An Improved PSO Algorithm For Robot Multi-Goal Path Planning

Xingcheng Pu [1, a], Junjie Li [2, b] and Yi Zhang [1, c]

[1]School of Science, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

[2]Research Center of Intelligent System and Robot Chongqing University of Posts and Telecommunications, Chongqing 400065, China

[a]MAME2013@163.com, [b]Conf_51EiSCi@163.com, [c]yyyy@ccc.com

Abstract

This paper presented a new algorithm to search the optimal multi-goal trajectory of path for mobile robot in an obstacle-filled environment using hybridization of Immune algorithm (IA) with improved particle swarm optimization (IPSO). The proposed method embedded the social essence of IPSO with the clonal selection mechanism of IA. Because of adopting dynamically inertia weight with learning factors and clonal selection mechanism to update the position and velocity simultaneously, the new method keeps a good balance between convergence quickly and the diversity. The purpose of the method is to minimize the walking path length corresponding to the minimized time of robot to the end position passing all goals in the environment. By comparing with MCO, RTPSO and PSO with five functions, the data analysis demonstrate the advantage of the new method. The comparison experiment of multi-goal path planning on Pioneer3-dx robot with robot operation system (ROS) show outperforms of IAIPSO as compared with IAPSO with respect to the optimization of robot walking length and time consuming.

Keywords

Multi-Goal Path Planning, IA, PSO, Clonal Selection Mechanism, Robot Operation System.

1. Introduction

The problem of path planning in mobile robotics is deemed as an important task. It has researched from the paper [1] that work finds a path for mobile robot to get the predefined destination from a known starting position with collision-free in the given environment. Mobile robot path planning has been classified into local and global path planning. In local path planning, mobile robot avoids the obstacles by steps and selects the next position to get to the goal by meeting some demands like the optimum of path, time and energy [2-4]. In the previous of mobile robot navigation to goal, it determines the collision-free route from a starting position with the help of global path planning. In the last decades, many scholars worked on the research of mobile robot path planning based on some common methods, such as artificial potential field method [5], A algorithm [6], grid method [7], simulated annealing [8] and neural network [9] and evolutionary algorithm [10]. For classical algorithms, however, too much time is spent on maintaining the open list in the context of large map; and sometimes, it is difficult to reach the destination for the influence of repulsion. Heuristic algorithms keep a balance between diversification and intensification to finish efficient global and local search. Hence, the heuristic algorithms will be used to deal with the problem of mobile robot multi-goal path planning.

Mobile robot multi-goal path planning is a task that applied in many robot applications, from cleaning robot and room robot to factory inspection and assembly. In mobile robot multi-goal path planning, robot has a known starting position and sequences of goals in the environment and robot has to search a collision-free route for passing a sequence of goals with the minimized distance. Hence, there are two steps in the method of solving multi-goal path planning, such as building a map of environment and finding a tour to visit all goals in the map. In this paper, we will take this method to accomplish mobile robot multi-goal path planning. There are many goals in the environment, however, the path planning of all goals and time consuming are the high-priced risk. To solve the problem, hence, particle swarm optimization (PSO) [11] is used to decide the choice of goals and plan a feasible path.

PSO is an intelligent algorithm that has been applied to develop an approach for mobile robot path planning [12]. Multi-objective optimization problem for robot path planning in uncertain environment has been solved by PSO [13]. A method for mobile robot navigational controller has been improved using advanced PSO [14]. In non-convex obstacle-filled environment, mobile robot path planning using PSO [15]. An algorithm for free-float space robot to plan trajectory by PSO [16]. Hybridization of PSO has been applied for path planning, such as IPSO-IGSA [17] and ICQL-IPSO [18].

However, the premature convergence of PSO is occurred in the running process while coping with complicated problems, such as navigation in real word by optimization problem like the solution of mobile robot path planning. It is sensitive for PSO to the coefficients, such as inertia weigh, learning factors and velocity. Therefore, it is necessary for PSO to further improve the performance for achieving an optimal algorithm for real world problem. Immune algorithm (IA) is a heuristic searching method by the animal immune system, which has some merits, such as keeping the diversity of the population and avoiding the trapping into local optimum. Thus, we plan to hybrid the improved PSO and IA for overcoming the disadvantage of PSO.

In this paper, we concentrate on the minimized distance, the time-consuming with the least for robot based on the proposed method. By the hybridization of IA and IPSO (IAIPSO), the novel method, which is used to realize multi-goal path planning of mobile robot in an obstacle-filled environment, is developed. This presented method effectively improved the performance in PSO. The experiment is tested on a pioneer3-dx (P3DX) robot, which is equipped with robot operation system (ROS). The result shows that IAIPSO can plan a successful path in a reasonable amount of time by avoiding the obstacles in the path to goals.

This rest of the paper is outlined as follows. Section 2 presents PSO algorithm and improved PSO. The idea of the IAIPSO is introduced in section 3. Condition of improved algorithm verification and performance analysis is proposed in section 4. Section 5 describes the experiment of P3DX robot based on IAIPSO. Finally, a conclusion and the future work of this paper are given.

