

# Particle Swarm Optimization With Adaptive Parameters and Boundary Constraints

Hui Niu, Yongshou Dai, Xing Peng

*Department of Electrical Engineering College of Information and Control Engineering China University of Petroleum (East)*

---

## Abstract

The core idea of PSO is that each particle searches the best solution of optimization problems according to “information sharing” between surrounding particles and itself. PSO has fast convergence speed and high global search capability. For low accuracy and divergent results of elementary PSO, this paper proposes a kind of PSO with adaptive parameters and boundary constraints. Inertia weight and learning factors increase or decrease linearly with iterative process, in order that the particles search the global space in early period of the algorithm and converge towards the global optimum later. At the same time, the author sets particle boundary constraints to ensure the optimization accuracy. Theoretical analysis and numerical simulation results show the efficiency and high optimization accuracy of the designed method.

**Index Terms:** PSO adaptive parameters boundary constraints

© 2012 Published by MECS Publisher. Selection and/or peer review under responsibility of the Research Association of Modern Education and Computer Science.

---

## 1. Introduction

PSO is the new nonlinear optimization method, proposed by Dr. Kennedy and Dr. Ebberhart of US in 1995[1].It originates from study of birds and fishes food searching behavior. The basic idea is that each particle searches the optimal solution through group collaboration and information sharing between particles. It has simple concepts, less parameters and fast convergence speed. But the accuracy of the algorithm has close relationships with parameter selections. Inappropriate parameters easily lead to divergent results. In optimization process of elementary PSO, particles may be beyond the range of effective solutions because there is no boundary constraint.

\* Corresponding author.

E-mail address: [nhui\\_1987@163.com](mailto:nhui_1987@163.com)

## 2. Particle Swarm Optimization

### 2.1. Mathematical Description

In the T-dimensional target searching space, there is a population of particles which represent N potential solutions of specific optimization problems.  $s = \{x_1, x_2, \dots, x_n\}$ ,  $x_i = \{x_{i1}, x_{i2}, \dots, x_{it}\}$ ,  $i = 1, 2, \dots, n$  means a vector point of the ith particle in the T dimensional solution space.  $p_i = \{p_{i1}, p_{i2}, \dots, p_{it}\}$  means the optimal solution searched by the ith particle;  $p_g = \{p_{g1}, p_{g2}, \dots, p_{gt}\}$ ,  $g = 1, 2, \dots, n$  means the optimal solution searched by all the particles;  $v_i = \{v_{i1}, v_{i2}, \dots, v_{it}\}$  means the searching speed of the ith particle. Every particle adjusts itself to find the optimal solution by tracking  $p_i$  and  $p_g$ . When the two extremes [2] are found, the particle updates its velocity and position according to (1) and (2) [3].

$$v_{ij}(t+1) = \omega v_{ij}(t) + c_1 r_{1j} (p_{ij}(t) - x_{ij}(t)) + c_2 r_{2j} (p_{gj}(t) - x_{ij}(t)) \quad (1)$$

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

$\omega$  : inertia weight;  $c_1, c_2$ : learning factors;  $r_1, r_2$ : random variables which are uniformly distributed in the range of  $[0, 1]$ ;  $t$ : the number of iterations.

From view of physics,  $\omega v_{ij}(t)$  calls as the memory item, for better ability to expand the searching space;  $c_1 r_{1j} (p_{ij}(t) - x_{ij}(t))$  calls as the particle cognitive item ,for the optimal solution from its own experience;  $c_2 r_{2j} (p_{gj}(t) - x_{ij}(t))$  calls as the group cognitive item ,reflecting collaboration and information sharing between particles[4], [5].

### 2.2. Steps of Particle swarm optimization

Steps of particle swarm optimization are as follows [6]:

Step 1: Initialize the particles. Set the initial position  $x_i$  and initial velocity  $v_i$  of every particle randomly;

Step 2: Calculate the fitness value of each particle. Determine the individual optimal solution  $p_i$  and its corresponding particle, the global optimal solution  $p_g$  and its corresponding particle;

Step 3: For each particle, compare its fitness value and that of its own optimal solution  $p_i$ . Update  $p_i$  if the fitness value of the particle is better than that of  $p_i$ ;

Step 4: For each particle, compare its fitness value and that of the global optimal solution  $p_g$ . Update  $p_g$  if the fitness value of the particle is better than that of  $p_g$ ;

Step 5: Adjust  $x_i$  and  $v_i$  according to (1) and (2) ;

Step 6: End the iteration process if the particle reaches the conditions; or transfer to step 2;

### 2.3. Defects of Elementary Particle swarm optimization

The defects of elementary particle swarm optimization algorithm are as follows:

(1) Accuracy of the algorithm has close relationships with parameter selections. Value of inertia weight is between 0.2 to 1.5, values of  $c_1, c_2$  are both 2 in normal conditions.

