Journal of Engineering Science and Technology Review 6 (2) (2013) 178-183

Research Article

JOURNAL OF
Engineering Science and Technology Review

www.jestr.org

A Novel Adaptive Particle Swarm Optimization

Xiaobing Yu[1,*] and Jun Guo[2]

[1]*School of Economics and Management, Nanjing University of Information Science & Technology, Nanjing 210044, China*
[2]*School of Mechanic and Electronic Engineering, Wuhan University of Technology, Wuhan 430070, China*

Received 25 July 2012; Accepted

Abstract

Particle swarm optimization (PSO) is a stochastic search technique for solving optimization problems, which has been proven to be efficient and effective in wide applications. However, the PSO can easily fly into the local optima and lack the ability to jump out of the local optima. A novel adaptive PSO is proposed by evaluating convergence of the fitness value. The novel mechanism is to ensure the diversity of particles. Simulations for benchmark test functions have illustrated that the proposed algorithm possesses better ability to find the global optima than other variants and is an effective global optimization tool.

Keywords: Particle Swarm Optimization, Inertia Weight, Time-varying Acceleration Coefficients, Convergence

1. Introduction

PSO has been introduced by Kennedy and Eberhart [1] and is inspired by the emergent motion of a flock of birds searching for food [2]. As a stochastic search scheme [3], PSO has characters of simple computation and rapid convergence capability; it has been widely extended to function optimization [4].

Unfortunately, when solving complex multimodal tasks, the standard PSO can easily fly into the local optima and lack the ability to jump out of the local optima [5, 6, 7].A novel adaptive PSO is proposed by evaluating convergence of the fitness value. If the convergence is lower to one certain value, a mutation operation is carried out to the position of the best particle found so far in the swarm. The novel mechanism is to ensure the diversity of particles. Well-known nonlinear benchmark functions have been tested and experiment results have demonstrated that the performance of the proposed PSO is better than that of standard PSO, linear PSO (LPSO) and LPSO with time-varying acceleration coefficients.

The rest part of the paper is organized as follows: Section 2 provides a brief introduction of the standard PSO. Section 3 presents the novel algorithm. Numerical examples used to illustrate the efficiency of the proposed algorithm are given in Section 4. Finally, conclusions are made in section 5.

2. Standard PSO

In the standard PSO, a swarm consists of *m* individuals (called particles) that fly around in an *n*-dimensional search space. The position of the *ith* particle at the *tth* iteration is used to evaluate the particle and represent the candidate solution for the optimization problem. It can be represented as $X_i^t = [x_{i1}^t, x_{i2}^t, ..., x_{in}^t]$, where x_{ij}^t is the position value of the *ith* particle with respect to the *jth* dimension $(j = 1, 2, ..., n)$. During the search process, the position of a particle is guided by two factors: the best position visited by itself (P_{best}) denoted as $P_i^t = [p_{i1}^t, p_{i2}^t, ..., p_{in}^t]$, and the position of the best particle found so far in the swarm (g_{best}) denoted as $G^t = [g_1^t, g_2^t, ..., g_n^t]$. The new velocity (denoted as $V_i^t = [v_{i1}^t, v_{i2}^t, ..., v_{in}^t]$) and position of particle *i* at the next iteration are calculated according to:

$$v_{ij}^{t+1} = w \times v_{ij}^t + c_1 \times r_1 \times (p_{ij}^t - x_{ij}^t) + c_2 \times r_2 \times (g_j^t - x_{ij}^t) \quad (1)$$

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

where w is the inertia weight, c_1, c_2 are respectively the cognitive and social learning parameters, and r_1, r_2 are random numbers between (0,1). Based on the above equations, the particle can fly through search space toward P_{best} and g_{best} in a navigated way [8, 9, 10].

