ATLANTIS PRESS

Advances in Engineering Research, volume 149

2nd International Conference on Mechanical, Electronic, Control and Automation Engineering (MECAE 2018)

Particle Swarm Optimization Algorithm Based on Artificial Potential Field

Dianna Song [a], Jianhua Qu [b]

School of Shandong, Normal University, Shandong 250000, China;

[a] 568875103@qq.com, [b] 112205193@qq.com

Keywords: Particle swarm optimization, artificial potential field.

Abstract. Artificial potential field method is a simple and effective path planning algorithm. In this paper, the basic idea of artificial potential field method is inherited. The gravitational potential field and repulsive potential field are introduced into particle swarm optimization. The gravitational potential field is used to enhance the optimization of particles. The repulsive potential field is used to increase the search range of particles to prevent the particles from falling into the local excellent solution. This paper tests the function, experiments show that this method is effective.

1. Introduction

Due to its simplicity and high efficiency, artificial potential field method has been widely used in robot path planning in the past ten years. The basic idea of the artificial potential field is that the robot in the environment is affected by the attraction potential field from the target point and the repulsion potential field from the obstacle. The robot decides its motion information according to the influence of the co-potential field formed by the attractive potential field and the repulsive force field in the environment.

Eberhart and Kennedy proposed a population-based particle swarm optimization algorithm in 1995, which is a random search algorithm. The source of this algorithm is inspired by bird foraging behavior. Targeting food, the position of birds in space is considered a variable, and the birds in the group regard their mass and volume as one particle. The particle has a certain initial velocity, the particles move irregularly in space, and the movement records the best location and the entire population through which each particle passes. Each particle moves in the best position the individual has passed and the optimal position of the population that updates their position and velocity through the collaboration and competition among population particles causes the crowd to move towards the fitness group and increase the optimal solution.

In this paper, the gravitational potential field and the repulsive potential field in the theory of artificial potential field are introduced into the particle swarm optimization algorithm, and the particle optimization range is increased by the virtual force generated between the particles to prevent the particle from falling into the local optimal value. The result analysis proves its feasibility and effectiveness.

2. Standard Particle Swarm Optimization and Artificial Potential Field

2.1 Standard Particle Swarm Optimization

In Particle Swarm Optimization, the particle is a dimension vector, and the size of the dimension is set according to the problem. Multiple particles form a population. The moving distance of the particle in n-dimensional space is called speed. The set objective function is called Fitness, the fitness of particles in the process of updating itself is called the optimal history of the individual, and the position where the fitness of the whole population is the best is called the global optimum of the population.

The mathematical description of Particle Swarm Optimization is as follows. The size of the particle search space D is related to the space required by the problem. The size of the population N, generally the larger the population, the better the PSO algorithm is. The population expansion algorithm becomes more and more complex, it will increase the solution time and not conducive to solving. If

it made relatively small, information collaboration between groups will become weak, so the size of the population will choose a suitable value.

The location of the first particle can be described as $X_i = (x_{i1}, x_{i2}, \ldots x_{id})$. The velocity represents the size of particle movement. $P_{best} = (p_{i1}, p_{i2}, \ldots p_{id})$ Represents the best history of particle i and $p_g = (p_{i1}, p_{i2}, \ldots p_{id})$ indicating the optimal location of the population. The standard particle velocity and position update according to the following formula:

$$v_{ij}(t+1) = w * V_{Ij}(t) + c_1 * r_1 * \left(p_{best}(t) - x_{ij}(t)\right) + c_2 * r_2 * \left(p_g(t) - x_{ij}(t)\right) \quad (1)$$

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

Where $i = 1,2,3, \cdots N$, ω is the inertia factor, c_1 is the degree of influence on the velocity of the optimal position passed by itself and $r_1, r_2 \in [0,1]$ is the random number , c_2 is the degree of the influence of the global optimum position on the velocity. t is the number of iteration. Generally, in order to control the particle search range, Set the position and speed of the range, to avoid particles without border flight.

The standard particle swarm optimization process can be described as follows:

Step1: Set the initial value: Set the particle's original position and velocity, and determine the acceleration factor C_1, C_2, inertia factor ω, set speed and position range to determine the particle's search range.

Step2: According to the objective function to determine the fitness of each particle; in the solution process, set the objective function to evaluate the merits of more solutions, as a basis for particle swarm update.

