IOP Conf. Series: Materials Science and Engineering **435** (2018) 012028 doi:10.1088/1757-899X/435/1/012028

Multiple Improvements to the Particle Swarm Optimization Algorithm

T Li [1,2] and Y Chen[1]

[1]School of Transportation Management, Dalian Maritime University, Dalian 116026, P.R.China
[2]China Waterborne Transport Research Institute, Beijing 100088, P.R.China

litao@wti.ac.cn

Abstract. To address the tendency of the particle swarm optimization algorithm to fall into local minima as well as to speed up its final stages of convergence, this paper proposes a combination of several improvements to the algorithm. First, the convergence factor is introduced as a special case of the inertia weight and transboundary particles are reset. Second, to increase the diversity of the particles and break the stagnation states of the particle swarm, a random number is introduced into the speed and position of the particles. The variation of the particles' position and velocity is hence optimized, and the search space for the optimal value is expanded. We use the single-mode sphere and Rosenbrock functions as well as the multimodal Rastrigrin and Griewank functions to verify the algorithm. The maximum, mean, variance, stability, and convergence accuracy of the proposed algorithm are improved.

1. Introduction

The particle swarm optimization (PSO) algorithm is a type of bionic intelligence optimization algorithm proposed by Kennedy and Eberhart [1]. It is based on the idea of iterative search for an optimal value and solves the optimization problem by simulating a group of foraging activities and individual foraging action laws. In the foraging behavior of birds, although the target is not clear, each bird knows how far it is away from a target, so the best foraging program is to fly to the food closest to the location of the bird. This is why flying birds will suddenly synchronize their direction of change or aggregation or, alternatively, scatter. In the PSO algorithm, each bird is abstracted as a random search particle, and the distance to the target is measured by the fitness function. Each particle corresponding to a solution of the optimization problem is solved in the d-dimensional search space, and there is a speed vector and position vector that determine the search direction and distance in the search space of each particle. There is also a fitness value based on the objective function. Each particle knows the best position and current position it has experienced so far. This represents the flight experience accumulated by the particle itself. Moreover, each particle also knows the best position experienced by all the particles in the current population, which is called the accumulated flight experience of the population. Using these two pieces of information, the particles continually update their own speed and position. Simultaneously, the global best position and fitness of the local particle and global population are constantly changing until the condition for terminating the iterations is met. The optimal solution is the final global optimal position.

This method records the particle's position vector $X_i = (x_{i1,} x_{i2,} \cdots, x_{id})$ and speed vector $V_i = (v_{i1,} v_{i2,} \cdots, v_{id})$. The individual extrema of the particles $P_i = (p_{i1,} p_{i2,} \cdots, p_{id})$ and the global extrema of the population $P_g = (p_{g1,} p_{g2,} \cdots, p_{gd})$ are also recorded, where $I = 1, 2, \ldots, m$. A population composed of m particles searches in d-dimensional space. For the (t + 1)th iteration, the particles update the particle position and velocity using the following equation.

$$\begin{aligned} v_{ik}^{t+1} &= v_{ik}^{t} + \mathrm{c}_1 r_1 (p_{ik}^{t} - x_{ik}^{t}) + \mathrm{c}_2 r_2 (p_{gk}^{t} - x_{ik}^{t}) \\ x_{ik}^{t+1} &= x_{ik}^{t} + v_{ik}^{t+1} \end{aligned} \tag{1}$$

Here, c1 and c2 are the learning factors, where generally c1 = c2 = 2. In addition, r1 and r2 are used to increase the particle flight randomness. In Equation (1), the first term is the speed of the last iteration of the particle, followed by the term that specifies the effect of the best position of the particle on the next iteration, which comes from the flight experience of the particle itself. Finally, the last term is the social experience, which indicates the effect of the global extreme point on the current particle position, that is, the experience of all the particles benefits the search process of the particle. In each dimension, the velocity and position of the particles are limited to avoid searching beyond the scope of the search, for example, if the particle speed is too high, it could fly out of the range of the optimal solution, and if the speed is too low, it could perform an invalid or partially optimal search.