2. Particle Swarm Optimization Algorithm

2.1 Standard Particle Swarm Optimization Algorithm (PSO)

PSO is a bio-inspired evolutionary intelligent algorithm that simulates the foraging behavior of birds, which was initially proposed by Eberhart and Kennedy [11]. In PSO, every bird is seen as a particle and the population of particles presents a swarm. The position of the particle, in D-dimensional space, is defined as a vector $x_i=\left(x_{i1},x_{i2},...,x_{iD}\right)$ and the vector $v_i=\left(v_{i1},v_{i2},...,v_{iD}\right)$ is its velocity. In the searching space, the velocity of particles is updated by learning from two best positions. One is p_{best} that is the own best position of particle. The other is g_{best} that is the global best position. Hence, the ith particle uses two equations to update its velocity and position:

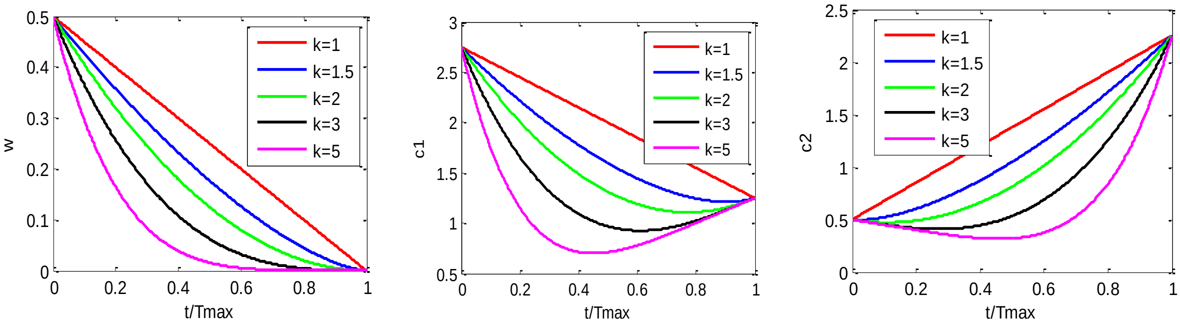
$$v_i(t+1)=\omega v_i(t)+c_1r_1\left(p_{best}-x_i\left(t\right)\right)+c_2r_2\left(g_{best}-x_i(t)\right) \quad . \tag{1}$$

$$x_i\left(t+1\right)=x_i\left(t\right)+v_i\left(t\right) \quad . \tag{2}$$

where, ω represents the inertia weight that controls the influence of the prior velocity of the particle at iteration t on its velocity at iteration t+1. c_1 and c_2 indicate the learning factors of cognition and social. r_1 and r_2 are two numbers randomly distribute in the range [0,1] that are determined every time when the iteration is occurred. The scope of position and velocity of particles are limited in $x_i\in\left[x_{\min},x_{\max}\right]$ and $v_i\in\left[v_{\min},v_{\max}\right]$ respectively. When the maximal velocity of the ith particle exceeds to $v_{\max}$, it is limited to $v_{\max}$.

2.2 Improved Particle Swarm Optimization (IPSO)

In the searching space, a global optimum is found successfully by particles via communicating with other particles and learning their experience. When we use PSO to obtain the optimal solution of an optimization problem, it is imperative for PSO to control its exploration and exploitation ability at iteration. In PSO, its performance rely on its three parameters, namely inertia weight ω, learning factors c_1 and c_2.



(a) (b) (c)

Fig. 1 The curve graph in different value of k . (a) Description of what is the curve graph of ω; (b) Description of what is the curve graph of c_1; (c) Description of what is the curve graph of and c_2.

The inertia weight ω in Equation (1) plays an affect on the performance of PSO. A larger ω is conducive to the global exploration, while a smaller ω facilitates exploitation. Thus, it is important that the suitable control of ω assure accurately PSO find the optimal solution. The search capability of particles is regulated by dynamically changing inertia weight. Here, a "dynamically changing inertia weight factor" (DCIWF) is proposed in Equation (3).

$$\omega = (\omega_{max} - \omega_{min}) \times \left({}^{T_{max} - T}\!/_{T_{max}} \right)^k \quad . \tag{3}$$

where, $\omega \in [0.4, 1.9]$, $\omega_{\max}$ 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. In Fig. 1(a), the curve graph of ω in different value of k is shown. In the early stage of running process, if the value of k is large, the damping velocity of ω is quick. Contrary, if the value of k is small, the damping of ω is slow. In the iterative process, ω is gradually decreasing, and the exploitation ability is gradually increasing.

