

# An Improved Particle Swarm Optimization with Gaussian Disturbance

Changjun WEN<sup>1</sup>, Changlian LIU<sup>1</sup>, Heng ZHANG<sup>1</sup> and Hongliang WANG<sup>1</sup>

<sup>1</sup>School of Mechanical Engineering, Hubei University of Technology, China

**Abstract.** The particle swarm optimization (PSO) is a widely used tool for solving optimization problems in the field of engineering technology. However, PSO is likely to fall into local optimum, which has the disadvantages of slow convergence speed and low convergence precision. In view of the above shortcomings, a particle swarm optimization with Gaussian disturbance is proposed. With introducing the Gaussian disturbance in the self-cognition part and social cognition part of the algorithm, this method can improve the convergence speed and precision of the algorithm, which can also improve the ability of the algorithm to escape the local optimal solution. The algorithm is simulated by Griewank function after the several evolutionary modes of GDPSO algorithm are analyzed. The experimental results show that the convergence speed and the optimization precision of the GDPSO is better than that of PSO.

## 1 Introduction

Particle swarm optimization (PSO) is a typical representative of Swarm Intelligence[1], which was proposed by Kennedy and Eberhart[2] in 1995 based on detailed research on the foraging behaviour of birds. The core idea of PSO is promote population development by using the information sharing mechanism and learn from each other's experience. PSO has been widely used in pattern recognition[3], electric power system[4], engineering technology, *etc.* Although PSO is easy to implement for its less parameters and fast convergence speed, however, its unreasonable theoretical contributed to poor convergence precision and low population diversity.

Aiming at the deficiency of PSO, an improved particle swarm optimization with Gaussian disturbance (GDPSO) is proposed in this paper. The structure of this paper is as follows: the first part is the introduction, which briefly describes the generation of PSO algorithm and the related features of the algorithm. The second part describes the related theory of the standard PSO. In the third part, the GSPSO algorithm is proposed, and several evolutionary models of the algorithm are analysed. The fourth part is the simulation experiment. The GDPSO algorithm is simulated by the Griewank function. The fifth part is the conclusion. The experimental data is analysed and summarized.

## 2 The standard PSO algorithm

In the particle swarm optimization, the potential solution of each optimization problem is a "bird" in the search

space, which is called "particle". All particles have the adaptive value determined by an optimized function, and each particle has a velocity that determines the flying direction and distance. Then the particles follow the current optimal particle and search in the solution space.

In each iteration, the particle is updated by tracking two extreme values: one is the optimal solution found by the particle itself, which is called the individual extreme value; the other is the optimal solution found by the whole population, and the extreme value is the global extreme value. By changing the velocity and motion direction of particles and learning from the historical optimal position and global optimal position, the PSO algorithm will keep updating the position of particles until the requirements are met.

The particle velocity and position update formulas are as follows:

$$v_i^d(t+1) = \omega v_i^d(t) + c_1 r_1 [p_i^d(t) - x_i^d(t)] + c_2 r_2 [p_g^d(t) - x_i^d(t)] \quad (1)$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \quad (2)$$

In the Eq.1 and Eq.2,  $v_i^d(t+1)$  and  $x_i^d(t+1)$  are symbolize the velocity and position of the d-dimensional subspace of particle  $i$  in the  $t+1$  iteration.  $\omega$  is the inertia weight, and its main function is to balance the local and global searching ability of particles;  $c_1$  and  $c_2$  are learning factors. The main function is to have the capability to self-learning and learn from other excellent particles in the group so that the particles can quickly move to their own historical optimal position and the global optimal position, generally taking 2.0.  $r_1$  and  $r_2$  are two random numbers that obey the uniform distribution (0,1);  $p_i^d(t)$  is the best position of the

\*Corresponding author: 794411947@qq.com

particle  $i$  in the  $t$  iteration;  $pd\ g(t)$  is the historical optimal position of the whole particle group when the  $t$  iteration is iterated.