(2) The parameters of the algorithm are constants regardless of the specific optimization models and their iterative processes.

Test and optimize the four standard benchmark functions, which are generally used to test and compare the performances of algorithms [7].

(1) Sphere function:

$$f(x) = \sum_{i=1}^{30} x_i^2 \quad (3)$$

$$-100 \leq x_i \leq 100, \min(f(x)) = f(0, 0, \dots, 0) = 0$$

(2) Generalized Griewank function:

$$f(x) = \frac{1}{4000} \sum_{i=1}^{30} x_i^2 - \prod_{i=1}^{30} \left( \frac{x_i}{\sqrt{i}} \right) + 1 \quad (4)$$

$$-600 \leq x_i \leq 600, \min(f(x)) = f(0, 0, \dots, 0) = 0$$

(3) Generalized Rosenbrock function:

$$f(x) = \sum_{i=1}^{30} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2] \quad (5)$$

$$-30 \leq x_i \leq 30, \min(f(x)) = f(0, 0, \dots, 0) = 0$$

(4) Generalized Rastrigin function:

$$f(x) = \sum_{i=1}^{30} [x_i^2 - 10 \cos(2\pi x_i) + 10] \quad (6)$$

$$-5.12 \leq x_i \leq 5.12, \min(f(x)) = f(0, 0, \dots, 0) = 0$$

Sphere function is a nonlinear symmetric single-peak function; Griewank function is a rotated, variable-dimensional function which cannot be separated; Rosenbrock function is the typical pathological and quadratic function; Rastrigin function is a typical complex multi-peak function with large number of local optimas.

Conclusion: As it is shown in Fig.1 to Fig 5, different values of  $c_1, c_2$  correspond different optimization results. The algorithms performances closely relate to parameter selections. Appropriate parameters determine higher accuracy of the algorithm, whereas accuracy will be reduced.

### 3. Improved PSO

#### 3.1. The core idea of improved PSO

For accuracy reduced of the algorithm because of improper parameter selections[8], this paper analyzes mathematical equation of elementary PSO. Inertia weight and learning factors increase or decrease linearly with the iterative process, so that the particles can search the entire space without falling into local optimum in the early period and find the global optimum in the end. Meanwhile, the improved PSO introduces differential mutation and random mutation, to increase the diversity of the particles group. At the end, this kind of PSO sets particle boundary constraints to ensure the effectiveness of solutions.

