finish the multi-goal path planning, and the path is shortest and consuming time is fewest.

Acknowledgements

Project support by Chongqing Education Commission Foundation under Grant (KJ1400432) and Chongqing Science Technology Commission Foundation under Grant (cstc2015jcyjBx0066, KJ1 600442).

References

[1] Wang Z W, Guo G. Status of mobile robot navigation technology and prospects [J]. Robot, 2003, 25(5):470-474.

[2] Deng X J. Mobile robot path planning based on improved artificial potential field method [J]. Journal of Shandong University of Technology (Natural Science), 2014 (1): 2021-2023.

[3] Wang H W, Ma Y, Xie Y, etc. Mobile robot path planning based on smoothing A* algorithm [J]. JOURNAL OF TONGJI UNIVERSITY (NATURAL SCIENCE), 2010, 38 (11): 1647-1650.

[4] Lu Q. The study of mobile robot path planning based on Grid Method [J]. Computer and Information Technology, 2007, 15 (6): 24-27.

[5] Pan J, Wang X S, Cheng Y H. Mobile robot path planning based on improved ant colony algorithm [J]. Journal of China University of Mining &Technology, 2012, 34 (01): 473-475.

[6] Li Q, Feng J L, Liu Y L, etc. Adaptive genetic algorithms in mobile robot path planning [J]. Journal of University of Scinece and Technology Beijing, 2008 (3): 316-323.
[7] Zhang Q, Ma J C , Xie W, Ma L Y. Mobile robot path planning based on improved ant colony algorithm [J] Journal of Northeastern University(Natural Science), 2013, 34 (11): 69-70.
[8] Li R f, Hu L. Mobile robot trajectory planning and experiment based on QPSO algorithm [J]. Control and Decision, 2014(12):2151-2157.
[9] Zhang Y, Gong D W, Zhang J H. Robot path planning in uncertain environment using multi-objective particle swarm optimization [J], Neurocomputing.2013, 103(2):172–185.
[10] Tao G, Michalewicz Z. Inver-over Operator for the TSP [J]. Lecture Notes in Computer Science, 1998, 1498:803-812.
[11] Eberhart R, Kennedy J. A new optimizer using particle swarm theory[C]// Micro Machine and Human Science, 1995. MHS '95. Proceedings of the Sixth International Symposium on. IEEE, 1995:39-43.