Step3: Perform the best-case-by-case and group-best global updates based on formulas (3) and (4):

$$P_{best}(t+1) = \begin{cases} X_i(t+1) & \text{if } fit\left(X_i(t+1)\right) < fit\left(P_{best}(t)\right) \\ P_{best}(t) & \text{others} \end{cases} \quad (3)$$

$$P_g = \min\left\{P_1(t+1), P_2(t+1), P_{,3}(t+1), \cdots P_n(t+1)\right\} \quad (4)$$

Step4: Update the velocity and position of the particles according to formula (1) and formula (2), and restrict them to the specified range.

Step5: If the termination condition satisfied, the algorithm ends and output P_g. Otherwise, return to Step2 to re-run the algorithm.

2.2 Virtual Potential Field Algorithm

The artificial potential field method can be expressed by different potential field functions, but the basic principles are similar. The most commonly used potential field method is the gradient potential field method. Using the negative gradient of the potential field of the robot. By the virtual force, the target point is attractive to the robot, the obstacle generates repulsion to the robot, and the robot moves from the starting point to the target point under the effect of the resultant force.

In the environment, q represents the position of the object, qr represents the target position, and U(q) represents the virtual potential field. Then the virtual and virtual repulsion potential fields are U_{att} (q) and U_{rep} (q) respectively.

The robot's virtual potential field and virtual force at a certain point are (the virtual force is obtained by differentiating the virtual potential field) as follow.

$$U(q) = U_{att}(q) + U_{rep}(q) \quad (5)$$

$$\vec{F}(q) = -\nabla U(q) = -\nabla U_{att}(q) - \nabla U_{rep}(q) \quad (6)$$

The potential field for defining obstacles and target points using the electrostatic field potential model is shown below.

$$U_{att}(q) = \frac{1}{2}\xi\rho_g^2(q) \quad (7)$$

$$U_{rep}(q) = \begin{cases} \frac{1}{2}\eta\left(\frac{1}{\rho(q)} - \frac{1}{\rho_0}\right)^2, \rho(q) \le \rho_0 \\ 0, \ \rho(q) > \rho_0 \end{cases} \quad (8)$$

$\rho_g(q) = \|q - q_g\|$ is the distance between the robot and the target point; ξ and η is the proportional coefficient; ρ_0 is the maximum range of the repulsive force of the obstacle; $\rho(q)$ is the minimum distance from the robot to the obstacle.

3. Particle Swarm Optimization Algorithm Based on Artificial Potential Field

The artificial potential field is introduced into particle swarm optimization algorithm. After initializing the particle position, the particles are prevented from accumulating too much and the particle distribution is more uniform. In the process of particle searching and location updating, the gravitational and repulsive forces are introduced, and the global optimum is set as the anchor node, that is, the target point at this time, whose position is invariable, attracts the surrounding particles within a certain range. At the same time, all the particles except the anchor nodes are prevented from falling into local optimum by the repulsive force.

Algorithm flow is as follows.

Step1: Initialize the particle swarm and set the number of particles, exclusion distance D and attraction distance d. Generate initial position and initial velocity randomly.

Step2: Calculate the distance between particle i and particle j (Sij). If Sij <D, produce repulsive force, the particles mutually exclude the same distance, update position.

Step3: Calculate the objective function of each particle, find the current global optimum value, set the global optimal position as the anchor node a. Calculate the distance between each particle and anchor node (Sia) and the distance between the other particles except the anchor node (Sij).

Step4: If Sia <d, the anchor node is attractive to particle i. If Sij <D, it will produce repulsive force and particles mutually exclude the same distance.

Step5: According to the formula (9) and (2) to update the speed and location which is limited within the specified range.

$$v_{ij}(t+1) = w * v_{Ij}(t) + c_1 * r_1 * \left(p_{best}(t) - x_{ij}(t)\right) + c_2 * r_2 * \left(p_g(t) - x_{ij}(t)\right) + c_3 * r_3 * vf_{ij}(t) \quad (9)$$

Step6: If the termination condition satisfied, the algorithm ends and output the global optimum value. Otherwise return to Step3 to re-run the algorithm.

