

# An Improved Particle Swarm Optimization Based on Repulsion Factor

Zhang Jie, Fan Chaozan, Liu Bo  
College of Science North China University of Technology,  
NCUT  
Beijing, China  
fanchaozan@126.com

Shi Fugui  
College of Mathematics Beijing Institute of Technology,  
BIT  
Beijing China

**Abstract**—In this paper, through the research of advantages and disadvantages of the particle swarm optimization algorithm, we get a new improved particle swarm optimization algorithm based on repulsion radius and repulsive factor. And a lot of test function experimental results show that the algorithm can effectively overcome the PSO algorithm precocious defect. PSO has significant improvement.

**Keywords**- PSO; RPSO; repulsion radius; repulsive factor

## 1. Introduction

Particle Swarm Optimization (PSO) was originally introduced by Kennedy and Eberhart. It is a population intelligence algorithm based on researching birds and fish populations behavior in 1995[1]. The thought is based on artificial life and evolutionary computation theory. Through imitating birds flying and foraging behavior, the population uses collective cooperation to achieve the optimal value.

PSO algorithm is a branch of evolutionary computation. It based on iterative searches to get the optimal value. Its advantages are fast convergence, the few parameters set and easy implement etc. The algorithm can effectively solve the complex optimization problem. Now it is used in a lot of field.

But PSO algorithm has flaws. Its local search ability is poor, the performance of the search is dependent on parameters, and especially it is easy to fall into the local optimum. Therefore, many researchers research the traditional PSO from the directions including parameters set, convergence, topology structure, and combine with other algorithm. They pointed out the deficiency then improved PSO in order to raise algorithm performance.

In order to overcome the premature drawback of PSO, we puts forward an improved algorithm based on the standard particle swarm optimization, that introducing repulsion radius and repulsive factor. And a lot of experimental data show that the algorithm can effectively overcome the PSO algorithm precocious defect. This algorithm has significant improvement to PSO.

## 2. Preparation

Traditional PSO imitates behavior of bird flocking and fish schooling in finding food. In the solving optimization problem, the potential solutions of problem correspond to positions of the birds in the search space (D dimension space). These birds called particles, each one has its own position

$x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$  and velocity  $v_i = (v_{i1}, v_{i2}, \dots, v_{iD})$  (decide flight direction and distance). And there is a fitness namely experience for every one decided by the optimization function (Fit). In the search process, one particle updates the position and velocity in flight according to its own experience and experience of nearby individuals (Pbest and Gbest). This process has implied parallelism.

Thereafter, the inertial weight  $w$  was first time introduced to the PSO update velocity formula by Shi. Y and Eberhart in [2], and they pointed out that the self-adapting inertia effect is one of the important parameters for arithmetic performance. The bigger  $w$  is more conducive to population to search into the broader space, and the smaller  $w$  ensures that the population eventually will converge to the optimal position. They put forward the improved evolution model with linear diminishing  $w$  [3]:

$$w = w_{\max} - (w_{\max} - w_{\min}) * G / \max G$$

In the formula,  $\max G$  is maximum iterating time,  $G$  is the current iteration time.

Standard particle swarm optimization formulas with the inertial weight  $w$  are as follows:

$$\begin{aligned} v_i(t+1) &= w * v_i(t) + \\ &c1 * rand * (P_{best} - x_i) + \\ &c2 * rand * (G_{best} - x_i) \end{aligned} \quad (2.1)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2.2)$$

In the formulas,  $c1, c2$  are acceleration coefficients;  $P_{best}$  is the current best position of the particle;  $G_{best}$  is the best particle position of the population;  $rand$  is the random number between  $[0, 1]$ ; and we give maximum velocity and the maximal displacement at the same time [3].

Shi. Y etc. discussed and researched other parameters including acceleration constant, maximum velocity limit, and population size etc. in [2] [4].