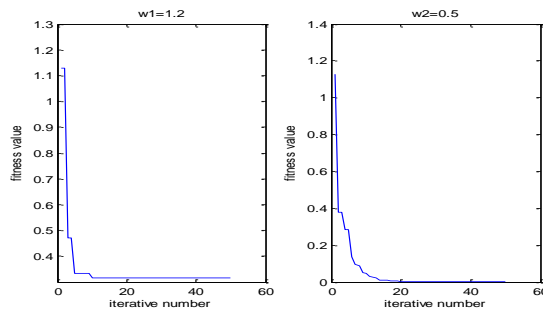


Fig. 1. (a) Sphere function: different  $\omega$ , same  $C_1, C_2$

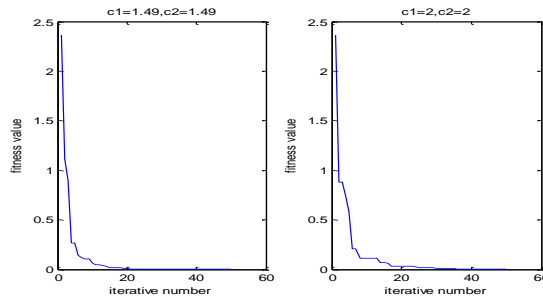


Fig. 2. Sphere function: same  $\omega$ , different  $C_1, C_2$

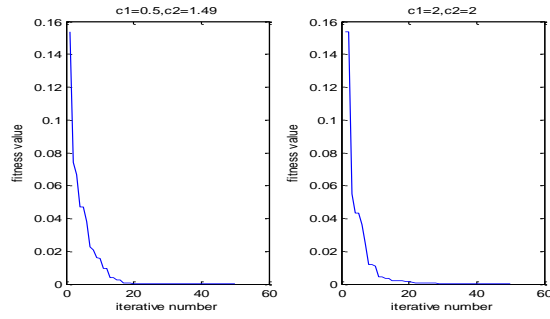


Fig. 3. Griewank function: same  $\mathcal{O}$ , different  $C_1, C_2$

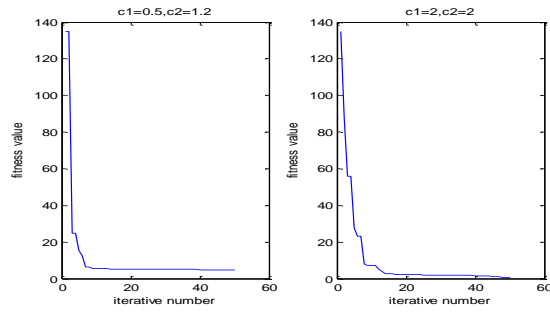


Fig. 4. Rosenbrock function: same  $\mathcal{O}$ , different  $C_1, C_2$

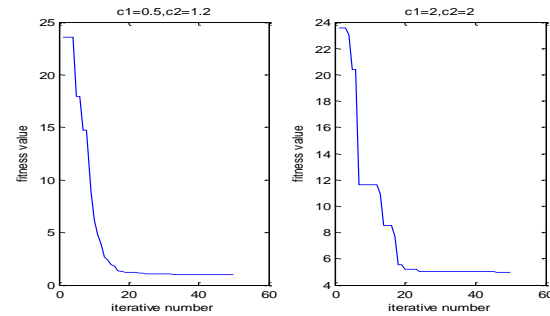


Fig. 5. Rastrigin function: same  $\mathcal{O}$ , different  $C_1, C_2$

The improvements of PSO are as follows:

(1) Inertia weight

From the mathematical expression of PSO, we can conclude that larger inertia ensures a more effective global search of particles, smaller inertia weight means a more efficient local search. According to this theory, Shi and Eberhart proposed a strategy which inertia weight linearly decreases with the increase number of iterations [8].

Expressed as follows :

$$\omega = \omega_{\max} - k * (\omega_{\max} - \omega_{\min}) / gen \quad (7)$$

$\omega_{\max}$  calls as the maximum inertia weight,  $\omega_{\min}$  calls the minimum inertia,  $gen$  calls the total number of iterations for the algorithm,  $k$  calls the current number of iterations for the algorithm.

(2) learning factors

$c_1, c_2$  respectively represents the abilities of local extremes and global extremes search. Larger  $c_1$  and smaller  $c_2$  make the particles search extremes in the entire feasible space at early of the algorithm; Smaller  $c_1$  and larger  $c_2$  ensure the particles converge quickly to the global optimum value later.

Expressed as follows:

$$c_1 = c_{1start} - k * (c_{1start} - c_{1end}) / gen \quad (8)$$

$$c_2 = c_{2start} + k * (c_{2end} - c_{2start}) / gen \quad (9)$$

Where  $c_{1start}, c_{2start}$  call as the initial values for the learning factors,  $c_{1end}, c_{2end}$  call as the final values for the learning factors.

(3)Introduction of differential and random mutation

Introduce a differential variation when the particles fly and set the optimal position [7]. It results in a new particle swarm, aimed to increase the diversity of population of particles.

Expressed as follows:

$$xv(i, j) = x(g_1, j) + r * (x(g_2, j) - x(g_3, j)) \quad (10)$$

Where  $r$  is a random number in the range of  $[0,1]$ ,  $g_1, g_2, g_3$  are the random values between  $[1, gen]$ .  $x$  is the position of particle before mutation and  $xv$  calls as the position after mutation.  $1 \leq j \leq T$  and  $i$  is greater than zero and less than  $n$  (number of particles). Compare the fitness value of  $xv$  with that of  $x$ , if better, then  $xv$  will replace  $x$ . Or the particle population keeps the same. The mutation enables particles to search optimal results, aimed to maintain algorithm accuracy and diversity of populations.

Then introduce a random mutation after differential mutation. It happens on each particle following the mutation probability  $p = 1 / T$  ( $T$  calls as the dimension of variables). This step ensures the particles to get out of the local optimal extremes and keep the diversity of populations.

(4) Set the boundary constraints of particles.

When the position of the particle is beyond the given position, redefine the particle's position using following formulas, to make sure that the particle is in the range of feasible solutions. At the same time, this redefinition ensures the diversity of new particles for undetermined boundary constraints.

$$\text{If } x_{ij} < l, x_{ij} = u - \text{mod}(l - x_{ij}, u - l) \quad (11)$$

$$\text{If } x_{ij} > u, x_{ij} = l + \text{mod}(x_{ij} - u, u - l) \quad (12)$$

Where  $u$  is the upper bound and  $l$  is the lower bound for the particle position, mod is the remainder operation.