4. Experiment and Result Analysis

In order to verify the effectiveness of the proposed algorithm, the algorithm is compared with the standard particle swarm optimization to solve four test functions. In order to facilitate the programming and display the results of the algorithm, all the experiments in this paper are running in the version of matlab7.0, using matlab language, the same HP machine running under the same environment, the machine is configured for Win7 operating system, The processor is Intel i3, 2.10GHZ, video card size is 1G, memory size is 2G.

The setting of the parameters in the algorithm has a great influence on the effect of the algorithm. By consulting a large amount of literature and conducting simple experimental tests, the parameters in the algorithm are set as follows: the size of the population $ps = 30$; in order to make the algorithm have an expanded search range in the initial stage , The inertia factor is updated with the following formula $\omega = \omega_{max} - (\frac{k*(\omega_{max}-\omega_{min})}{Maxnum})$, ω_{max} set to 0.9 and ω_{min} set to 0.4; the parameters C_1, C_2, C_3 are set to the same number $C_1 = C_2 = C_3 = 2$; the iterations of PSO are 100, 300, 500, 1000, and 2000. In order to ensure the fairness of the result, each algorithm runs 100 times independently.

The comparison results are shown in Table 1. It can be seen from the experiment that as the number of iterations increases, the running result of the algorithm gradually approaches the optimal value. In the optimization process, the VFPSO converges faster and the effect is obvious. However, the accuracy of the final result The effects of force may decline, requiring better control.

Table 1. Algorithm iteration comparing

Iterations	Algorithm	Sphere	Griewank	Rastrigin	Rosebrock
100	VFPSO	79.1	0.92	287	98202
	PSO	236.8	2.9	716	69928
300	VFPSO	28.7	0.66	241	20154
	PSO	49.9	1.4	388	68954
500	VFPSO	19.7	0.55	211	12054
	PSO	23.05	1.2	315	67924
1000	VFPSO	11.5	0.39	169	3984
	PSO	6.13	1.04	248	67194
2000	VFPSO	6.9	0.32	128	2145
	PSO	1.67	0.84	193	67682

5. Summary

In this paper, artificial potential field algorithm and particle swarm optimization are combined to introduce the idea of force interaction between charges into the particle optimization process, and the optimal particle is produced by using the optimal particle as the target point and particle in the iteration process Repulsiveness avoids the algorithm getting into the local optimum and finally finding the optimal solution. In this paper, the validity of the algorithm is verified through experiments. The convergence is obviously better than the standard particle swarm optimization algorithm. But in the process of algorithm implementation, the final result is not accurate enough due to the influence of the artificial potential field, which may cause the particle to oscillate.

References

[1]. ZHANG W, MA D, WEI J, et al. A parameter selection strategy for particle swarm optimization based on particle positions[J].Expert Systems with Applications, 2014, 41(7):3576-3584.

[2]. KUNDU R, DAS S, MUKHERJEE R, et al. An improved particle swarm optimizer with diference mean based perturbation [J]. Neurocomputing, 2014, 129(10):315-333.

[3]. Lu S, Yu S. An improved particle swarm optimizer with attraction and repulsion[C]// International Conference on Computing and Convergence Technology. IEEE, 2013:735-740.

[4]. Howard A, Matarić MJ, Sukhatme GS. Mobile sensor network deployment using potential field: A distributed scalable solution to the area coverage problem. In: Proc. of the 6th Int'l Sympon Distributed Autonomous Robotics Systems (DARS 2002). 2002.299-308.

[5]. Ji Zhen, Liao Huilian,Wu Qinghua. Particle Swarm Optimization Algorithm and Its Application[M]. Beijing:Science Press, 2009

[6]. WANG X, WANG S, BI D. Virtual force - directed particle swarm optimization for dynamic deployment in wireless sen- sor networks[J].Lecture Notes in Computer Science, vol. 4681, pp. 292-303.

[7]. Rahman A U,Alharby A,Hasbullah,et al. Corona based deployment strategies in wireless sensor network: A survey [J]. Journal of Network & Computer Applications,2016,64(C):176-193

[8]. LIU Bo. Application of Particle Swarm Optimization Algorithm and Its Engineering [M]. Beijing: Publishing House of Electronics Industry, 2010.

[9]. Li Li, Niu Ben. Particle Swarm Optimization Algorithm [M]. Beijing: Metallurgical Industry Press, 2009.

[10]. Xiangyuan Jiang and Shuai Li, BAS:Beetle Antennae Search Algorithm for Optimization Problems,201710.10724.