Path Planning of Mobile Robot Based on Simulated Annealing Particle Swarm Optimization Algorithm

Jie Zhao(✉), Xuesong Sheng, and Jianghao Shi

Heilongjiang University of Science and Technology, Harbin, China
zhao_xxsc@163.com

Abstract. In view of the problem of premature convergence of traditional particle swarm optimization (PSO) algorithm in path planning, which is easy to converge to local optimal solution and poor path quality, some theory about the corresponding PSO algorithm of simulated annealing optimization is studied in this paper. While planning the moving path of the robot, it analyzes the effect of initial temperature and cooling coefficient on path length and iteration times from the major contributing factors of simulated annealing algorithm. Thus deduce the law of its change and seek the optimal parameter matching. Simulated annealing algorithm can not only move the updated particle position on the basis of the particle swarm optimization formula, but also select the updated position with a certain probability. The method is used to avoid the particle converging into the local optimal solution in the whole iterative process. The capabilities of the global optimization is strengthened. Compared with the traditional PSO algorithm, the simulated annealing PSO in complex environment has better optimization ability, shorter path and fewer iterations in the simulation results.

Keywords: Particle swarm · Path planning · Simulated annealing · Linear inertia weight

1 Introduction

The path planning of mobile robot has always been the focus of the research in the development of intelligent robot. It is a major task for a moving robot that its travel path has the lower energy consumption, shorter distance and less time. It is necessary to avoiding all obstacles based on the origin and destination point coordinates in the work environment with obstacles.

There are many existing path finding methods, including artificial potential field, visual graph, etc. But these methods have some limitations, such as artificial potential

This work is supported by basic scientific research projects of Heilongjiang University of Science and Technology "The research on visual servo control of mine driving section forming based on image".

X. Jiang and P. Li (Eds.): GreeNets 2020, LNICST 333, pp. 131–139, 2020.
https://doi.org/10.1007/978-3-030-62483-5_14

field method can not find the path in front of the near obstacles, visual graph method has complicated search path and low efficiency [1].

In recent years, many experts and scholars have studied some new algorithms, such as genetic algorithm, ant colony, bee colony and so on. Compared with some traditional algorithms, an intelligent bionic algorithm is proposed by simulating one or several behaviors of natural creatures, which provides a new way to the path ending in complex environment [2].

PSO has the benefits of fast searching speed, easy accomplishment and simple programming language. Due to its own limitations, such as being very possible to shrink to local optimum solution and being greatly affected by complex constraints in some problems, its search results are not ideal. To prevent premature maturation to the point of convergence of population, an improved PSO algorithm is proposed in this research. PSO uses linear adaptive inertia weight factor and simulated annealing algorithm to optimize, which improves the convergence of PSO. PSO combined with simulated annealing algorithm can make particle swarm jump out of the local optimal point. So the global optimal or approximate optimal point independent of the initial point selection can be accomplished [3]. Through the simulation of robot path planning, the result is analyzed and compared, which shows the effectiveness and rationality of PSO algorithm.

2 PSO Path Planning

2.1 The Basic Principle of PSO

Based on the study of predatory behavior of birds, the path planning of PSO algorithm is proposed. When solving the optimization value, the solution to algorithm corresponds to the flight location of a bird in the searching space, so it becomes a “particle”. Every particle lies in a different place and at a different velocity. The fitness value of the function is determined to judge quality of a particle [4]. Each particle continuously updates itself by tracking the influence of the individual optimal location I_{best} and the global optimal location B_{best} of the whole population, so as to generate the next round of new population [5]. The velocity and position update formulas of particles are as follows:

$$T_i^{k+1} = wT_i^k + c_1 rand(I_{best} - X_i^k) + c_2 rand(B_{best} - X_i^k) \tag{1}$$

$$X_i^{k+1} = X_i^k + T_i^{k+1} \tag{2}$$

Where: T_i^k is the k^{th} dimension speed in the i^{th} iteration; X_i^k is the k^{th} dimension location in the i^{th} iteration; w is the inertia weight; c_1 and c_2 are learning factors; ‘rand’ is the random number distributed between 0 and 1.