3. Adaptive Particle Swarm Optimization (APSO)

As PSO has no ability to jump out of the local optima, APSO is proposed to improve the performance of PSO. It focuses on the convergence of particles. When a particle discovers a current optima position, the other particles will

* E-mail address: yuxb111@163.com
ISSN: 1791-2377

draw together to the particle. If the position is the local optima, the PSO will be convergence and clustered in local optima. The premature may appear. Suppose that the population size of APSO is N , the fitness value of ith particle is f_i and the average fitness value is f_{avg} . The convergence degree is defined as following:

$$c = \sqrt{\sum_{i=1}^{N} \left(\frac{f_i - f_{avg}}{\max\{1, \max_{1 \le i \le N}(f_i - f_{avg})}\right)^2} \quad (3)$$

The parameter c reflects the convergence degree. When the parameter c is large, the algorithm is in random search. On the other hand, the algorithm will be convergence and premature maybe occur. In order to evaluate the parameter c , σ_c is given as following, where p_m is the mutation probability.

$$p_m = \begin{cases} k & c < \sigma_c \\ 0 & others \end{cases} \quad (4)$$

Generally, $\sigma_c \in [0.5, 2]$ .If the parameter c is less than σ_c , the mutation probability p_g is equal to k and is as following:

$$p_g = (1 + 0.5\eta) p_g \quad (5)$$

where the parameter η obeys Gauss(0,1) distribution.

With the efforts, the APSO has the ability to jump out of the local optima. Besides, according to the research [5,11], the inertia weight, cognitive and social learning parameters are adjusted by Eq.(6-8). The nonlinear descending can achieve faster convergence speed than that with linear inertia weight.

$$w = (w_1 - w_2) \times \frac{(iter - iter_{\max})^2}{(iter_{\max})^2} + w_2 \quad (6)$$

$$c_1 = (c_{1f} - c_{1i})(\frac{iter}{iter_{\max}}) + c_{1i} \quad (7)$$

$$c_2 = (c_{2f} - c_{2i})(\frac{iter}{iter_{\max}}) + c_{2i} \quad (8)$$

where w_1, w_2 are the initial and final values of weight, $c_{1i}, c_{1f}, c_{2i}, c_{2f}$ are initial and final values of cognitive and social acceleration factors respectively, usually $c_{1i} = c_{2f} = 2.5$ and $c_{1f} = c_{2i} = 0.5$.

The main procedure of APSO is presented in Table 1.

Table 1: The main procedure of APSO

for i=1 to the swarm size do
  Initialize x_i , v_i within the search range randomly
End for
Evaluate each particle
$P_i = x_i$ and Identify the best position P_g
While $iter \le Iter_{\max}$
  Update weight w by Eq.(6)
  Update acceleration coefficients c_1 and c_2 by Eq.(7,8) respectively
  for i=1 to the swarm size do
    Update velocity v_i^{t+1} and position x_i^{t+1} according to Eq.(1,2) respectively.
    Evaluation fitness(x_i^{t+1})
    If x_i^{t+1} is better than P_i
      set $P_i = x_i^{t+1}$
    End if
    If x_i^{t+1} is better than P_g
      set $P_g = x_i^{t+1}$
    End if
End for
Calculate parameter c using Eq.(3)
  If parameter meets the requirement of Eq.(4)
    Generate a random number *rand* in (0,1).
    If *rand* is less than p_m Update P_g using Eq.(5)
    End if
  End if
End while

4. Experimental Results
4.1Experiment Setup

Eight benchmark functions are listed in Table 2 and (9)-(16) are utilized to test the performance of APSO. Asymmetric initializations are used for the functions whose global optimum is at the center of the search range. Algorithms parameters initializations are presented in Table 3. For all test functions, PSO algorithms including SPSO, LPSO and PSO-TVAC[11] carry out 50 independent runs in order to eliminate random discrepancy.