Similarly, the particles will trap into local optimum, when the value of social factor c_2 is larger than cognitive factor c_1, and the larger value of cognitive factor leads the particle to wander around the searching space. The quality of the solution is enhanced by changing the cognitive and social ability of PSO in such a way that c_1 is reduced and c_2 is increased in the process of generation. The updating equations are listed as follows:

$$c_1 = c_{1\max} \times \left({}^{T_{\max} - t}\!/_{T_{\max}} \right)^k + c_{1\min} \left({}^{t}\!/_{T_{\max}} \right) \quad . \tag{4}$$

$$c_2 = c_{2\max} \times \left({}^{t}\!/_{T_{\max}} \right)^k + c_{2\min} \left({}^{T_{\max} - t}\!/_{T_{\max}} \right) \quad . \tag{5}$$

where, $c_{1\max}$ and $c_{1\min}$ represent the maximum and minimum of c_1 respectively. $c_{2\max}$ and $c_{2\min}$ are the maximum and minimum of c_2 respectively. In Fig. 1(b), the curve graph of c_1 in the value of $c_{1\max}$ and $c_{1\min}$ are shown. In Fig. 1(c), the curve graph of c_2 in the value of $c_{2\max}$ and $c_{2\min}$ are shown. In the early stage of running process, if the value of k is large, the decreasing velocity of c_1 is quick and the increasing velocity of c_2 is slow. Contrary, if the value of k is small, the damping of c_1 is slow and the damping of c_2 is quick.

In the running time, the value of ω is gradually attenuating, the local searching capability is improved, and the value of c_1 is also gradually decreasing and the c_2 is gradually increasing, the ability of autognosis is weaken and the social ability is enhanced. Therefore, in the paper, the value of k is set to 2 according to the Figure. 1, PSO can keep the balance between global searching and local searching abilities.

3. Immune Algorithm With Particle Swarm Optimization Algorithm (IAIPSO)

3.1 Immune algorithm

In the nature, all living organisms have an immune system whose intricacy changes with species. When the living beings are under attack of pathogens, the immune system try to resist and exterminate the pathogens by the immune cells, which via activating, differentiating and proliferation. In immune system, however, the offspring only copy the information from clonal parents, rather than communicating the different information.

IA (Immune algorithm) is a bionic algorithm based on immune system, which simulates mainly the mechanism and function of animal immune system in the nature. In IA, clonal selection algorithm (CSA) based on clonal selection theory is an immune optimization algorithm, including the clone, mutation and selection, by simulating the process of micro-evolution of immune system [19].

3.2 The procedure of IAIPSO

In PSO, when a particle finds the current best position, other particles will draw close to it in the running time, which results in the performance of local optimum and the diversity of particles decreasing. To overcome the shortcoming of PSO, IA is introduced into IPSO (IAIPSO). The detail flowchart of IAIPSO is shown in Fig. 2. The detail procedure of IAIPSO are listed as follows:

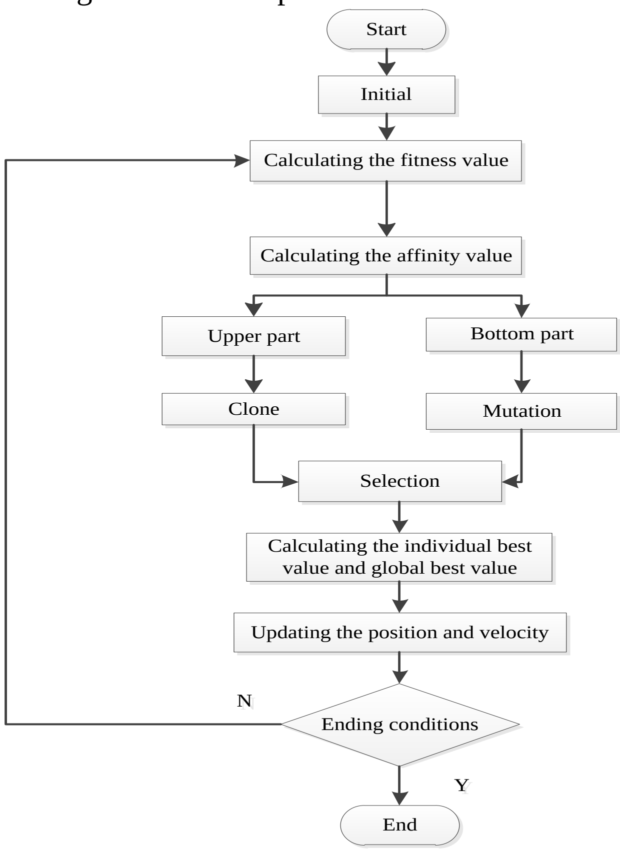


Fig. 2 The flowchart of IAIPSO

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, and particles are seen as antibodies.

Step 3 Computing the affinity of each antibody. According to the fitness value and location of particle, the affinity formula as follow:

$$affinity_i = \frac{fitness_i + g_{best}}{dist_i + 1} \quad . \tag{6}$$

where, $dist_i$ is the distance of ith particle to global best particle:

$$dist_i = \sqrt{x_i - g_{best}} \quad . \tag{7}$$

According to the affinity of each particle, there are two steps executed. Firstly, the particle swarm of current step is ranked according to the affinity value of each particle. Secondly, the population was divided into upper half part and bottom half part. The upper half part was seen as the parents for clone, and the bottom half part was regard as the parents for mutation.

Step 3.1Clone. According to affinity value of antibody, the antibody with high affinity value will be selected for cell division to enhance the diversity. Clonal operator of particle x_i is $C(x_i) = \left\{x_i^1, x_i^2, \ldots x_i^{q_i}\right\}$. q_i is the clone size of x_i. The expression of q_i is expressed as follow:

$$q_i = int\left(N_c \cdot \frac{f(x_i)}{\sum_{i=1}^{M} f(x_i)} \right) \quad . \tag{8}$$

where, $\mathrm{int}(\,)$ indicates the top integral function. N_c is the upper limit of the clone. $f(x_i)$ is the fitness value of particles. M is the number of particles that need be cloned. As we can see from Equation (8), the clonal size of x_i is decided by the fitness value of x_i.