### 3 The PSO algorithm with Gaussian disturbance

#### 3.1 The GDPSO algorithm

In the PSO algorithm, the personal optimal position and the global optimal position act as attractor in the search process. Particles in a population will oscillate between their personal optimal position and the global optimal position which has been found by all particles. The PSO algorithm has been proved to be convergent in certain conditions[5]. In the later period of the algorithm, all particles will fly in the direction of local best or global best. At this point, the optimal position of each particle, the global optimal position of the population and the current position of each particle will approach to one point, and the particle's velocity will approach to zero. If all the particles are around the local optimal solution at this time, the algorithm will fall into the local optimum and appear the premature phenomenon. Considering the slow convergence speed and the poor convergence precision of the PSO, the Gaussian disturbance term is introduced to enhance the algorithm's ability of escaping the local optimal and improve the convergence accuracy of the algorithm.

The speed update formula is as follows:

$$v_i^d(t+1) = \omega v_i^d(t) + c_1 r_1 [p_i^d(t) + gauss_1 - x_i^d(t)] + c_2 r_2 [p_g^d(t) + gauss_2 - x_i^d(t)] \quad (3)$$

$$gauss_1 = r_3 * N(\mu, \sigma^2) \quad (4)$$

$$gauss_2 = r_4 * N(\mu, \sigma^2) \quad (5)$$

In the Eq.4 and Eq.5,  $r_3$  and  $r_4$  are two random numbers that obey the uniform distribution.  $N(\mu, \sigma^2)$  is a Gaussian distribution which the mathematical expectation is  $\mu$  and the variance is  $\sigma^2$ .

#### 3.2 The evolution model of GDPSO algorithm

To simplify the analysis, it is assumed that  $c_1=c_2$ ,  $r_1=r_2$  in the Eq.3. The position of the particles in the population can be divided into the following cases:

A. The current position of particles is not the historical optimal position, nor is it the global optimal position. Moreover, the historical optimal position of particles is not the global optimal position in the current population, namely  $xd\ i \neq pd\ i$ ,  $xd\ i \neq pd\ g$  and  $pd\ i \neq pd\ g$ . Then the evolutionary pattern of the algorithm is:

$$v_i^d(t+1) = \omega v_i^d(t) + c_1 r_1 [p_i^d(t) - x_i^d(t)] + c_2 r_2 [p_g^d(t) - x_i^d(t)] + c_1 r_1 (gauss_1 + gauss_2) \quad (6)$$

Compared with the standard PSO algorithm, the Gaussian disturbance term is mainly added in the

formula. By increasing the speed increment, the particle's searching range can be expanded. It can also enable the particles to find better solutions and improve the convergence speed of the algorithm.

B. The current position of the particle is not its historical optimal position, nor is its the global optimal position, but the historical optimal position of the particle is the global optimal position of the current population, namely  $xd\ i \neq pd\ i$ ,  $xd\ i \neq pd\ g$  and  $pd\ i = pd\ g$ . Then the evolutionary pattern of the algorithm is:

$$v_i^d(t+1) = \omega v_i^d(t) + 2c_2 r_2 (p_g^d(t) - x_i^d(t)) + c_1 r_1 (gauss_1 + gauss_2) \quad (7)$$

In this situation, the convergence speed of the PSO algorithm is very fast and the particles in the population completely move toward the current global optimal position. However, if the current global optimal position is only one of the local extremes rather than the optimal position, the algorithm will fall into the local optimal position and it is difficult to escape. At this time, the GDPSO algorithm is equivalent to adding two social models, which can avoid the algorithm being in the local optimal position.

C. The particle's current position is its historical optimal position, but it is not the global optimal position, namely  $xd\ i = pd\ i$  and  $xd\ i \neq pd\ g$ . Then the evolutionary pattern of the algorithm is:

$$v_i^d(t+1) = \omega v_i^d(t) + c_2 r_2 (p_g^d(t) - x_i^d(t)) + c_1 r_1 (gauss_1 + gauss_2) \quad (8)$$

In this moment, the particle velocity increment is larger than the standard PSO algorithm, which can help the particles escape the local optimum and makes the algorithm have the stronger global searching ability and the faster convergence speed.

