Improved PSO Algorithm and its Application in Function Optimization

ZHANG Dun-Li

Mechanical and Electrical Engineering Department Hunan Radio and TV University Changsha, China E-mail: 11506043@qq.com.cn

Abstract- To solve the problem in function optimization using Particle Swarm Optimization algorithm (PSO), this paper propose an improved PSO algorithm adopting gradient information. There are some merits such as simplicity, high effectiveness and so on about function optimization using PSO algorithm. But some studies shows that some shortcomings such as slow computing speed, easiness to fall in local peak in large scale problem are appeared in the optimization, as is determined by the randomness of the algorithm. The grad algorithm is a type of traditional optimization way and has the characteristic which is along the descending grad direction of optimization values. So the grad method can reduce the time for the optimization values because the direction for optimization values is determined by the grad of grad algorithm. In order to overcome the disadvantages of the standard PSO algorithm, the principle of grad method was used in PSO algorithm. Therefore, the Grads-PSO algorithm (regulated by grad method) was proposed in this paper. The Grads-PSO has been compared by the ones of the standard PSO algorithm. The simulation results show that the improved PSO algorithm increase the computing speed than the standard **PSO** algorithm.

Key words: grads algorithm, improved PSO, function optimization

I. INTRODUCTION

PSO (Particle Swarm Optimization) is a new optimization algorithm, which was proposed by the Kennedy

ZHOU Guo-dong Mechanical and Electrical Engineering Department Hunan Radio and TV University Changsha, China E-mail: zgd79@qq.com.cn

and Eberhart in 1995 for the first time. It has a lot of merits such as simplicity, high effectiveness and so on contrasted with genetic algorithms (GA), evolutionary programming (EP) algorithms. At present, researches shows that the PSO algorithm has achieved better results in optimization.

But the PSO algorithm, as long as other evolutionary algorithm appears shortcomings that are calculated a long time, easy to fall into local minima. The gradient method, one of the traditional optimization methods can find extreme along the negative gradient direction, as reducing the optimization time, quickly find a global optimum. This article will combine gradient method with PSO algorithm to design the gradient method (G-PSO). Validated by function optimization, compared with the conventional PSO algorithm, a conclusion can be drawn: G-PSO optimization can reduce computing time and improve the optimization accuracy.

II. STANDARD PSO ALGORITHM

PSO algorithm was proposed by Kennedy and Eberhart, originated in the simulation of a simple society, originally conceived as a simulation of birds foraging in the process, but later used as a good optimization tools. PSO optimization algorithm is similar to other evolutionary algorithms, optimized parameters combined into groups, by the degree of adaptation to the environment, the individuals of population will move to better areas. Apart from other evolutionary algorithms, Particle will approach the target in a certain speed, which is described as an optimal particle without individual volume in a D-dimensional search space, considering the best self-location and historical best position information of particles. X_i may be expressed micro particles as dimensional vector, $X_i = [x_{il}, x_{i2}, ..., x_{iD}]$, V_i represents a particle velocity $V_i = [v_{il}, v_{i2}, ..., v_{iD}]$, the fine particles experience optimum position (corresponding to the best fitness) is represented as $P_i = [p_{il}, p_{i2}, ..., p_{iD}]$ which also are known as P_{best} , preferably all of the group of particles through the position of the index number are represented by g, denoted as P_g , which also are called g_{best} , the number i particles evolved from n to n + 1 and substituting is updated by the following formula.

$$v_{id}^{n+1} = wv_{id}^{n} + c_1 r_1^{n} (p_{id}^{n} - x_{id}^{n}) + c_2 r_2^{n} (p_{gd}^{n} - x_{id}^{n}), \quad (1)$$

$$x_{id}^{n+1} = x_{id}^n + v_{id}^{n+1},$$
(2)

Where, i = 1,2, ..., m is the number of particles; W represents the inertia weight, so that particles remain inertia, which will expand the search space to help searching new zone; c_1 , c_2 are positive real numbers, which are called the acceleration constant as means statistical acceleration weight of each particle toward the right P_{best} and g_{best}; r_1^n and r_2^n are random number interval [0,1].

III. IMPROVED PSO

In order to accelerate the convergence of particle swarm optimization, take advantage of traditional numerical optimization methods in speed and accuracy of the calculation, the article mixes particle swarm algorithm with gradient method, constitute the G-PSO algorithm with gradient search factors. Concrete realization of ideas are as follows : as is shown in Figure 1, assuming that the optimization function f has a global minimum in the domain space S, and the optimization function f can achieve global minimum f(g) in the center point of g if circle A (radius r) the which contains the optimal area. For each particle in the near global optimal value, we can use the gradient method to update the location. And away from the global optimum value, the location will be updated by PSO algorithm. The merits brought by that change is reducing the calculating time for the random position replacement. Once the particles come

into the optimal area, a gradient method regressive will be actived which can get a new location if it is the best position of the particle swarm currently and can guide other particles to update as soon as possible to the optimal position to accelerate the speed of optimization and improve efficiency also.

Due to the gradient method, the single particle into the area making is not again far from optimal area because of the pure randomness of PSO, as improve the efficiency of single optimization. Although that lost some randomness and flexibility, it will enhance the overall optimization algorithm efficiency.



Figure 1. Illustration of optimization problem base on grads-PSO

According to the above description, the improved speed and location updating equation of G-PSO algorithm is (3):

$$\begin{cases} v_{id}^{n+1} = wv_{id}^{n} + c_1 r_1^{n} (p_{id}^{n} - x_{id}^{n}) + c_2 r_2^{n} (p_{gd}^{n} - x_{id}^{n}), f(x_{id}^{n}) > f(g) + r \\ x_{id}^{n+1} = x_{id}^{n} + v_{id}^{n+1} \end{cases}$$

$$x_{id}^{n+1} = T(x_{id}^{n}), f(x_{id}^{n}) \le f(g) + r$$
(4)

Where, $T(x_{id}^n)$ represents the gradient optimization method to get the next point arithmetic.