Kennedy and Eberhart proposed PSO in 1995. PSO uses a set of particles that represent potential solutions to an optimization problem. The particle moves toward an optimal solution based on its present velocity and the individual best position. It has found in previous iterations while also incorporating the globally best solution found by its companion particles [2]. To gain deeper insight into the mechanism of PSO, many theoretical analyses have been conducted on the algorithm, and most of these studies have focused on the behavior of a single particle in PSO, analyzing the particle's trajectory or stability using deterministic or stochastic methods [3–8]. To view PSO from a new perspective, we constructed a relationship between the dynamic PSO process and the transition process of a control system to identify factors that influence the convergence speed and stability of the basic PSO without increasing the algorithm's complexity [3]. In recent years, PSO has captured a great deal of attention and has been successfully applied in many applications because of its simplicity of implementation [9-12]. However, many experiments have indicated that the canonical PSO suffers from premature convergence when solving complex problems [13-14]. For this reason, a number of methods have been proposed to improve the efficiency and/or effectiveness of the PSO algorithm through adjustments to the code pattern, inertia weight, and maximum speed of the particles, mutation operators, neighborhood operations, and boundary conditions [15-18]. Some of the studies used hybrid (mixture and parallel) algorithms, for instance PSO with immune algorithms, simulated annealing algorithms, and chaos algorithms, to improve search effectiveness [19-21].

Overall, PSO algorithm research includes the following main aspects. For theoretical research, PSO convergence theory mainly focuses on a simplified PSO system. The convergence of the PSO can be analyzed using dynamic system stability theory to obtain some convergence conditions [22–25] by dynamically expressing the behavior of the particles as a nonlinear feedback control system. Furthermore, it obtains a statistical law of the PSO when approaching an attraction center [26]. To solve dynamic optimization problems, a PSO with a dynamic and hierarchical domain structure has been used, and it was proven that this structure can help keep the particles dispersed in a dynamic environment [24].

Using the above techniques, when an initial random solution is used such that the iterations of the particle position bring it close to the optimal solution, which is easy to implement, the algorithm converges faster. Furthermore, there are no complicated crossover and mutation operations as in a genetic algorithm [27]. However, it is also easy for a PSO to fall into a local extremum [28] and the final

phase of the convergence is slow, among other problems [29]. To address these shortcomings, this paper presents three improvements to the PSO algorithm as described in detail in the next section.

2. Improvement Strategies for PSO

2.1 Convergence factor

Shi [30] proposed the inertia weight method to guarantee the convergence of the PSO algorithm and changed the speed update formula. This equation is now regarded as the standard PSO model, and is calculated as follows.

$$v_{ik}^{t+1} = wv_{ik}^{t} + \mathrm{c}_1 r_1(p_{ik}^{t} - x_{ik}^{t}) + \mathrm{c}_2 r_2(p_{gk}^{t} - x_{ik}^{t}) \quad (2)$$

In Equation (2), *w* is the inertia weight, and its value reflects the effect of the historical state of the particle on the current state during particle iteration. The algorithm is optimized using fixed or linearly decreasing inertia weights. Usually, weights in the range [0.1, 0.9] obtain better convergence performance. When the value is greater than 1.2, the particle group will not converge [31].

To improve the convergence speed and performance of the PSO algorithm, Clerc [32] added a convergence factor, as follows.

$$v_{ik}^{t+1} = \chi[v_{ik}^{t} + \mathrm{c}_1 r_1(p_{ik}^{t} - x_{ik}^{t}) + \mathrm{c}_2 r_2(p_{gk}^{t} - x_{ik}^{t})] \quad (3)$$

Here, $\chi = \dfrac{2}{\left|2-\varphi-\sqrt{\varphi^2-4\phi}\right|}$ is the convergence factor, where $\varphi = c_1 + c_2 > 4$, φ is usually set to 4.1, giving a convergence factor of 0.7298. The convergence factor can be regarded as a special case of the inertia weight. In this paper, the convergence factor is used to optimize PSO.

2.2 Handling transboundary particles.

During particle parameter optimization, if a particle's parameters exceed the range of allowable values, they are usually reset. The usual approach is to set the velocity and position of the transboundary particle parameters to the boundary values. That is, when the particle speed exceeds the upper bound, i.e., $v_{ik} > v_{k,\max}$, it is set to the maximum speed $v_{ik} = v_{k,\max}$, and when the velocity of the particle is below the lower bound, i.e., $v_{ik} < v_{k,\min}$, it is set to the minimum speed, $v_{ik} = v_{k,\min}$. The position of a particle beyond the border of the solution space is treated similarly. This method was suggested by Fu [33]. If there is a locally optimal solution in the vicinity of the boundary value, it will cause a large number of particles to move closer to the boundary value to form a particle aggregation, which may cause the particle group to fall into a locally optimal extremum and result in a combination of parameters that are not optimal.