Table 2: Benchmark configurations

Function	Name	Search Space	Initial Range
f_1	Sphere	[-100,100]	[-100,50]
f_2	Weighted Sphere	[-100,100]	[-100,50]
f_3	Rosenbrock	[-5,5]	[-5,2]
f_4	Griewank	[-600,600]	[-600,200]
f_5	Rastrigrin	[-5,5]	[-5,2]
f_6	Noncontinuous Rastrigin	[-5,5]	[-5,2]
f_7	Ackely	[-32,32]	[-32,16]
f_8	Penalized	[-50,50]	[-50,20]

Table 3: Algorithms Parameters Initializations.

Algorithm	Parameters
SPSO	$w = 0.729; c_1 = c_2 = 1.49$
LPSO	$w_{\max} = 0.9; w_{\min} = 0.4; c_1 = c_2 = 2$
LPSO -TVAC	$w_{\max} = 0.9; w_{\min} = 0.4;$ $c_{1i} = c_{2f} = 2.5; c_{1f} = c_{2i} = 0.5$
APSO	$w_1 = 0.9; w_2 = 0.2;$ $c_{1i} = c_{2f} = 2.5; c_{1f} = c_{2i} = 0.5;$ $\sigma_c = 1.2; k = 0.3$

The maximum number of generations is set as 1000 and 1500 corresponding to dimensions 10 and 20 respectively. $f_1(x)$ and $f_2(x)$ are used to test the convergence speeds of algorithms. It is very difficult to optimize $f_3(x)$ which can be viewed as a multimodal problem. $f_4(x)$ to $f_8(x)$ are also multimodal problems which are hard to optimize.

$$f_1 = \sum_{i=1}^{n} x_i^2 \tag{9}$$

$$f_2 = \sum_{i=1}^{n} i x_i^2 \tag{10}$$

$$f_3 = \sum_{i=1}^{n} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2] \tag{11}$$

$$f_4 = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{i^{1/2}}) + 1 \tag{12}$$

$$f_5 = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10] \tag{13}$$

$$f_6(x) = \sum_{i=1}^{n} (y_i^2 - 10\cos(2\pi y_i) + 10), y_i = \begin{cases} x_i & |x_i| < 0.5 \\ \frac{round(2x_i)}{2} & |x_i| \geq 0.5 \end{cases} \tag{14}$$

$$f_7 = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)) + 20 + e \tag{15}$$

$$f_8(x) = \frac{\pi}{n}\{10\sin^2(\pi y_i) + \sum_{i=1}^{n}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$$

$$y_i = (1 + \frac{1}{4}(x_i + 1)), u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \leq x_i \leq a \\ k(-x_i - a)^m & x_i < -a \end{cases} \tag{16}$$

4.2. Performance Comparison

The mean solutions of the algorithms in 50 independent runs are listed in Table 4. The best results among these PSO algorithms are indicated by boldface. Figs. 1 – 8 show the comparisons in terms of evolution processes in solving the eight benchmark functions.

Table 4: The mean fitness values for test functions of PSO, LPSO, LPSO – TVAC and APSO

Function	Dimension	Generation	SPSO	LPSO	PSO-TVAC	APSO
Sphere	10	1000	0.2196	3.0936e-5	1.3120e-6	**2.5092e-184**
	20	1500	7.3770	0.0653	0.0528	**0**
Weighted Sphere	10	1000	2.7986	4.2905e-5	9.3364e-8	**0**
	20	1500	134.201	1.9801	0.5226	2.1649e-139
Rosenbrock	10	1000	13.6857	6.5100	5.2725	5.2294
	20	1500	338.4549	29.7961	24.5617	16.2419
Griewank	10	1000	3.1070	0.2406	0.2119	**0**
	20	1500	23.3454	0.5490	0.8046	**0**
Rastrigrin	10	1000	20.9339	10.9448	0.0698	**0**
	20	1500	94.5326	49.8762	38.8608	**0**
Non continuous Rastrigin	10	1000	18.0000	12.0200	8.0000	**0**
	20	1500	58.0000	47.0000	39.0000	0.9000
Ackely	10	1000	5.8409	3.2224	0.0023	8.8818e-16
	20	1500	14.8498	1.6440	1.2093	8.8818e-16
Penalized	10	1000	9.1749	0.9632	1.8741e-4	5.3945e-7
	20	1500	3.851.4	1.9195	0.0194	4.6805e-6