Step 3.2 Mutation. In order to ensure the ergodicity of particle, these particles with lower affinity value are selected to mutating operation.

Step 4 Selection. Selecting the particles with high affinity value to constitute a new swarm whose size is same as the old one.

Step 5 Updating the velocity and position of particles according to Equation (1) and Equation (2) and calculating the p_{best} and g_{best} of particles.

Step 6 Ending.

4. Experimental Setting and Result Analysis For Benchmark Function

4.1 Evaluation criterion

Two criteria are presented to evaluate the performance of IAIPSO. They are listed as follows:

Global best value: it is the optimal solution of fitness function, and it reflects the accuracy of global optimum that is searched by the improved algorithm.

Standard deviation: it is used to test 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 [20]:

$$\bar{f} = \frac{1}{S}\sum_{i=1}^{S} f(x_i) \quad . \tag{9}$$

$$\delta^2 = \frac{\sum_{i=1}^{S}\left(f(x_i) - \bar{f}\right)}{S-1} \quad . \tag{10}$$

where, $\bar{f}$ is the average value of the fitness function; S represents the size of particle population; $f(x_i)$ is the global best value of the ith iteration.

4.2 Benchmark functions and control parameter settings

In this section, there are five well-known benchmark functions provided to evaluate the performance of IAIPSO. In the paper, we use IAIPSO to find out the optimum solution of four benchmark functions after a foregone iterative number. The functions, equations and solution space are presented in Table 1. Rosenbrock function is a simple unimodal function. Rasrigin function and Grewank function are two multimodal functions designed with a vast number of local minimum. Matyas function is an unimodal and impartibility function. There is a narrow global best basin and some minor local best solution with Ackley function.

Table 1. Five typical fitness functions

Functions	Equation	Solution Space
Rastrigin	$f_1(x)=\sum_{i=1}^{n}\left(x_i^2-10\cos(2\pi x_i)+10\right)$	[-5.12,5.12]
Mat yas	$f_2(x)=0.26\left(x_1^2+x_2^2\right)*0.48*x_1*x_2$	[-10,10]
Grewank	$f_3(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i-\prod_{i=1}^{n}\cos\left(\frac{x_i}{i}\right)+1$	[-600,600]
Ackley	$f_4(x)=-20\exp\left(-0.2\sqrt{\sum_{i=1}^{n}x_i^2/n}\right)-\exp\left(\sum_{i=1}^{n}\frac{\cos 2\pi x_i}{n}\right)+20+e$	[-32,32]
Rosen brock	$f_5(x)=\sum_{i=1}^{n}\left[100\left(x_{i+1}-x_i^2\right)^2+\left(1-x_i\right)^2\right]$	[-50,50]

The inertia weight $\omega_{\max}$ and $\omega_{\min}$ are 0.4 and 1.9 respectively. $c_2\in[1.25,2.75]$ and $c_2\in[0.5,2.25]$. The particle swarm size is 40. The iteration number is 100, 400 and 600 respectively. The dimension of Matyas is 2 and the remaining functions are 10. There are three algorithms, multiagent coordination optimization (MCO) [21], ring topology PSO (RTPSO) [22] and standard PSO, selected to compare with IAIPSO.

To evaluate the performance of algorithms, four algorithms all run 30 times. The mean value (average best value) and the standard deviation are use to evaluate the convergence precision and stability of IAIPSO. The result is listed in Table 2 and Table 3, and the iteration processes are displayed in Fig. 3.

4.3 Results and discussion

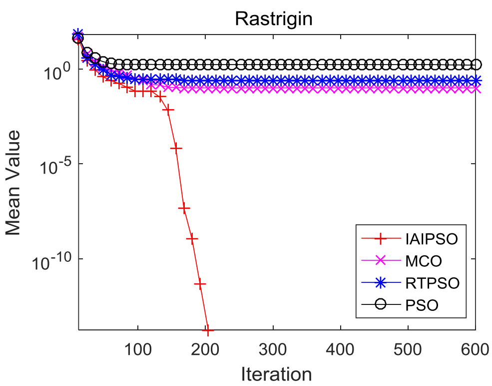
From Table 2, we can conclude that the mean values are closed to the optima with the increase of iteration number, especially IAIPSO. According to Table 3, the stability of IAIPSO is superior to three other algorithms. Hence, we can draw a conclusion that the stability and convergent efficiency of IAIPSO have been enhanced.

The curve of mean best value is shown in Fig. 3. The searching speed of IAIPSO is the fastest in four algorithms, especially in Fig. 3(a), the best value of Rastrigin function is obtained at the 200th iteration approximately by IAIPSO. Though there are three algorithms searching the best value of Matyas function, the convergence rate of IAIPSO algorithm is fastest. The searching capacity of IAIPSO is stronger than the other algorithms in the later stages of iteration. Hence, the experiment shows that the searching accuracy of IAIPSO is improved.