Hence, this paper proposes an improved way to set the speed and location of transboundary particles. When $v_{ik} > v_{k,\max}$, the velocity of the particle is combined with a random number, $v_{ik} = v_{k,\max} - (v_{k,\max} - v_{k,\min}) * \mathrm{rand}()$. When the velocity of the particle is below the minimum boundary value, $v_{ik} < v_{k,\min}$, it is changed using $v_{ik} = v_{k,\min} + (v_{k,\max} - v_{k,\min}) * \mathrm{rand}()$. Using the same improvement strategy, the position of transboundary particles is also treated with random values; that is, when the particle position exceeds the highest boundary value, i.e., $x_{ik} > x_{k,\max}$, the location of the particles is corrected as $x_{ik} = x_{k,\max} - (x_{k,\max} - x_{k,\min}) * \mathrm{rand}()$. Likewise, when $x_{ik} < x_{k,\min}$, the position of the particle is corrected using $x_{ik} = x_{k,\min} + (x_{k,\max} - x_{k,\min}) * \mathrm{rand}()$. The rand() function generates a random value in [0, 1].

2.3 Adaptive particle mutation.

In the PSO algorithm, the position of the particle is determined by both the optimal merits of the particle itself and the optimal merits of the particles in the whole population. Hence, if there is a particle in the population that is close to the optimal solution, the other particles in the population will move rapidly in the direction of this particle. If there is no better position than the current global extremum, the algorithm is more likely to fall into a search for a locally optimal solution, which may lead to the convergence of the algorithm. Most of the particles are clustered into one place, so there is no way for particles to search the space again. Hence, because the final stage of the convergence is too slow, the lack of particle diversity caused by premature convergence, and other shortcomings, the diversity of the particles should be increased to break the stagnation of particle groups. It is possible to consider increasing the position or velocity of the particles so that the particle swarm can continue to find the optimal value, which strengthens the convergence speed and accuracy of the particle group. Because the key to avoiding locally optimal solutions is to use the optimal position experienced by each individual particle, this paper introduces an adaptive mutation based on the best position experienced by the particle itself.

Consider the example of looking for a minimum value. After the *k*-th search is performed, using $f_{avg}(k)=\frac{1}{n}\sum_{i=1}^{n} f_i(k)$, we can obtain the average and variance of the particle fitness of the particle group:

$$\sigma^2=\sum_{i=1}^{n}\left(\frac{f_i-f_{avg}}{f}\right)^2$$

$$f=\begin{cases} max\{|f_i-f_{avg}|\}, max\{|f_i-f_{avg}|\}>1 \\ 1, others \end{cases} \tag{4}$$

Here, *f* is a normalization factor that controls the population fitness variance of the data range. When the variance of the overall fitness of the particle group is small, the particle group will be more closely aggregated. If the algorithm begins to prematurely converge, to reduce the possibility of local convergence of the current particles, this paper proposes randomly selecting a dimension for the position mutation operation according to a certain probability. This probability is the probability of occurrence of particle variation, which is related to the variance of the population fitness.

$$P=P_{\max}-(P_{\max}-P_{\min})\frac{\sigma^2}{N} \tag{5}$$

$$x_{ik}=x_{k,min}+(x_{k,max}-x_{k,min})*rand \tag{6}$$

In each iteration, a random number r is generated, and if it is smaller than the probability P of the mutation operation, the position of the particle can be updated, and the dimension of the particle position is arbitrarily selected.

When the variance of the overall fitness of the particle group is large, the diversity of the current particle swarm is relatively high and it is still in a state of random search. Here, it is not necessary to increase the diversity of the particle swarm. The probability of mutation should also be relatively small. When the variance of the population fitness is small, the current particle group is relatively dense, and the states of the particles need to be more diverse. Therefore, according to Equation (5), P is large and indicating that mutation of the particles is more likely. Hence, this method of calculating the mutation probability increases the diversity of particles when the algorithm begins to converge.