### 3.2. Steps of improved PSO

Improve the elementary PSO and get the improved PSO flowchart (Fig.6). The implementation steps are as follows:

Step 5: Update the position and velocity of the particle according to (1) and (2).  $c_1, c_2$  change linearly with the increase of iteration, which are no longer constants. Expressed as (8) and (9);

Step 6: Introduce the differential mutation, expressed as (10) after determining the particle's best optimal position. Compare the particle mutated and those of before mutation. If better, replace  $x$  to  $xv$ . Or keep still;

Step 7: Introduction of the random mutation. This occurs on each particle with the probability  $p = 1/T$ .

Step 8: If the particle's position is beyond the given boundary, redefine the position of the particle based on (11) and (12);

Step 9: End the iteration when the PSO reaches its conditions (maximum number of loops or the requirement of accuracy of algorithm). Or transfer to step 2;

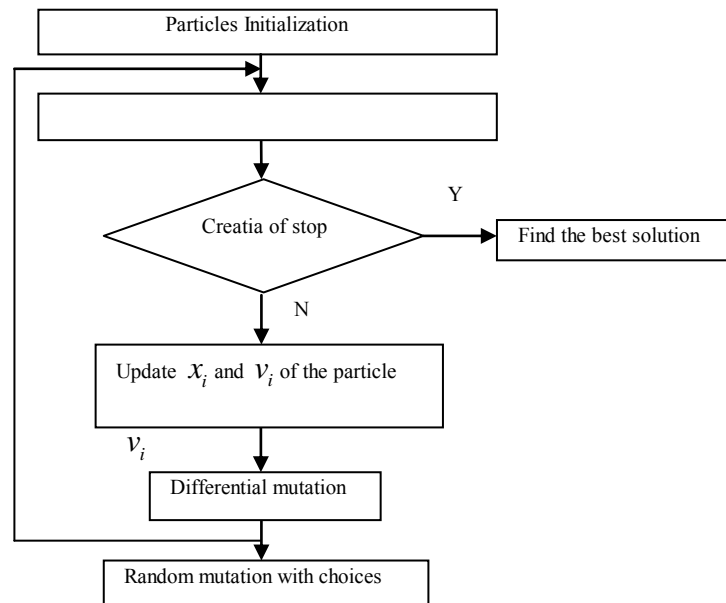


Fig. 6. the flowchart of Improved PSO

### 3.3. The results of Improved PSO

The improved PSO algorithm is used in standard test functions (Sphere, Griewank, Rosenbrock, Rastrigin). The optimization results are shown in Tab. 1 and Fig. 7 to Fig 10.

Table 1. Optimization results of Improved PSO

Test functions	Algorithm	Iterative numbers	Optimal results	Optimal solution
Sphere	Elementary PSO	18	0.8052	0.2003
	Improved PSO	75	$5.5412 \times 10^{-10}$	$1 \times 10^{-6}$
Griewank	Elementary PSO	13	0.0931	0.1005
	Improved PSO	76	$4.6103 \times 10^{-11}$	$6 \times 10^{-6}$
Rosenbrock	Elementary PSO	25	1.1005	0.4201
	Improved PSO	45	0.5001	0.0793
Rastrigin	Elementary PSO	30	0.7643	0.1965
	Improved PSO	67	$7.0703 \times 10^{-5}$	0.0002