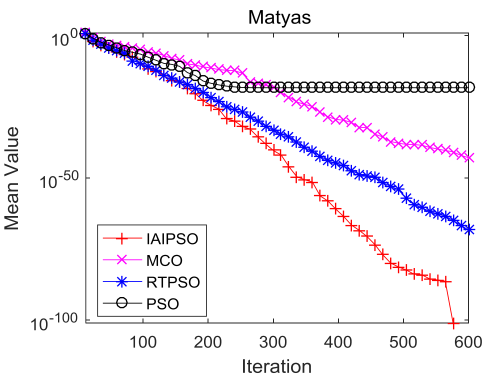
According to the Fig. 3 and the analysis of the above tables, we can summarize that the IAIPSO performs better than other algorithms in this paper.

Table 2. Comparison of mean values

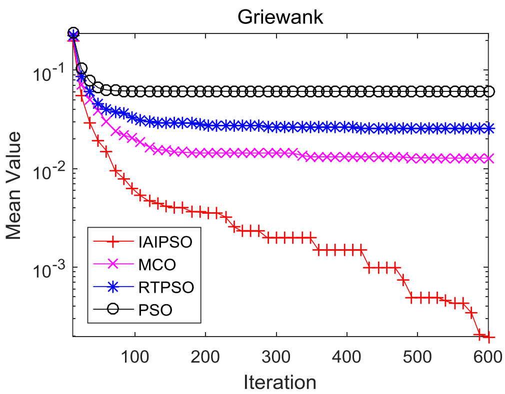
Iter		IAIPSO	MCO	RTPSO	PSO
100	f_1	0.033165422	0.692379468	0.173239074	0.220016036
	f_2	9.357E-12	0.255586591	8.6876E-06	3.50856E-08
	f_3	0.008044507	0.026209046	0.02710015	0.03162724
	f_4	5.8E-14	0.005902916	9.07135E-08	0.658832071
	f_5	0.00703674	0.016710406	0.01102847	0.022568052
400	f_1	1.0e+02	1.0e+02*1.189259917	1.0e+02*0.000663306	1.0e+02*0.00331653
	f_2	1.0e-07*0.035265596	1.0e-07*0.289491378	1.0e-07*0.667543741	1.0e-07*0.884027516
	f_3	0.003779979	0.023093949	0.01536848	0.077794472
	f_4	1E-15	1E-15	1E-15	0.808579789
	f_5	0.005022305	0.008747897	0.007320817	0.011350625
600	f_1	0	0.198991811	0.033165793	0.366578849
	f_2	0	1.0e-20*0.788638809	1.0e-20*0.554833905	1.0e-20*0.854416091
	f_3	0.000493069	0.018163063	0.01109406	0.054426865
	f_4	1.0e-09*0.000000888	1.0e-09*0.000000888	1.0e-09*0.000000888	1.0e-09*0.537291397
	f_5	0.002928859	0.016703909	0.007185658	0.021276431



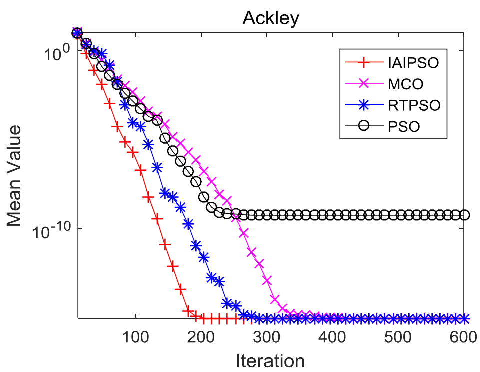
(a)



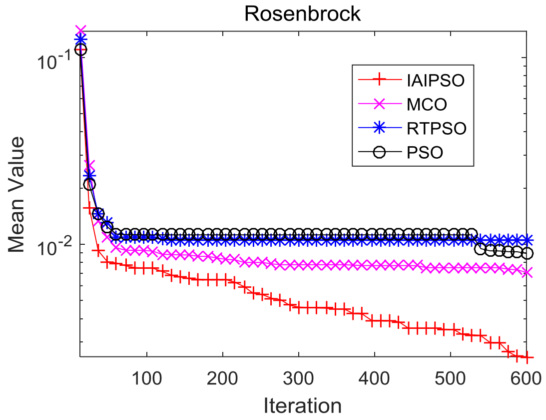
(b)



(c)



(d)



(e)

Fig. 3 The iteration process

Table 3. Comparison of Standard Deviation

Iter		IAIPSO	IPSO	RTPSO	PSO
100	f_1	0.178600594	0.452188903	0.370914734	0.885338152
	f_2	3.4E-14	0.001258124	3.25604E-08	9.6572E-11
	f_3	0.010192821	0.044226826	0.04445793	0.045797276
	f_4	2.06E-13	0.016997584	2.30055E-07	3.547898573
	f_5	0.004137661	0.018504585	0.01280617	0.058576373
400	f_1	1.0e+02	1.0e+02*0.004690282	1.0e+02*0.002481864	1.0e+02*5.43857612
	f_2	1.0e-06	1.0e-06*0.155895871	1.0e-06*0.114800751	1.0e-06*0.554008237
	f_3	0.005465294	0.023382844	0.023164988	0.195310811
	f_4	0	0.808579789	1E-15	3.644680043
	f_5	0.00477341	0.00286046	0.004055025	0.012645445
600	f_1	1.0e+02	1.0e+02*0.003979836	1.0e+02*0.001786005	1.0e+02*1.047678361
	f_2	1.0e-19	1.0e-19*0.002346275	1.0e-19*0.014925721	1.0e-19*0.296521191
	f_3	0.001844897	0.0259047	0.013056287	0.178774041
	f_4	1.0e-08	1.0e-08*0.015841552	1.0e-08*0.001284006	1.0e-08*0.258835613
	f_5	0.004443501	0.004072297	0.0250183	0.071947361