2.4 Pseudo Code of the Improved PSO Algorithm.

In this paper, the PSO algorithm is improved by introducing the convergence factor, combining it with the random reset for the velocity and position of transboundary particles, and adding particle position mutation to increase the diversity of the particle swarm. The pseudo code of the algorithm after these improvements is as follows.

```
Begin
        %Initial population
```

```
        For i = 1 to N
          For d = 1 to Dimension D
              Randomly initialize the population and speed
                End
      Calculate the initial fitness value fitness (i)
        End
        % Find individual extremes and global extremes
        [global_fit bestindex] = min(fitness)
        Indiv_fit = fitness
        % Iterative optimization
        For i = 1 to MaxIter
                For j = 1 to N
                        For d = 1 to Dimension D
                         Vj = w*Vj + c1*rand*(indiv_xj − xj) + c2*rand*(global_x − xj)  % Update Speed
                        if Vj > Vmax then
      Vj = Vmax − (Vmax − Vmin)*rand    % Cross speed processing
          end if
        if Vj < Vmin then
      Vj  = Vmin + (Vmax − Vmin)*rand
    end if
                Xj = Xj + Vj   % Update location
                if        Xj > Xmax then
      Xj = Xmax − (Xmax − Xmin)*rand  %Transboundary position processing
        end if
        if        Xj < Xmin then
        Xj = Xmin + (Xmax − Xmin)*rand
    end if
                End
                if rand < p
                        k = ceil(D*rand)
                        Xj, k = Xk, min + (Xk, max − Xk, min)*rand  % Particle position mutation operation
                end if
                update fitness(j), global_fit, indiv_fit(j)
                End
                For k = 1 to N
                        f(k) = abs(fitness(k) − avgfit_gen(i)) % Fitness variance
                End
                For k = 1 to N
                        sigma(k) = (fitness(k) − avgfit (i))/fmax;
                End
                square = sum(sigma^2)
                p = pmax − (pmax − pmin)*square/N          % Mutation probability
        End
End
```

3. Experimental analysis

3.1 Standard test functions.

To verify whether the improved PSO algorithm proposed in this paper improves its performance, four benchmark functions widely used to evaluate the performance of the PSO algorithm are used. The four benchmark functions are divided into two categories: single-mode functions and multimodal functions. That is, they are divided by their number of extrema. Each category contains two test functions.

A. Single-mode functions

The sphere function is a simple single-peak continuous function, usually used to detect the optimization accuracy of an algorithm. Its value is (0, ..., 0) at the globally optimal solution, and its three-dimensional map shown in Figure 1. The function is expressed as formula(7).

$$f_1 = \sum_{i=1}^{n} x_i^2 \qquad (7)$$

$$f_2 = \sum_{i=1}^{n} 100(x_{i+1}^2 - x_i) + (1 - x_i)^2 \qquad (8)$$

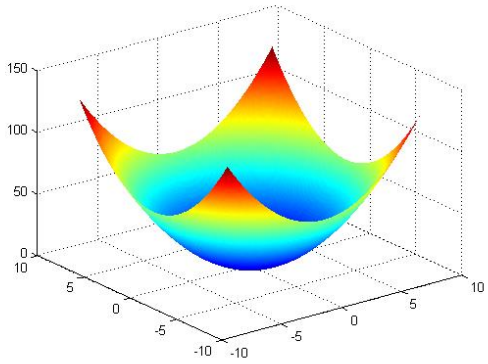

Figure 1 Sphere function

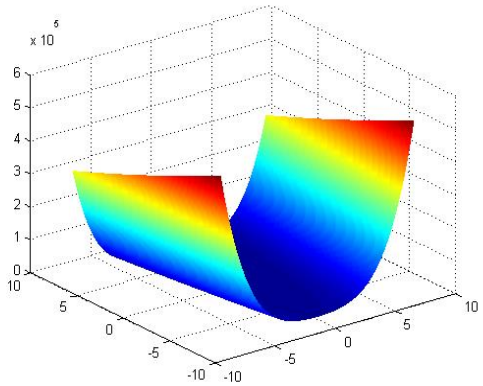

Figure 2 Rosenbrock function

The Rosenbrock function is a nonconvex function with a globally optimal solution at (1, ..., 1). Its three-dimensional graph is a parabolic valley, as shown in Figure 2, and the function is expressed as formula(8) .