Particle path planning is to search for the shortest path without collision from start to end. Its function can be expressed

$$F = \sum_{i=0}^{m} \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \tag{3}$$

In the basic PSO algorithm, w is a fixed value. When it is large, the particle's global search ability is strong, which may make particle fly over the lowest point. When it is small, it can make the local search ability of particles stronger. In this way, particles tend to convergence to local optimum and lose the global search ability. PSO algorithm usually consists of three parts: the velocity of particles in the last time, the cognitive part of particles, and the social cognitive part of the whole population. The value of inertia weight affects the flying velocity of particles. Learning factors can regulate self cognition and social cognition. Therefore, there is an important relationship between the inertia weight and the change of learning factors. In view of the above defects, a group of scholars represented by Kennedy proposed an algorithm to change the inertia weight through the linear decreasing law [6], that is, with the iterative process, the value of the inertia weight is constantly reduced. The algorithm has a strong exploration ability in the early stage and can search in a larger space range. In the later stage, the reduction of weight value makes it possible to convergence to a local region for more meticulous search. The weight variation formula is as follows [7]:

$$w = w_{\max} - \frac{(w_{\max} - w_{\min})}{iter_{\max}} i \tag{4}$$

Where: $w_{\max}$ is maximum; $w_{\min}$ is minimum; $iter_{\max}$ is the maximum value; i is the current value of iterations of particles. The program flow is similar to the basic PSO, but the final step determines whether the algorithm meets the termination conditions. If not, adjust the value w and replace the particle speed and location to continue the cycle. However, with the different optimization problems, the adjustment strategies of linear inertia weight are also different. So the linear inertia weight have great limitations in application.

2.2 The General PSO Process

The PSO executing steps are as follows:

Step 1: set initialization scale, learning factor, maximum number of iterations and other related parameters;
Step 2: Initialize the velocity and location of the individual, record the optimal location of each particle currently searched as I_{best}, and define the optimal location of particle global searched as B_{best};
Step 3: calculate the objective function of each particle, such as fitness, and save its particles and fitness values;
Step 4: update the particle's velocity, replace the particle's location according to the constraint conditions, initialize the particles beyond the boundary again, and then calculate its fitness value according to the optimal positioning in the process of historical optimization and replace the previous position;
Step 5: check whether the particle can meet the stop condition. If it meets the condition, it will stop and output the optimal value result of the algorithm program. If it does not meet the condition, it will return to the third step to continue to update the particle velocity and position.

3 PSO Path Planning Based on Simulated Annealing

3.1 The Basic Principle of Simulated Annealing

The algorithm of simulated annealing was first proposed by Metropoli. The idea of the algorithm comes from the annealing process, that is, first heating the solid to a higher temperature, and then gradually making the temperature lower. When heating, the internal particles are in a disordered state. When cooling, the particles become orderly and reach equilibrium at each temperature. At room temperature, the energy can reach the minimum value. Therefore, the algorithm has good global search ability, high efficiency and easy to understand program.

The simulated annealing algorithm can generate new solutions randomly between the given solution and its local domain. Through the metropolis acceptance criterion, the solution with better adaptability can be accepted, or the solution with worse adaptability can be accepted within a certain probability. In the process of particle swarm optimization, because the local optimal solution generated by each iteration of the algorithm does not necessarily meet the constraint conditions of the problem, so by introducing the local optimal solution for optimization search, we expect to get more local optimal solutions that meet the constraint conditions and have better adaptive values. A better global optimal solution is generated, which leads to the evolution of the population [8].

The solution process of simulated annealing algorithm is as follows [9]. The initial control parameters are set as follows: enough initial temperature, initial iterative solution, cycle counter, maximum number of iterations of algorithm, decay function of temperature and end criterion of program; The first random disturbance is carried out to generate a new solution X_N. According to the objective function value of the new solution, and then the comparison, if $f(X_N)$ less than $f(X_0)$, the new solution X_N is accepted. Otherwise calculate the selection probability, if it meets the requirements, it will also accept the new solution. According to the desuperheating function, carry out the desuperheating operation, and reach the end criterion to end the desuperheating operation, otherwise compare again.

3.2 The PSO Algorithm with Simulated Annealing

In the original PSO in order to prevent the particles from generating large offset, the flight velocity of each particle will be controlled within a range. If a particle flies to a better location than the current position, then later iterations will be carried out at that position. At this time, the simulated annealing algorithm is added to make particle jump out of the local optimal solution. After each particle flies to the latest position after iteration, the fitness value of the particle is calculated. If the fitness value is greater than the previous position, the particle moves to the new position. If it is not better than the current position, calculate the change value of fitness to judge whether it is greater than the annealing value. If it is greater than the annealing value, the particle moves to the new location to complete annealing operation [10]. The algorithm can make every particle be annealed. The updated particle position can not only move according to the optimization formula of PSO, but also select the updated position with a certain probability. This characteristic of accepting the new solution with a certain probability can effectively avoid the local search, thus greatly improving the search performance of particle swarm algorithm [11].