5. Experiment and Analysis

5.1 Experimental Platform

In this section, a P3DX robot is used to further investigation in this experiment, which is provided with indoor navigation. An external laser sensor, named Hokuyo URG, and an onboard PC that is with ROS (Robot Operation System) are carried on P3DX robot. ROS is an open-source robot development framework that is designed by Willow Garage [23]. There are some advantages about ROS, such as node-to-node design, the independent programming language, and open source, etc [24-25]. ROS has been applied to many robotic applications. In Fig. 4, the picture of robot platform is shown, with the PC and laser sensor.

Fig. 4 The robot platform in the experiment

5.2 Experiment process and analysis

(1) Four goals

The multi-goal path planning is begun in two scenarios, four goals and more than four goals. The experiment environment of four goals is shown in Fig. 5(a) and 5(b). There are corner,

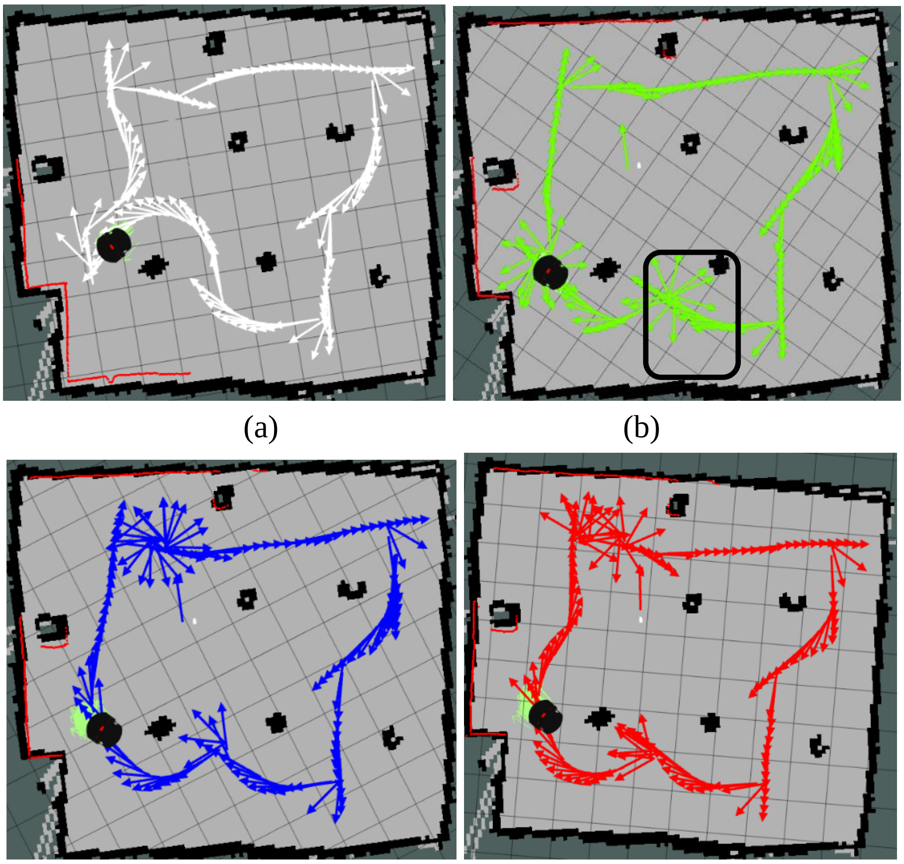
cylinder-shaped obstacle and quadrature obstacle in this environment. The map of four goals is shown in fig. 5(c). S is the starting point, G1, G2, G3 and G4 are the positions of four goals. Black blocks are marked as obstacles, the gray area is free grid. The process of multi-goals path planning is shown in the map by different colors in Figure 6. Robot begins to move from starting point and return to starting point finally, passing G1, G2, G3 and G4.

The multi-goal path planning with IAIPSO is shown by white color in Fig. 6(a). In order to analyze the practicability of IAIPSO in multi-goal path planning, there are three algorithms selected to compare with IAIPSO, such as MCO algorithm, A algorithm and standard PSO algorithm, and their trajectory is shown in Fig. 6(b), 6(c), 6(d). The black square indicates robot is relocating. By contrastive analysis in Fig. 6, it can be obtained that four algorithms can accomplish the multi-goal path planning and avoid the obstacle effectively. The performance, the global search ability and the self-adaptive of IAIPSO are the best in four algorithm.