From the Table 4 and Figs. 1 – 8, it is very clear that the proposed PSO has the strong ability to jump out of the local optima. It can effectively prevent the premature convergence and significantly enhance the convergence rate and accuracy. It provides best performance on the f_4, f_5 and f_6, which reach the highest accuracy on them. The APSO ranks the second on f_1, f_2, f_7 and f_8. The performances of LPSO and LPSO-TVAC are among APSO and SPSO. The LPSO-TVAC is a little better than that of LPSO. It's easy to see that the SPSO algorithm converges quickly and slows its convergence speed down when reaching the local optima, which exhibits significant prematurely. All the algorithms perform badly on f_3 which approves that the function is very hard to optimize. With the increasing of population size, the performance of SPSO, LPSO and LPSO-TVAC become bad.

Through the thorough comparison, the performance of APSO is the best among the four algorithms.

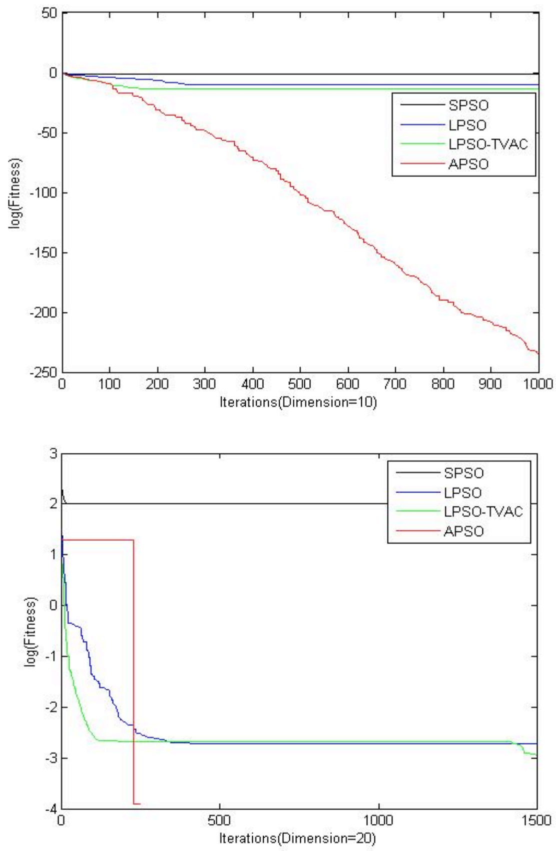
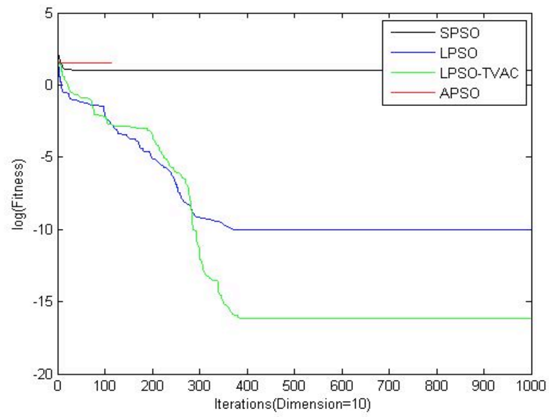


Fig.1. Evolution of logarithmic average fitness of Sphere function for SPSO, LPSO, LPSO-TVAC and APSO.