---

This work was supported by the National Natural Science Foundation of China(61074151) and (10971242), the Beijing Natural Science Foundation Program (1102016), Funding Project for Academic Human Resources Development in Institutions of Higher Learning Under the Jurisdiction of Beijing Municipality (PHR201108055) , the Beijing College Scientific Research and entrepreneurship Plan Project, the College of Mathematics Beijing Institute of Technology and the College Scientific and technological Activities of North China University of Technology.

Clerc M etc. first started to study the mathematical foundation and the convergence of the PSO algorithm. After analyzing PSO working mechanism and its convergence, the compression factor  $k$  was introduced to ensure convergence of the algorithm by them. And compression factor  $k$  could be regarded as a special example of inertia weight [7].

Krink T etc. put forward the spatial extension. If two particles will collide, they will go back the previous way. The improved algorithm used that to overcome premature convergence in iterative optimization [6].

Among the researchers of the combination of PSO and other algorithms, Ratnaweem introduced one mutation operator, that if test distance between particles and the current optimal is less than certain value than let the particle random initialization, into PSO to ensure the diversity of the population. [8].

Pant M etc. introduced a new diversity guided particle swarm optimizer in [11]. They put the population into three parts with the diversity of the population. The three parts used different evolution models. The model of middle part was similar to the way of Krink T.

In [12], the standard PSO formula (2.1) is divided into three parts, they are "inertia", "cognitive" and "social". And a kind of PSO improvement strategies was introduced. According to the Gaussian distribution, we can use the population statistics information to updates different types of particles with different evolution model. Its purpose is to accelerate the convergence.

### 3. Repulsion Particle Swarm Optimization

Before The PSO is proved to be convergent by Clerc M and other researchers. In experiments, we also can get that velocity of all particles gradually tend to be 0 in PSO. At the same time, all the particles are convergent around one point that generally the global optimal or local optimal (this is "premature"). When the particles converge to the optimal position, "cognitive" part basically is 0. At this time, the main influence of the particle movement is "social" factor, and "inertia" factor is mainly decided by the previous "social" factors. In order to avoid "premature" convergence and keeping particle population activity, we should mainly adjust the relationship between particles and the optimal position.

We consider that the repulsion factor between microscopic particles in physic can be a reference for avoiding "premature". That means when the distance between the particle and optimal position is smaller than a certain radius  $r$ , the Repulsive force outweigh the gravitational force, the particle will not tend to optimal particle, but to the opposite direction. This purpose is to keep population diversity.

This improvement uses the following principles to update the position and velocity:

(1) when the distance between particle and the optimal position is equal or more than repulsion radius  $r$ , the iteration process still uses the formula (2.1) (2.2) to adjust;

(2) otherwise, use the following formula adjustment:

$$\begin{aligned} v_i(t+1) &= w * v_i(t) + \\ &c1 * rand * (P_{best} - x_i) - \\ &c3 * rand * (G_{best} - x_i) \end{aligned} \quad (3.1)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (3.2)$$

The parameter is the same as (2.1) (2.2), the repulsive factor  $c3$ , should be bigger than  $c2$ .

But we find that the bigger radius and  $c3$  in the early period of the process are more conducive to jumping out premature points, and the smaller radius and  $c3$  in the later period will ensure that the population eventually are not far away from the optimal position. This is similar to the inertial weight  $w$ . So we think that the radius will be exponential decrease and  $c3$  linear decrease. And the lower limit of  $c3$  finally should be  $c2$  to insure the convergence of this algorithm.