The experiment of multi-goal path planning is run 30 times with four algorithms. The experiment data is gotten by the P3DX robot, and the mean distance and time are listed in Table 4 (Four goals). From Table 4 (Four goals), the distance and the time acquired by IAIPSO are shortest and least. The performance of IAIPSO in multi-goal path planning is the best in four algorithms. The property of IAIPSO is enhanced.

(a) (b) (c)

Fig. 5 Experiment environment and map of five goals.



(a) (b)

(c) (d)

Fig. 6 The path using different algorithm in scenario 1.

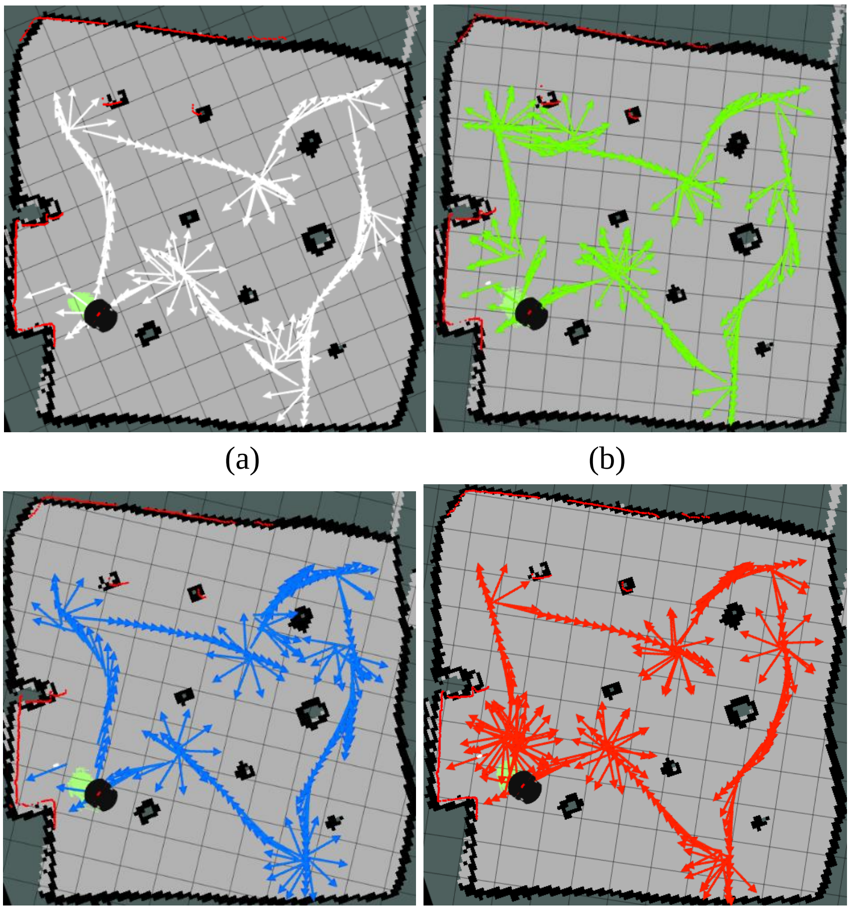
(2) more than four goals

In order to certify the effectiveness of the new method further, another experiment environment that consists more goals and obstacles is set. In this experiment, more than four goals are set and some goals positions are different from (1), the experiment environment is shown in Fig. 7(a) and 7(b). the experiment map is shown in Fig. 7(c). The robot trajectory by IAIPSO is shown in Fig. 8(a)

In the experiment, there are also three algorithms selected to compare with IAIPSO, such as MCO algorithm, A algorithm and standard PSO algorithm. Their trajectories are shown in Fig. 8(b), 8(c), 8(d). From Fig. 8, we can observe that four algorithms can finish multi-goal path planning, avoiding the obstacles. The trajectory is run 30 times with four algorithms. The experiment result is listed in Table 4.

(a) (b) (c)

Fig. 7 The environment and map of more than four goals.



(a) (b)

(c) (d)

Fig. 8 The path using different algorithm in scenario 2.

According to data in Table 4, IAIPSO, MCO, A and PSO can accomplish multi-goal path planning in different environment, but with the complexity of environments increasing, it takes less time for IAIPSO than other three algorithms to finish multi-goal path planning. From analyzing the data in Table 4, we can have a conclusion that IAIPSO not only has a better performance, but also provides a new algorithm for mobile robot multi-goal path planning.

Table 4 Evaluation of IAIPSO and IAPSO in different environment

Algorithms	Four goals		More goals	
	Time(s)	Distance(m)	Time(s)	Distance(m)
IAIPSO	174	11.8	384	13.9
MCO	228	12.2	402	14.0
A	252	13.9	444	14.0
PSO	306	13.9	426	14.0

6. Conclusion

In this research, a hybridization algorithm of IA-IPSO was proposed for mobile robot multi-goal path planning to search a collision free shortest route from starting position, reaching a sequence of goals, to last goal. In IAIPSO, an exponent k is proposed to control the attenuation extent of ω, c_1 and c_2 of PSO; also, the clone selection mechanism is introduced to improve the diversity of PSO. The IAIPSO is used to mobile robot multi-goal path planning that is equipped with ROS. Five benchmark functions are introduced to evaluate the performance of IAIPSO. Simulation demonstrates that the optimal solution of IAIPSO is more accurate and the stability is better than MCO, RTPSO and PSO. In order to evaluate the practicability of IAIPSO in path planning, the IAIPSO is compared with A algorithm, MCO and PSO by P3DX robot, the experimental results show that IAIPSO can plan a optimal route in different environment better than other three algorithms. The optimal route is the shorter and the time consumption is fewest.