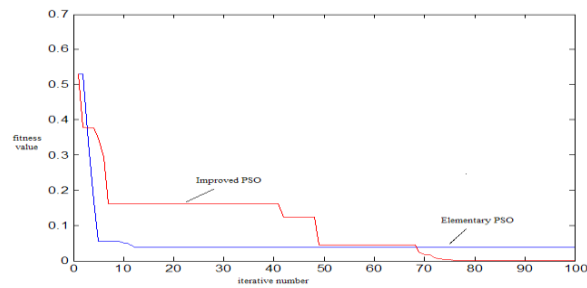


Fig. 7. The iterative results of Sphere function based on different PSO

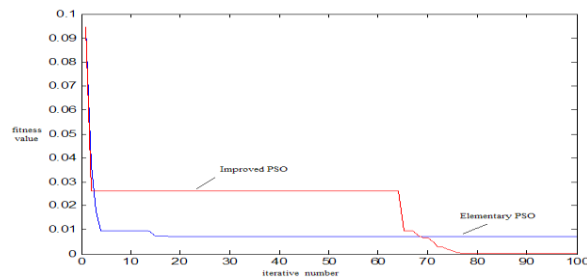


Fig. 8. The iterative results of Griewank function based on different PSO



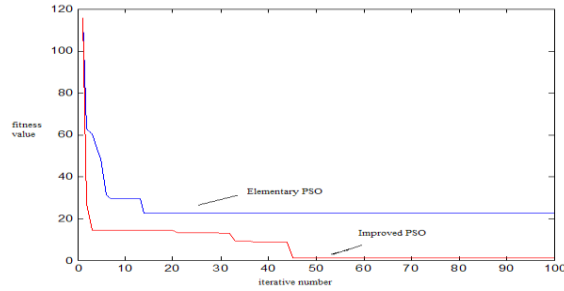


Fig. 9. The iterative results of Rosenbrock function based on different PSO

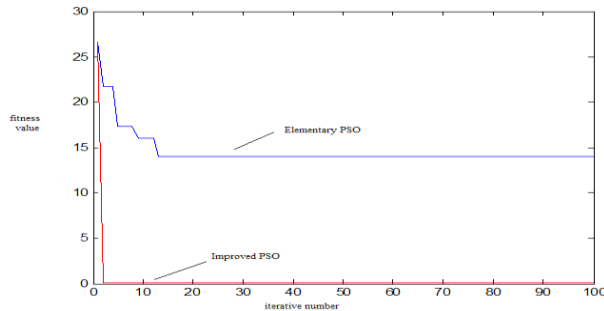


Fig. 10. The iterative results of Rastrigin function based on different PSO

### 3.4. Evaluation of algorithm performances

The performances of algorithms are evaluated using the following standards: (1) Evaluation of convergence rates (2) Evaluation of stability and quality of convergence.

#### (1) Evaluation of convergence rate

The convergence rates of algorithms can be reflected by test of average evolution generations. The average iterative number of elementary PSO is from 18 to 30 because of the advantage of “memory”. However, improved PSO owns more iterative numbers for the increase of calculation.

#### (2) Evaluation of stability and quality of convergence.

The indicators of evaluation are the average value which converges to optimal value and the average evolution generation. It is shown that the elementary PSO has poor

local searching ability, low convergence rate and max error between the global optimal values. It can be seen that improved PSO improves the accuracy of the algorithm and stability of the global optimal value in Fig 7 to Fig 10.

## 4. Conclusions

In this paper, an improved PSO algorithm is proposed for the deficiencies of elementary PSO.

(1)The improved PSO has its superiority. The results of four standard test functions show more accurate performances of improved PSO.

(2)PSO is a new evolutionary algorithm based on swarm intelligence [7]. But unlike the in-depth research of genetic and simulated annealing algorithms, there are many issues to solve. For example, there must be strict theoretical proof about convergence and global optimality of PSO. Another focus will be how to combine PSO and other evolutionary algorithms, and establish the appropriate PSO model for different optimization problems.

## **References**

- [1]KENNEDY J,EBERHART C. Particle swarm optimization. Proceeding of IEEE International Conference on Neural Networks. Piscataway ,NJ:IEEE,1995:1942-1948
- [2]KENNEDY J,EBERHART R C.A new optimizer using particle swarm theory[A].Proceedings of the Sixth International Symposium on Micro Machine and Human Science [C].Nagoya,Japan:IEEE,1995:3943.
- [3]Yahya Rahmat-Samii,Genetic algorithm(GA) and particle swarm optimization(PSO):evolutionary optimization paradigms in modern electronic engineering.
- [4]KENNEDY, EBERHART R C.A new optimizer using particle swarm theory[A].Proceedings of the Sixth International Symposium on Micro Computation.Piscataway,NJ:IEEE Press,1998:303-308.
- [5]Clerc M, Kennedy J. The particle swarm-Explosion, stability,and convergence in a multidimensional complex space. IEEE Trans.on Evolutionary Computation,2002,6(1):58-73
- [6]Cristian TI.The particle swarm optimization algorithm:Convergence analysis and parameter selection.Information Processing Letters,2003,85(6):317-325.
- [7]Yuhui Shi and Russell C. Eberhart. Parameter selection in particle swarm optimization. Evolutionary Programming, New York,1998:591-600.
- [8]SHI Y,EBERHART R C.A modified particle swarm optimizer.IEEE International Conference on Evolutionary Computation.Piscataway,NJ:IEEE Press,1998:303-308.