The improved algorithm is repulsion particle swarm optimization (RPSO), algorithm procedure is as follows:

(1) Initialization, including population size  $N$ , the position and velocity of each particle, inertial weights, and accelerated factor, repulsion radius and repulsive factor;

(2) Calculate the fitness value of each particle,  $Fit$ ;

(3) Individual optimal position of the particle, for each particle, with its fitness value of  $Fit$  and previous individual optimal value  $P_{best}$  comparison, if  $Fit > P_{best}$ ,  $Fit$  replace  $P_{best}$ , else keep  $P_{best}$ ;

(4) Optimal position of population, each particle's fitness value,  $Fit$ , and previous global optimal value  $G_{best}$ , if  $Fit > G_{best}$ ,  $Fit$  replace  $G_{best}$ , else keep  $G_{best}$ ;

(5) Updated velocity and position of the particles; each particle, calculating the distance between the particles and the best, if the distance is equal or greater than rejection radius, update velocity and position of the particles according to the formula (2.1) (2.2); Otherwise, update of particles velocity and position according to the formula (3.1) (3.2)

(6) if meet the end conditions (to achieve maximum iterating times), exit, or back to (2).

### 4. Data Test

The experiments are used to compare standard particle swarm optimization and repulsion particle swarm optimization.

Use the following five standard test functions to test algorithm performance:

(1) Sphere function:

$$f_1(x) = \sum_1^n x_i^2$$

$$x = (x_1, x_2, \dots, x_n), -100 \leq x_i \leq 100$$

(2) Rastrigrin function

$$f_2(x) = \sum_1^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$$

$$x = (x_1, x_2, \dots, x_n), -5.12 \leq x_i \leq 5.12 ;$$

(3) Rosenbrock function

$$f_3(x) = \sum_1^{n-1} 100 (x_{i+1} - x_i)^2 + (1 - x_i)^2$$

$$x = (x_1, x_2, \dots, x_n), -30 \leq x_i \leq 30 ;$$

(4) Schaffer function

$$f_4(x) = 0.5 + \frac{\sin^2(\sqrt{x^2 + y^2}) - 0.5}{(1 + 0.0001(x^2 + y^2))^2}$$

x, y both real number ;

(5) Griewank function

$$f_5(x) = \frac{1}{4000} \sum_1^n x_i^2 - \prod_i \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

$$x = (x_1, x_2, \dots, x_n), -100 \leq x_i \leq 100 .$$

The best results are as follow:

$$f_1(0) = 0, f_2(0) = 0, f_3(1) = 0, f_4(0) = 0, f_5(0) = 0$$

Process parameters are: maximum of inertial weight w is 0.9, minimum 0.4; accelerated factor c1=c2=2; repulsive factor c3 takes 50 that decrease progressively to 2; square (for easy operation) of repulsion radius r takes 1e-4 that decrease progressively to 1 e-45. The population takes 30, maximum iterations is 1500, dimension is 10, each algorithm is 20 time. We get the all mean, best and worst results. The experimental results see Table 1 and Fig. 1 ~ Fig. 5.

TABLE I. TEST FUNCTION

Test Function	Standard PSO			RPSO		
	Best	Media	Worst	Best	Media	Worst
$f_1$	1.44E-42	6.97E-39	2.50E-38	1.91E-41	4.85E-39	4.16E-38
$f_2$	1.990	3.188	4.975	0	3.382	6.965
$f_3$	5.83E-15	5.942	17.71	1.82E-19	7.009	20.56
$f_4$	0	0.0029	0.0097	0	1.77E-05	8.12E-05
$f_5$	0	0.0014	0.0236	0	5.01E-18	1.11E-16