In the paper, there is a static relative between the environment, the obstacles and the robot. In the future, we will do the experiment in an extremely complicated environment and add the practical functions. Moreover, we will use IAIPSO in dynamic obstacle environment.

Acknowledgments

Project support by Chongqing Education Commission Foundation under Grant (KJ1400432) and Chongqing Science&Technology Commission Foundation under Grant (cstc2015jcyjBx0066, KJ1600442).

References

[1] X Liu, D Gong. A comparative study of A-star algorithms for search and rescue in perfect maze[C]// International Conference on Electric Information and Control Engineering. IEEE, 2011:24-27.

[2] Q Zhang, M Y Zhao. Minimum Time Path Planning for Robotic Manipulator in Drilling/ Spot Welding Tasks[J]. Journal of Computational Design & Engineering, 2015, 3(2):132–139.

[3] H Wei, B Wang, Wang Y, et al. Staying-alive path planning with energy optimization for mobile robots[J]. Expert Systems with Applications, 2012, 39(3):3559-3571.

[4] 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.

[5] Kovács B, Szayer G, Tajti F, et al. A novel potential field method for path planning of mobile robots by adapting animal motion attributes[J]. Robotics & Autonomous Systems, 2016, 82:24-34.

[6] Hawa M. Light-assisted A path planning[J]. Engineering Applications of Artificial Intelligence, 2013, 26(2):888-898.

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

[8] H Miao, Y C Tian. Dynamic robot path planning using an enhanced simulated annealing approach[J]. Applied Mathematics & Computation, 2013, 222(5):420-437.

[9] Jolly K G, Kumar R S, Vijayakumar R. Intelligent task planning and action selection of a mobile robot in a multi-agent system through a fuzzy neural network approach[J]. Engineering Applications of Artificial Intelligence, 2010, 23(6):923-933.

[10] Aghababa M P. 3D path planning for underwater vehicles using five evolutionary optimization algorithms avoiding static and energetic obstacles[J]. Applied Ocean Research, 2012, 38(5):48-62.

[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.

[12] H Mo, L Xu. Research of biogeography particle swarm optimization for robot path planning[J]. Neurocomputing, 2015, 148(148):91-99.

[13]Y Zhang, D W Gong, J H Zhang. Robot path planning in uncertain environment using multi-objective particle swarm optimization [J], Neurocomputing.2013, 103(2):172–185.

[14] Deepak B B V L, Parhi D R, Raju B M V A. Advance Particle Swarm Optimization-Based Navigational Controller For Mobile Robot[J]. Arabian Journal for Science & Engineering, 2014, 39(8):6477-6487.

[15] Muñoz D M, Llanos C H, Coelho L D S, et al.Hardware opposition-based PSO applied to mobile robot controllers [J]. Engineering Applications of Artificial Intelligence, 2014, 28(1):64-77.

[16] M Wang, J Luo, Walter U. Trajectory planning of free-floating space robot using Particle Swarm Optimization (PSO)[J]. Acta Astronautica, 2015, 112:77-88.

[17] Das P K, Behera H S, Panigrahi B K. A hybridization of an improved particle swarm optimization and gravitational search algorithm for multi-robot path planning[J]. Swarm & Evolutionary Computation, 2016, 28:14-28.

[18] Das P K, Behera H S, Panigrahi B K. Intelligent-based multi-robot path planning inspired by improved classical Q-learning and improved particle swarm optimization with perturbed velocity[J]. Engineering Science & Technology An International Journal, 2015, 19(1):651-669.

[19] De Castro L N, Von Zuben F J. Learning and optimization using the clonal selection principle[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(3):239-251.

[20] X Pu, X Tan. A novel PSO algorithm based on dual group of dispersing and clustering mechanism[J]. Journal of Computational Information Systems, 2015, 11(2):547-557.

[21] H Zhang, Q Hui. Multiagent Coordination Optimization: A control-theoretic perspective of swarm intelligence algorithms[C]// Evolutionary Computation. IEEE, 2013:3339-3346.

[22] Cagnina L C, Esquivel S C, Coello C A C. Solving engineering optimization problems with the simple constrained particle swarm optimizer.[J]. Informatica, 2008, 32(3).

[23] J W Zhang, L W Zhang. Open Source Robot Operating System: ROS [M]. Science Press, 2012.

[24] Cousins S, Gerkey B, Conley K, etc. Sharing Software with ROS [J]. IEEE Robotics & Amp Amp Automation Magazine, 2010, 17(2):12-14.

[25] Zaman S, Slany W, Steinbauer G. ROS-based mapping, localization and autonomous navigation using a Pioneer 3-DX robot and their relevant issues[C]// Electronics, Communications and Photonics Conference (SIECPC), 2011 Saudi International. IEEE, 2011:1-5.