B. Multimodal functions

The Rastrigrin function is a multi-peak function of sinusoidal ripple. There is a globally optimal solution at (0, ..., 0). To see the position of its global optimal solution, we present the three-dimensional graph of the negative of the function in Figure 3. There are several local extrema in the range (−5.12, 5.12), the number of which is related to the dimension, and the function expression is as formula(9), where the two coefficients are 10 here, but can be other values.

$$f_3 = \sum_{i=1}^{n} (x_{i+1}^2 - 10\cos(2\pi x_i) + 10) \qquad (9)$$

$$f_4 = \frac{1}{4000}\sum_{i=1}^{n} x_{i+1}^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1 \qquad (10)$$

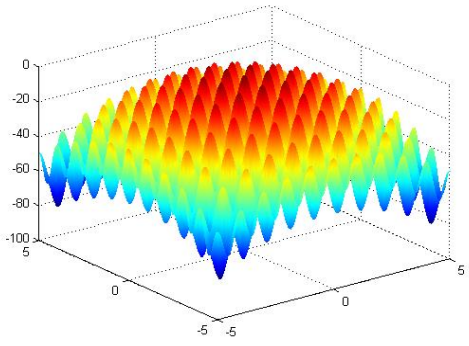

Figure 3 Rastrigrin function

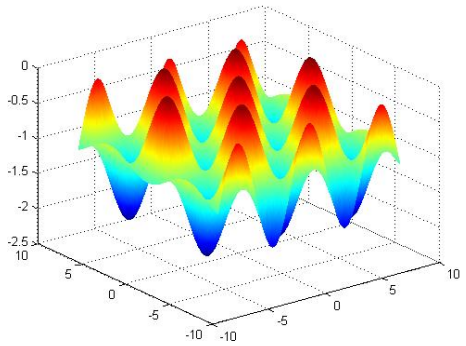

Figure 4 Griewank function

The Griewank function is a typical nonlinear multi-peak function, and it has many local extrema. Their number is related to the dimension of the search space, and the globally optimal solution is obtained at (0, ..., 0). We show the three-dimensional graph of the negative function in Figure 4. The function is expressed as formula(10).

The search ranges of the four test functions are shown in Table 1, and the search space dimension is 10 for all functions.

Table 1 Search range of the benchmark functions

functions	range	optimum	optimum points
Sphere	(-100, 100)	0	(0, …, 0)
Rosenbrock	(-30, 30)	0	(1, …, 1)
Rastrigrin	(-5.12, 5.12)	0	(0, …, 0)
Griewank	(-300, 300)	0	(0, …, 0)

3.2 Results

The performance of the improved PSO algorithm is compared with those of the standard PSO, LDWPSO [34] and CLSPSO [35] algorithms on the four benchmark functions. Table 2 shows the parameters of each algorithm in this experiment.

Table 2 Parameter settings for each algorithm

algorithms	w	c_1	c_2	Population size
PSO	0.9	2	2	20
LDWPSO	[0.5, 0.9]	2	2	20
CLSPSO	0.9	1.496	1.496	20
Proposed	0.729	1.496	1.496	20

The four algorithms were run 10 times on each benchmark function and the maximum number of iterations was 1,000. The minimum, maximum, mean, and variance of each PSO algorithm are shown in Table 3. Figures 5–Figures 8 shows the convergence of the four algorithms in different benchmark functions.

Table 3 Comparison of each PSO algorithm on the benchmark functions

functions	algorithms	min	max	average	variance
Sphere	PSO	3.57316E-05	0.995077471	0.17262312	0.096997182
	LinWPSO	1.13012E-07	2.622217715	0.294078248	0.675193305
	CLSPSO	0	183313.1499	18331.31499	3360371093
	Proposed algorithm	0.001006009	0.004771275	0.002622363	1.29E-06
Rosenbrock	PSO	8.612494913	336.0685119	81.45418547	10114.34847
	LinWPSO	9.444753583	1273.833536	201.2536662	152751.5962
	CLSPSO	0	1.46577E+11	14657663272	2.14847E+21
	Proposed algorithm	8.743861127	81.93926906	21.21815365	6.61E+02
Rastrigrin	PSO	7.825758006	32.72522317	16.58103419	49.48017738
	LinWPSO	11.94200845	29.85006006	18.41880806	34.80968821
	CLSPSO	0	113737.0288	11373.70288	1293611172
	Proposed algorithm	5.97915844	20.90221235	10.75755236	28.78808246
Griewank	PSO	0.054095474	0.175401891	0.110346599	0.00241495
	LinWPSO	0.118265615	1.552401333	0.781081812	0.259327117
	CLSPSO	0	42.40817951	4.240817951	179.8453689
	Proposed algorithm	0.029118976	0.363126298	0.187540239	0.008228349