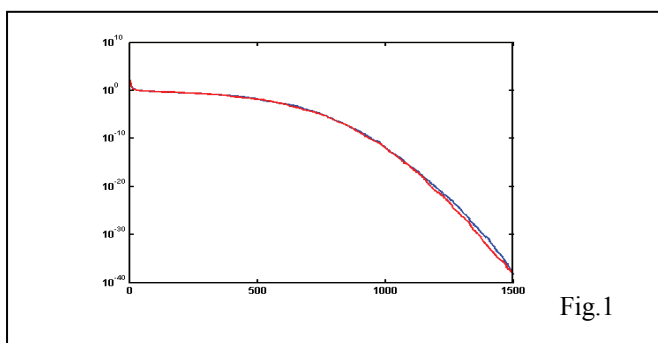


Fig.1

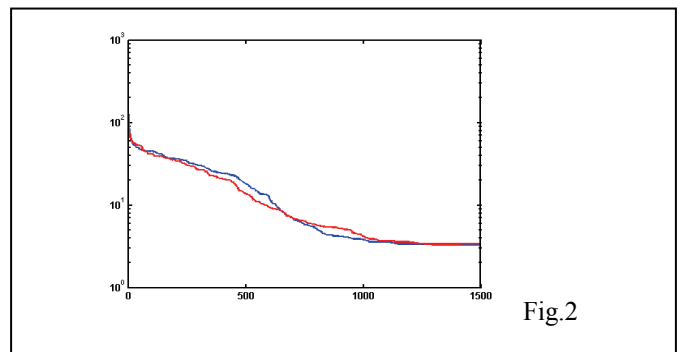


Fig.2

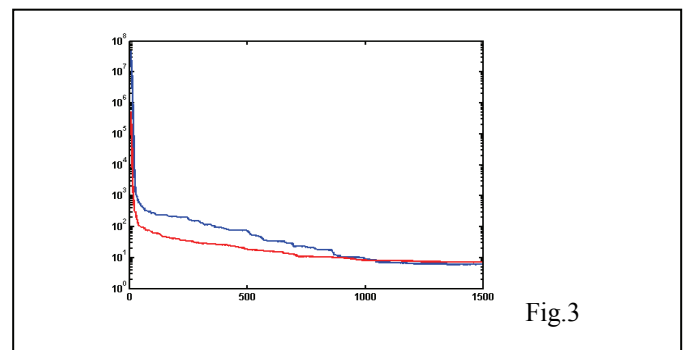


Fig.3

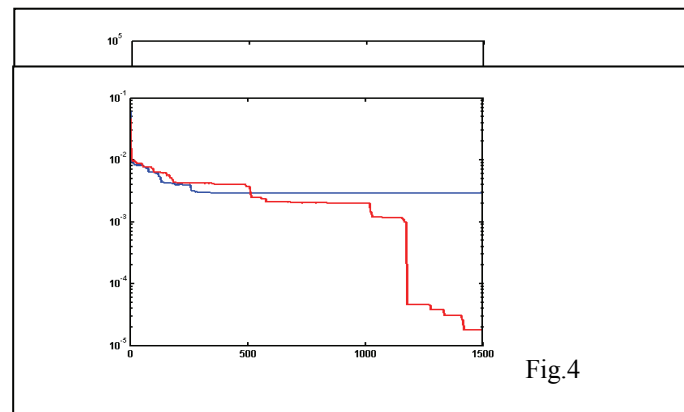


Fig.4

In Fig. 1~Fig. 5, blue line is PSO, red RPSO. All abscissa is 1-1500. All ordinate are index form.

From the table and curve images, we can find that RPSO has a very good performance relative to standard PSO in jumping out "premature" and it keeps the good performance of standard PSO even better. In function tests, the improvement effect especially is proved. Particles escape completely from the local optimal. But we can see that the improved algorithm has defects in the accuracy and velocity of convergence. The problem is mainly from repulsion radius and repulsive factor selection. In the experiments, different choices of radius and factor can produce unstable results. The best relation between

repulsion radius and repulsive factor is unknown. The modification research still needs to be further explored.”

## 5. Conclusion

Through the repulsion concept of microscopic particle in physics, this paper introduces the concept of repulsion radius and repulsive factor, and gives a new improved particle swarm optimization algorithm. The results of test functions have good performance, especially in Schaffer and Griewank function, both that local optimum is dense exist around the global optimal. The improve effect is obvious; this optimization has the value of further investigation.

## 6. Acknowledgment

This work was supported by the National Natural Science Foundation of China(61074151) and (10971242), the Beijing Natural Science Foundation Program (1102016), Funding Project for Academic Human Resources Development in Institutions of Higher Learning Under the Jurisdiction of Beijing Municipality (PHR201108055), the Beijing College Scientific Research and entrepreneurship Plan Project, the College of Mathematics Beijing Institute of Technology and the College Scientific and technological Activities of North China University of Technology.