D. The current position of the particle is the global optimal position of its population, but it is not its historical optimal position, namely  $xd\ i = pd\ i$  and  $xd\ i \neq pd\ g$ . Then the evolutionary pattern of the algorithm is:

$$v_i^d(t+1) = \omega v_i^d(t) + c_1 r_1 (p_i^d(t) - x_i^d(t)) + c_1 r_1 (gauss_1 + gauss_2) \quad (9)$$

At this point, GDPSO algorithm is equivalent to adding a Gaussian disturbance term on "self-cognition". In this way, the particle's velocity increment is larger than the standard PSO, which helps the particle escape from the local optimum.

E. The current position of the particle is both its historical optimal position and its global optimal position, namely  $xd\ i = pd\ i$ ,  $xd\ i = pd\ g$ . Then the evolutionary pattern of the algorithm is:

$$v_i^d(t+1) = \omega v_i^d(t) + c_1 r_1 (gauss_1 + gauss_2) \quad (10)$$

In this case, the historical optimal position of the particle and the global optimal position have no effect on the current particles in the standard PSO algorithm. The particle  $i$  only depends on the original velocity inertial motion, so that the particle can easily fall into the local optimum. The Gauss disturbance term makes the particle velocity faster than the standard PSO algorithm, which

helps the particles jump out of the current local optimal solution.

## 4 Simulation experiment

### 4.1 Griewank function

The Griewank function can be used to test the global search capability of the algorithm and the ability to jump out of local traps due to its continuous and multi-peak features. The expression is as follow [6]:

$$F(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) \quad (11)$$

### 4.2 Steps of solving Griewank function by GDPSO

In order to test the performance of the algorithm, the specific steps are as follows:

- (1) Initializing the position and velocity of the particle;
- (2) Evaluating and calculating the particle's fitness value;
- (3) Calculating the historical optimal position of each particle and the optimal position of the particle swarm;
- (4) Updating the position and speed of the particle based on the formula (2) and formula (3);
- (5) If the termination condition is reached, the algorithm will stop searching. Otherwise, the GDPSO will jump to step (2) and continue the calculation.

The flow chart of GDPSO algorithm is as follow: (see Fig.1)

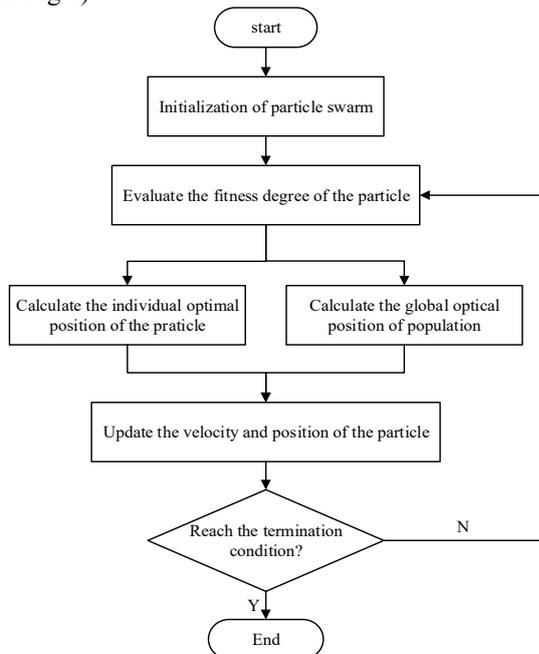


Figure 1. Flowchart of GDPSO

## 4.3 The simulation results

The simulation results are as follows:(see Fig.2-Fig5)