IV. G-PSO ALGORITHM IMPLEMENTATION PROCESS

Based on the above description, G-PSO algorithm to achieve simple steps is described below:

(1) Setting the initial parameters such as population size m; inertia weight w; acceleration constants c1, c2; optimization domain range of parameter θ ; condition of algorithm ends; optimal area radius r and so on. (2) Initialize groups of the particles, including the initial position and initial velocity of the particles.

(3) Evaluate of the fitness function fitness value of each particle.

(4) For each particle, update P_{best} with the current fitness value if the current fitness value is better, comparing with the best historical position Pbest.

(5) Update the best fitness value g_{best} comparing with value of each particle group's experienced historical best position. If the current group is preferable, use it as the new g_{best} . At the same time, record the index.

(6) Accord to equation (2) to update the position and velocity of each particle.

(7) Calculate the optimize performance, if meets the end of the condition, return results of the current best particle, and terminates the algorithm, otherwise returns to step(3), and continue to the next cycle.

V. SIMULATION EXPERIMENTS AND RESULTS

In order to verify the effect of G-PSO algorithm proposed in the paper, several common functions have been optimized. Following is the course of the experiment and analysis of results.

Function 1:

$$f_1(x) = \sum_{i=1}^{10} 100(x_{i+1}^2 - x_i)^2 + (x_i - 1)^2, -30 \le x_i \le 30$$

The function f_1 is a single peak nonlinear function, which reaches the global minimum when $x_i=1, (i=1,2,...10)$ and $f_{min} = 0$. Though the function seems simple, but the global solution locate in the narrow band of $x_i + 1 = x_i^2$, in which function values change slowly and searching is easy to fall into local solution.

G-PSO and PSO are both used to optimize the function 1. The standard PSO's parameters: w = 1.2-0.1, means inertia weight varieties from 1.2 to 0.1 gradually with linear attenuation; Acceleration constant c1 = 0.15, c2 = 0.2; Population size m = 20; Parameters related to the function 1 are (*x1*, *x2*...*x10*), so D = 10. G-PSO parameters: w = 1.2-0.1; c1 = 0.15; c2 = 0.2; m = 20; D = 10; optimum area radius r = 0.03. Each algorithm compute 10 times, and the average results obtained are shown in Table 1.

 TABLE I.
 COMPARISON OF PSO AND G-PSO FOR FUNCTION 1

function	PSO		G-PSO	
	Function value	Average number of iterations	Min num	Average number of iterations
$f_I(x)$	0.0009876	42	0.0004017	30

Function 2:

$$r = \sqrt{(x - 50)^{2} + (y - 50)^{2}} + e,$$

$$f_{2} = \frac{\sin(r)}{r} + 1.$$

Function 2 is a multi-peak function optimization function, which reaches global maximum 1.1512 at (50, 50), whose second maximum value is 1.12837. Traditional optimization methods can hardly find the global optimum. Respectively, use standard PSO and G-PSO to optimize the operation of the function 2. The standard PSO parameters: w= 1.2-0.1; c1 = 0.2; c2 = 0.2; m = 20; D = 2. G-PSO algorithm parameters: w = 1.2-0.1; c1 = 0.2; c2 = 0.2; m = 20; D = 2. When optimum area radius r = 0.02, each algorithm runs 10 times and average the performance of each optimized algorithm, the last results are shown in table 2.

TABLE II. COMPARISON OF PSO AND G-PSO FOR FUNCTION 2

	PSO		G-PSO	
functio n	Functio n value	Averag e number of iterations	Min num	Averag e number of iterations
$f_2(x)$	1.142981	20	1.1506925	15

Through the analysis of the results of the optimization function 1 and function 2, we can conclude that : by using the gradient method, G-PSO optimization improves the properties in time and accuracy than the standard PSO. Individual particles can use gradient information to move directly to the optimal solution in close solutions of small areas, which can avoid individual particles jumping out the optimal solution of adjacent areas again due to the randomness of the standard PSO algorithm, as can speed up the pace to reach the optimal solution.

VI. CONCLUSIONS

This article mixes the gradient method with PSO algorithm, and gives G-PSO algorithm with gradient search factors. Under the guidance of gradient searching factors, PSO algorithm optimization process obvious more regularly, as can increase efficiency and optimizing accuracy. By optimizing the value of the single peak and multi-peak function, tests indicate that G-PSO is an effective method to decrease the computing time and improve precision

REFERENCES

- Kennedy J, Eberhart R. Particle swam optimization[C].IEEE Proceedings, IEEE Int Confon Neural Networks. Perth: IEEE, 1995: 1492-1948.
- [2] Ray T, Liew KM. A swam with effective information sharing mechanism for unconstrained and constrained single objective optimization problems[C]. IEEE Proceedings, IEEE Int Confon Evolutionary Computation Seoul; IEEE, 2001:75-80.
- [3] ShiYuhui, Eberhart R. Parameter selection in particle swam optimization[C]. Proc of 7th Annual Conf on Evolutionary Programming Washington DC, 1998:591-600.
- [4] EberhartR, ShiYuhui. Tracking and optimizing dynamic system with particle swams[C]. IEEE Proceeding. IEEE Int Confon Evolutionary Computation Hawaii. IEEE, 2001: 94-100.
- [5] Xu Showjiang, Zhu Qingbao. Particle swam optimization based on lp data fitting and its applications[J]. Journal of Nanjing Normal University. Engineering and Technology Edition. 2006, 6(3):62-65.