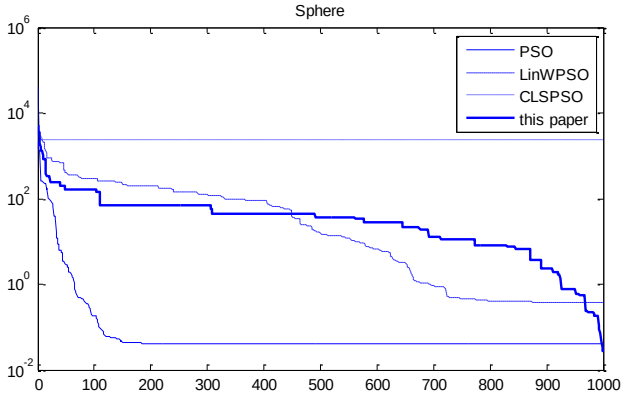

Figure 5 Sphere function results

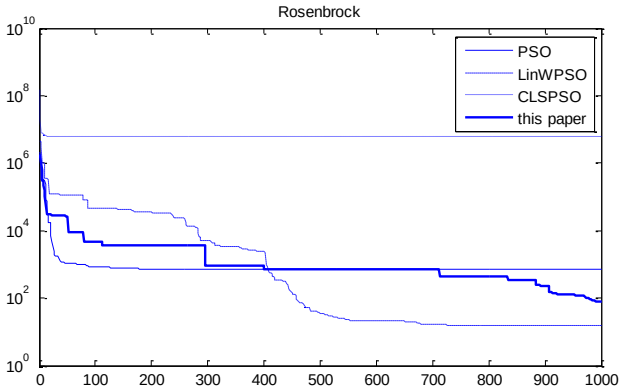

Figure 6 Rosenbrock function results

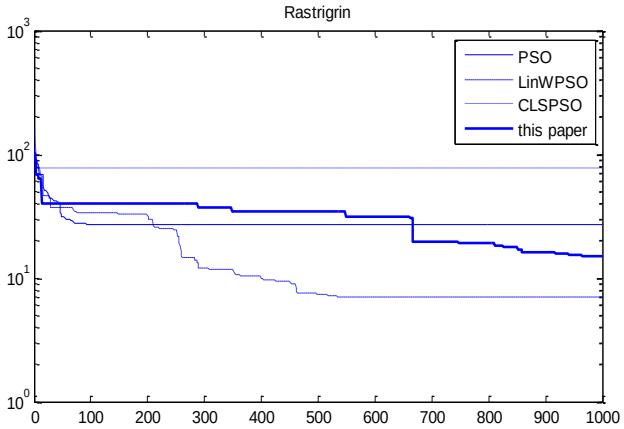

Figure 7 Rastrigrin function results

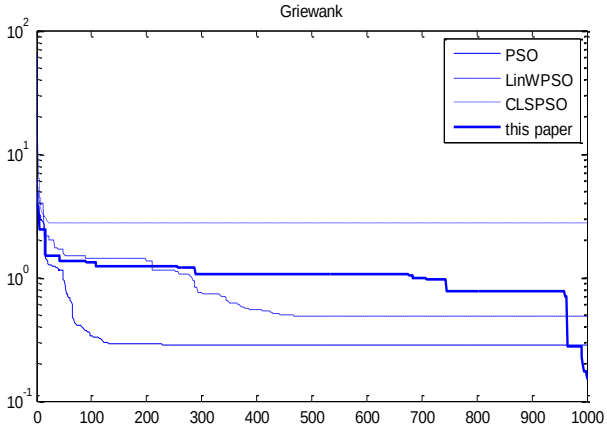

Figure 8 Griewank function results

The results show that the improved PSO algorithm has better local search ability than the other PSO algorithms. Later increases in the diversity of particles to boost the algorithm out of a local search obtains a better convergence. For the sphere function, the proposed algorithm has better stability than the other algorithms with respect to mean and variance, and the convergence precision of the algorithm is better than those of the other algorithms. The complicated Rosenbrock function optimization problem provides more detailed information. Each algorithm has large differences in the single-experiment results, but the proposed algorithm still has better stability and convergence accuracy. For the multimode Rastrigrin and Rosenbrock functions, several obvious inflection points can be seen in Figures-8, which is the embodiment of the particle self-mutation. A comparison of the experimental data obtained on the four test functions shows that the algorithm proposed in this paper performed better with respect to the maximum, mean, and variance than the other three algorithms.