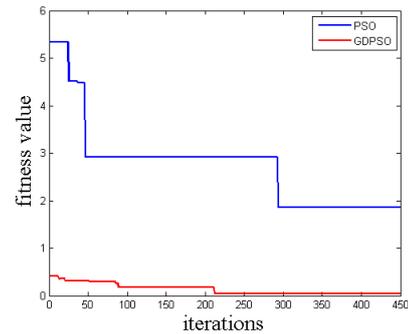


Figure 2. Result of PSO and GDPSO comparison

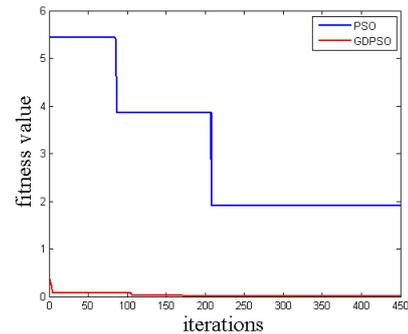


Figure 3. Result of PSO and GDPSO comparison

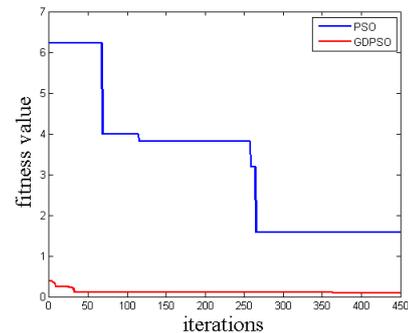


Figure 4. Result of PSO and GDPSO comparison

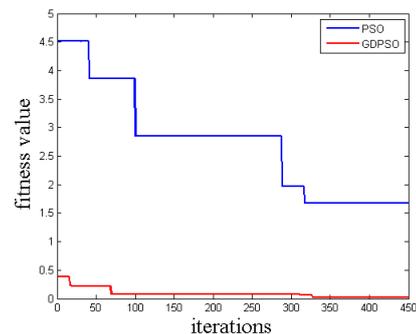


Figure 5. Result of PSO and GDPSO comparison

The comparison of results in Fig.2-Fig.5 are shown in the table below.

	Fig.2		Fig.3		Fig.4		Fig.5	
	GDPSO	PSO	GDPSO	PSO	GDPSO	PSO	GDPSO	PSO
Best fitness value	0.0247	1.7315	0.0146	1.2732	0.0387	1.4311	0.0113	1.6619
First iteration value	11	24	6	84	10	67	15	40

## 5 Conclusion

Based on the standard PSO algorithm, this paper proposed an improved particle swarm optimization algorithm with Gaussian disturbance. Compared with the standard PSO algorithm, the GDPSO algorithm improves its shortcomings, which is slow in convergence and easy to fall into local optimum. The GDPSO algorithm improves the calculation result by one or two decimal points than the standard PSO, and the initial iteration number is at least ten times earlier. The Gaussian disturbance term introduced in this paper is a standard normal distribution. How the Gaussian disturbance with different expectation and variance values affects on the algorithm will be another subject area worth studying.

## References

1. Bonabeau E, Dorigo M, Theraulaz G. Swarm intelligence: from natural to artificial systems[J]. Santa Fe Institute Studies on the Sciences of Complexity, 1999.
2. Kennedy J, Eberhart R. Particle swarm optimization[J]. Proc. of 1995 IEEE Int. Conf. Neural Networks, (Perth, Australia), Nov. 27-Dec. 2011, 4(8):1942-1948 vol.4.
3. Omran MGH, Engelbrecht AP, Salman A. Particle Swarm Optimization for Pattern Recognition and Image Processing[J]. Studies in Computational Intelligence, 2006, 34:125-151.
4. Chakrabarti S, Venayagamoorthy GK, Kyriakides E. PMU placement for power system observability using binary particle swarm optimization[C]// Power Engineering Conference, 2008. AUPEC '08. Australasian Universities. IEEE, 2008:1-5.
5. Clerc M, Kennedy J. The particle swarm - explosion, stability, and convergence in a multidimensional complex space[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(1):58-73.
6. Griewank AO. Generalized descent for global optimization[J]. Journal of Optimization Theory & Applications, 1981, 34(1):11-39.