Here its detailed steps is.

Step 1: Initialize the following parameters: the scale of the initial population, the learning factors c_1 and c_2, the position x_i and velocity v_i of particles, and the initial temperature T_0 of the maximum value of iterations $iter_{\max}$;
Step 2: Store location and fitness value in the individual extremum, and calculate the fitness value according to the objective function;
Step 3: Update corresponding information of each particle

$$T_i^{k+1} = wT_i^k + c_1 rand(I_{best} - X_i^k) + c_2 rand(B_{best} - X_i^k) \tag{5}$$

$$X_i^{k+1} = X_i^k + T_i^{k+1} \tag{6}$$

$$w = w_{\max} - \frac{(w_{\max} - w_{\min})}{iter_{\max}} i \tag{7}$$

Step 4: Calculate the updated fitness value. If it is more superior than the previous particles, update the values of particles P_{id} and the values of groups P_{gd}. Otherwise, carry out the next step;
Step 5: Add a disturbance near the particle to make the particle generate a new solution X_N.
Step 6: Carry out desuperheating operation and renewal formula of temperature, $T(k+1) = \alpha * T(k)$, $k = 0, 1, 2 \ldots\ldots$, where α is a number close to 1 and the end temperature is set to 0.2;
Step 7: Judge whether the function reaches the number of termination iterations. If it reaches the number, stop the program and output the last iteration result, which is the optimal extreme point and optimal value. Otherwise, skip to step 3 to continue the iteration.

4 Study on the Influence of Simulated Annealing Parameters

The optimal configuration of important parameters in the simulated annealing algorithm is very important for path planning, including the initial temperature and annealing coefficient. The setting of these two parameters plays a decisive role in whether algorithm can achieve ideal results.

4.1 Initial Temperature

The initial temperature may influence the global optimal solution of the system. The pros and cons of the algorithm are determined by the selection probability, the denominator of which is the initial temperature. An important step of simulated annealing algorithm is to design a reasonable initial temperature. If its probability is not reasonable, it is likely that it will get a global optimal solution. Generally, a large initial temperature is set at the beginning of the algorithm, but too high temperature will make the calculation time longer.

This paper selects the initial temperature as low temperature, medium temperature and high temperature to carry out simulation experiments and compares the effects of different initial temperature on the results. Here, when the temperature is 800, 4000, 25000, 35000 to carry out five simulation experiments under the complex environment map with annealing coefficient of 0.95, the data is recorded and the average value of each group of data is curve fitted (Fig. 1).

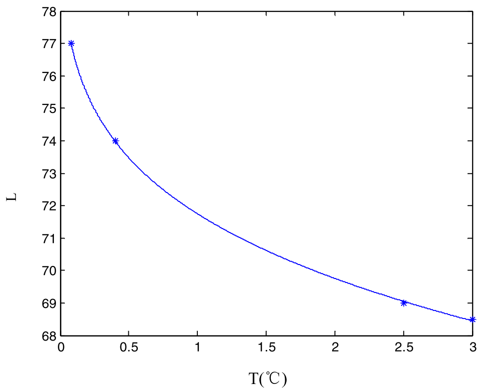

Fig. 1. Path length at different initial temperatures

It can be seen that at the initial temperature of high temperature, the average path length can get better value than at low temperature and medium temperature, and the result at 25000 °C is basically consistent with that at 35000 °C. So when the initial temperature is set to high temperature, better simulation results can be obtained.

Most of the reasons for this effect come from the most important judgment basis of simulated annealing algorithm $\exp(f(X_0) - f(X_N)/T) < randm$. The difference between the new solution and the original solution will be adopted within 0.69 times of the initial temperature. And because the initial temperature will become smaller with the iteration, the annealing is realized.

4.2 The Cooling Coefficient

The cooling coefficient is also an important parameter. Only by setting a reasonable desuperheating coefficient, the algorithm can slowly cool down like the solid annealing, so that the balance can be reached at each temperature. Therefore, the setting of the cooling coefficient is usually between 0.90 and 0.99. The smaller the value is, the faster the cooling is. But the algorithm can not fully optimize.