## REFERENCES

- [1] Kennedy , Eberhart RC . Particle swarm optimization[C] . Proc of the IEEE International Conference on Neural Networks Piscataway, NJ : IEEE Service Center,1995 : 1942-1948.
- [2] Y. Shi and RC.Eberhart. Parameter selection in particle swarm optimization[C]. Evolutionary Programmin VII , V.W. Porto, N.Saravanan, D.Waagen, and A.E.Eiben, Eds. Berlin Germany: Spring-Verlag, 1997:591-600.
- [3] Shi Yuhui, Eberhart RC. A modified particle swarm optimizer[C] Proc of the IEEE International Conference on Evolutionary Computation. Piscataway, NJ : IEEE Service Center, 1998 : 69-73.
- [4] R. C. Eberhart and Y. Shi. Comparing Inertia Weights and Constriction Factors in Particle Swarm Optimization. IEEE,2000:84-88.
- [5] R. C. Eberhart and Y. Shi. Fuzzy Adaptive Particle Swarm Optimization. IEEE,2001:101-106.
- [6] Krink T., Vesterstrom J.S., Riget J. Particle swarm optimisation with spatial particle extension[C] Congress on Evolutionary Computation, 2002. CEC '02. 2002 : 1474 - 1479
- [7] Clerc M, Kennedy J. The particle swarm-explosions, stability and convergent in a multidimensional complex space [J] IEEE. Transaction Evolutionary Computation, 2002, 6(2) : 58-73.
- [8] Ranlaweera A, Halgamuge S K, Watson HC. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients[C]. IEEE Trans Evol Comput, 2004 : 240-255.
- [9] Sedlaczek K, Eberhard P. Using augmented Lagrangian particle swarm optimization for constrained problems in engineering.[J]Struct Multidiscip Optim2006,32(4)277-286.
- [10] Zhu Xiaoliu, Xiong Weili, Xu Baoguo. QDPSO Algorithm Based on Simulated Annealing Technique [C]. Computer Engineering, Vol.33 No.15, 2007.8:209-210.
- [11] Pant M., Radha T., Singh V.P. A Simple Diversity Guided Particle Swarm Optimization. Congress on Evolutionary Computation, 2007. CEC 2007. IEEE. 2007 : 3294 – 3299.
- [12] Bi Xiaojun, Liu Guoan. An improved particle swarm optimization algorithm based on population classification[J]. Journal of Harbin Engineering University, 2008,29 (9) : 991-996.
- [13] Zhang J, Shi Y, zhan ZH. Power electronic circuits design : A particle swarm optimization approach[C]. SEAL, 2008 : 605-614.
- [14] Mingquan Chen. Second Generation Particle Swarm Optimization. Congress on Evolutionary Computation,IEEE,2008:90-96.
- [15] Jakob V. and Rene Thomsen. A Comparative Study of Differential Evolution, Particle Swarm Optimization, and Evolutionary Algorithms on Numerical Benchmark Problems[C]. IEEE,2004,1980-1987.
- [16] Eberhard P, Sedlaczek K. Using augmented Lagrangian particle swarm optimization for constrained problems in engineering[C]. Advanced Design of Mechanical Systems: From Analysis to Optimization. 2009:253–271.
- [17] Sun C, Zeng J, Pan J. An improved particle swarm optimization with feasibility- based rules for constrained optimization problems[C]. Next-Generation Applied Intelligence 2009:202–211.
- [18] Lingfeng Wang and Chanan Singh. Multicriteria Design of Hybrid Power Generation Systems Based on a Modified Particle Swarm Optimization Algorithm. Transactions on Energy Conversion, IEEE, 2009.3:163-172.
- [19] Venter G, Haftka R. Constrained particle swarm optimization using a bi-objective formulation. Struct Multidiscip Optim 2010;40:65-76