4. Conclusions

In PSO, the individual particles make full use of their own experience and the group's experience to adjust their own state to obtain the optimal solution. However, it has the problem of premature convergence (especially when dealing with complicated, multi-extrema search problems) and poor global optimization ability. In this paper, we increased the convergence factor and used a random number with the velocity and position to reset transboundary particles. To avoid becoming trapped in a locally optimal solution, that is, the optimal position of an individual, the particle's location was modified by adaptive mutation to increase the diversity of the particles in a group. The improved algorithm was compared with the standard PSO, LDWPSO, and CLSPSO algorithms, and the maximum, minimum, mean, and variance are improved on standard single-mode and multimodal test functions. The performance results prove the effectiveness of the improved PSO algorithm.

Acknowledgments
This work was supported by the National Science and Technology Major Project of China (Grant No. 2017YFC1404602).

References

[1] Kennedy J, Eberhart R C. (1995)Particle swarm optimization[C]. *In Proc. of IEEE conference on neural networks*. 1942-1948.
[2] Zhang W, Liu J, Fan L B, Liu Y H, Ma D. (2016) Control strategy PSO[J]. *Applied Soft Computing*, 38(C):75-86.
[3] Clerc M., Kennedy J., (2002) The particle swarm-explosion, stability, and convergence in a multidimensional complex space [J], *IEEE Trans. Evolut. Comput*. 6(1): 58-73.
[4] Kadirkamanathan V., Selvarajah K., Fleming P.J. (2006) Stability analysis of the particle dynamics in particle swarm optimizer [J], *IEEE Trans. Evolut. Comput*. 10(3):245-255.
[5] Eberhart R.C., Shi Y. (2000) Comparing intertia weights and constriction factors in particle swarm optimization [C]. *in: Proceedings of the Congress on Evolutionary Computation*, 1(5):84-88.
[6] Trelea I.C. (2003) The particle swarm optimization algorithm: convergence analysis and parameter selection [J], *Inf. Process. Lett*. 85(6): 317-325.
[7] Jiang M., Luo Y., Yang S. (2007) Particle Swarm Optimization-Stochastic Trajectory Analysis and Parameter Selection., *I-TECH Education and Publishing*.
[8] Jiang M., Luo Y.P., Yang S.Y. (2007) Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm [J], *Inf. Process.Lett.* 102 (1): 8-16.
[9] Ciuprina G, Ioan D, Munteanu I. (2002) Use of intelligent-particle swarm optimization in electromagnetics [J], *IEEE Trans. Magn*. 38(2): 1037-1040.
[10] Ling S.H, Iu H.H.C, Chan K.Y. (2008) Hybrid particle swarm optimization with wavelet mutation and its industrial applications [J], *IEEE Trans. Syst. Man Cybern. Part B: Cybern*., 38(3): 743-763 .
[11] Soh H., Y.S., Q.C. Nguyen, Nguyen Q.H., (2010) Discovering unique, low-energy pure water isomers: meme Ong tic exploration, optimization and landscape analysis [J], *IEEE Trans. Evol. Comput*., 14(3): 419-437.
[12] Wachowiak M.P., Smolikova R, Zheng Y.F, (2004) An approach to multimodal biomedical image registration utilizing particle swarm optimization [J], *IEEE Trans. Evol. Comput.* 8(3): 280-288.
[13] Zhan Z.H., Zhang J., L i Y., (2009) Adaptive particle swarm optimization [J], *IEEE Trans. Syst. Man Cybern. Part B: Cybern*. 39(6): 1362-1381.
[14] Xia X W, Xie C W, Wei B, et al. (2017) Particle swarm optimization using multi-level adaptation and purposeful detection operators [J], *Information Sciences*, 385-386:174–195.
[15] Feng Y.M., Li C.J., Zhang M., (2008) Research on the function of reservoir long term operation based on improved particle swarm optimization (IPSO)[J], *Water Power*. 34(2): 94–97.
[16] Li G.Y. (2006) Regulation of water and sediment for the Yellow River based on joint operation of reservoirs and artificial intervention [J], *Journal of Hydraulic Engineering*, 37(12):1439-1446.
[17] Marler R.T., Arora J.S. (2004) Survey of multi-objective optimization methods for engineering [J], *Structural & Multidisciplinary Optimization*, 26(6):369-395.
[18] Massimiliano K., (2013) A multi-start opposition-based particle swarm optimization algorithm with adaptive velocity for bound constrained global optimization [J], *Journal of Global Optimization*, 55(1):165-188.
[19] Wang Y., Zhou J.Z., Zhou C., et al, (2012) An improved self-adaptive PSO technique for short-term hydrothermal scheduling [J], *Expert Systems with Applications*, 39(3):2288-2295.
[20] Yin X.A., Yang Z.F., Yang W., et al. (2010) Optimized reservoir operation to balance human and riverine ecosystem needs: model development, and a case study for the Tanghe reservoir, Tang