However, when the initial temperature is set high, the temperature is still large when reaching the limit of iterations times, which may not be conducive to the later local search. Set the initial temperature as 25000 °C, and the coefficient of desuperheating as

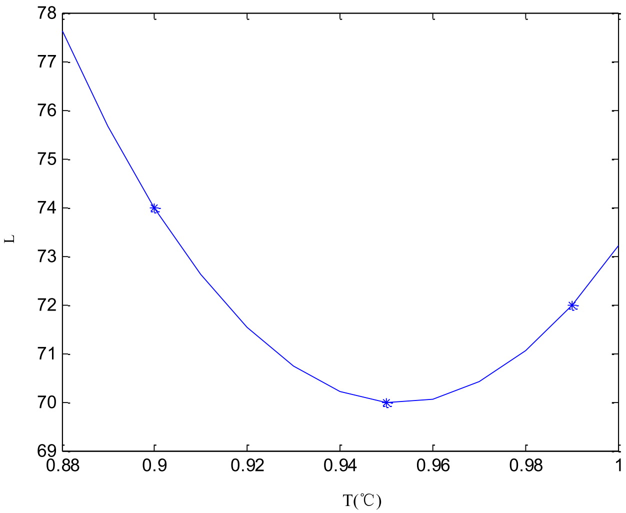

Fig. 2. Path length under different cooling coefficient

0.90, 0.95 and 0.99 to carry out five simulation experiments and make the average value of the data into a fitting curve as shown in Fig. 2.

It can be seen that in the case of selecting high temperature, with the increase of annealing coefficient, the path length of simulated annealing particle swarm optimization algorithm first decreases and then increases. Therefore, we consider adding an exponential annealing coefficient $\alpha = \exp(-CK^{1/N})$, where C is constant set to 2, K is the maximum value of iterations, and N is quantity of parameter. In this way, annealing coefficient has a large value in the beginning, and the slow cooling makes the annealing temperature have a better search for the whole at a high temperature. In the later stage, the value is smaller, and the annealing speed is faster, which can reach a smaller temperature and strengthen the local search.

It can avoid the situation that the simulation result is not ideal when setting a higher desuperheating coefficient at high temperature. Finally, through the simulation experiment, the path length obtained by using this exponential type of temperature coefficient is better than that obtained by using constant temperature coefficient.

In conclusion, the particle swarm optimization algorithm of simulated annealing chooses the initial temperature as 25000 and the annealing coefficient as index.

5 The Global Path Planning Simulation and Result Analysis

The grid method is used to build the environment map. In the environment, the above PSO and the simulated annealing PSO after parameter matching are used to plan the path, and the ability of the two methods in the complex environment is compared and analyzed.

The initialization settings of the parameters are m = 200, $iter_{\max} = 500$, $c_1 = 1.7$, $c_2 = 1.8$, $T_0 = 25000$, $\alpha = \exp(-CK^{1/N})$. The simulation results are shown in Figs. 3 and 4.