river basin, China [J], *Hydrological Processes*, 24(4):461-471.
[21] Yu X.B., Cao J., Shan H.Y., et al, (2014) An adaptive hybrid algorithm based on particle swarm optimization and differential evolution for global optimization [J], *The Scientific World Journal*. (6): 25-40.
[22] Trelea I C. (2003) The particle swarm optimization algorithm: convergence analysis and parameter selection[J]. *Information processing letters*, Elsevier, 85(6): 317-325.
[23] Van Den Bergh F. (2002) Analysis of Particle Swarm Optimizers [D], Doctoral Dissertation, Department of Computer Science, University of Pretoria, Pretoria, South Africa.
[24] Van Den Bergh F, Engelbrecht A P. (2006) A study of particle swarm optimization on particle trajectories[J]. *Information Sciences*, 176(8): 937-971.
[25] Kadirkamanathan V, Selvarajah R, Fleming P. (2006) Stability analysis of the particle dynamics in particle swarm optimizer [J]. *IEEE Transactions on Evolutionary Computation*, 10(3): 245-255.
[26] Poli R. (2009) Mean and Variance of the Sampling Distribution of Particle Swarm Optimizers During Stagnation [J]. *IEEE Transactions on Evolutionary Computation*, 13(4): 712-721.
[27] Zhang J B, Wang X L. (2010) A Novel Algorithm Based on K-Means Clustering and Binary Particle Swarm Optimization [J]. *COMPUTER TECHNOLOGY AND DEVELOPMENT*, 20(6):25-28.
[28] Jiang T, Zhang Y F, Wang Y H. (2006) A Study of Application of an Improved PSO Algorithm in BP Network [J]. *COMPUTER SCIENCE*, 33(9):164-165.
[29] Yu H S. (2015) Research of adaptive mutation of Particle Swarm Optimization [D]. Master Dissertation, Department of Applied Mathematics, Nanjing University of Information Science & Technology, Nanjing, China.
[30] Shi Y E, Berhart R C. (1998) A Modified Particle Swarms Optimizer [C]. *in Proc of IEEE Congress on Evolutionary Computation*. 6:69-73.
[31] Sun X, Zhou D W, Zhang X W. (2010) Convergence analysis and parameter selection of PSO model with inertia weight [J]. *COMPUTER ENGINEERING AND DESIGN*, 31(18):4068-4071.
[32] Clerc M. (1999) The Swarm and the Queen: Towards a Deterministic and Adaptive Particle Swarm Optimization [C], *in Proc of the Congress on Evolutionary Computation*, III: 1951-1957.
[33] Fu G J, Wang S M, Liu S Y, et al. (2005) A PSO with Bounded Mutation Operator [J]. *JOURNAL OF WUHAN UNIVERSITY OF TECHNOLOGY*, 27(9): 101-103.
[34] Lin W M, Zhou N N. (2014) A Particle Swarm Optimization Algorithm of Linear Decreasing [J]. *Computer Technology and Development*, 24(10):67-70.
[35] Meng F, Pan P P. (2011) Neural Network Trained by Hybrid Algorithms Based on Chaos Particle Swarm Optimization and Back-Propagation Algorithm [J]. *COMPUTER SIMULATION*, 28(2):196-199.