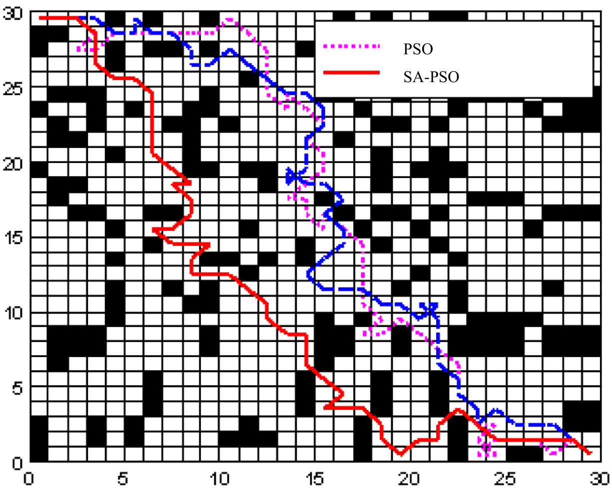

Fig. 3. Simulation of each algorithm in complex environment

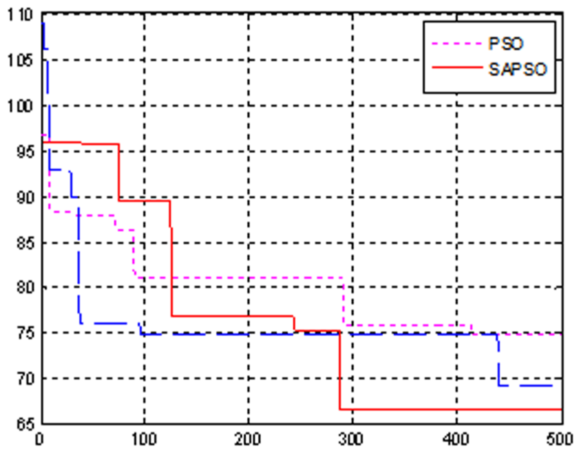

Fig. 4. Iterative curve of each algorithm in complex environment

After five times of simulation, the simulated annealing particle swarm optimization algorithm can finally get a smaller objective function, namely the path length, which is 66.4975.

In the complex case, the simulated annealing PSO iterates to the optimal value as soon as possible, and the objective function value of the path length is the minimum.

From the above simulation chart and convergence curve, we can know that in the complex environment, the basic particle swarm optimization algorithm after iterating 421 times falls into the local optimal solution, the optimal path fitness function is 74.8112, and the inflection point is 39. The PSO algorithm of linear inertia weight gets the local optimal solution in 442 iterations, the function of path is 69.1534, and the inflection point is 41.

When the simulated annealing PSO iterates 290 times, it jumps out of the local optimal value. Under the current maximum number of iterations, the most effective path fitness function is 66.4975, and the inflection point is 35. The turning points involve the turning times of the robot. It can be seen from the path above that the simulated annealing PSO algorithm is more smooth than the other two algorithms.

6 Conclusion

Traditional PSO has the advantages of simple modeling, simple calculation process, fast convergence and less parameters for robot path planning, but it is also easy to fall into the local optimal solution, so it can not get the global optimal solution in some complex environments, which makes the optimization result not ideal. The PSO algorithm combined with simulated annealing can make up for the above shortcomings. In conclusion, the simulated annealing PSO algorithm has stronger search ability in complex environment, better path and shorter path length and the algorithm is effective and reasonable.

References

1. Zhao, Z., Lin, Y.: Mean particle swarm optimization algorithm based on adaptive inertia weight. Comput. Eng. Sci. **38**(2), 501–505 (2016)
2. Lozano-Perez, T., Wesley, M.: An algorithm for planning collision free paths among polyhedral obstacles. Commun. ACM (5), 436–450 (1975)
3. Li, W., Zhao, D.: Whole area robot based on grid method and neuron covering algorithm. Mech. Des. Manuf. **08**, 232–238 (2017)
4. Montiel, O., Sepúlveda, R., Orozco-Rosas, U.: Optimal path planning generation for mobile robots using parallel evolutionary artificial potential field. J. Intell. Robot. Syst. **79**(2), 237–257 (2015)
5. Liu, X.: Particle swarm optimization and its application development. Sci. Technol. Econ. Guide **26**(04), 138–139 (2018)
6. Shi, Y., Eberhart, R.C.: A modified particle swarm optimize. In: Proceedings of IEEE Congress on Evolutionary Computation, Anchorage, pp. 69–73 (1998)
7. Erdem, H.A.: Solving container loading problem with genetic algorithm. In: 15th International Symposium on Computational Intelligence and Informatics, pp. 391–396. IEEE, Budapest (2014)
8. Wang, R., Xiao, H., Mo, Q., Huang, G.: Multi base station cooperative power allocation based on simulated annealing algorithm. J. Guilin Univ. Electron. Sci. Technol. **37**(05), 350–354 (2017)
9. Li, J., Xu, R.: Research on the combination of annealing algorithm and neural network algorithm in path planning. Autom. Instrum. **32**(11), 6–31 (2017)
10. Sun, B., Zhu, D., Yang, S.X.: An optimized fuzzy control algorithm for three-dimensional AUV path planning. Int. J. Fuzzy Syst. **20**(2), 1–14 (2018)
11. Guo, X., Zhang, B.Y., Feng, G.H.: Multi-objective optimization of low-speed and high-torque direct-drive permanent magnet synchronous motor based on hybrid particle swarm optimization. J. Mech. Electr. Eng. **35**(11), 1214–1219 (2018)