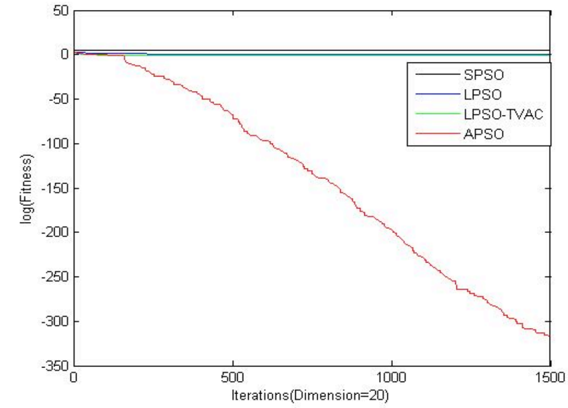


Fig.2. Evolution of logarithmic average fitness of Weighted Sphere function for SPSO, LPSO, LPSO-TVAC and APSO.

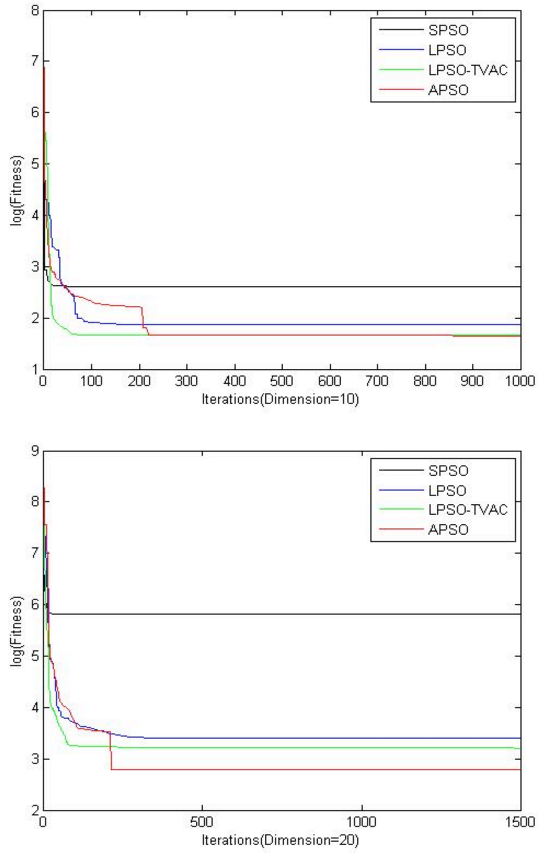
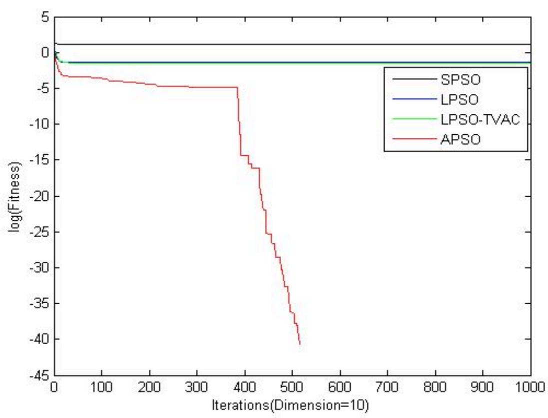


Fig.3. Evolution of logarithmic average fitness of Rosenbrock function for SPSO, LPSO, LPSO－TVAC and APSO.



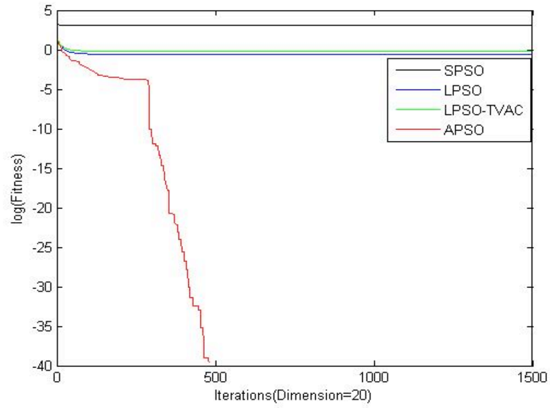


Fig.4. Evolution of logarithmic average fitness of Griewank function for SPSO, LPSO, LPSO－TVAC and APSO.

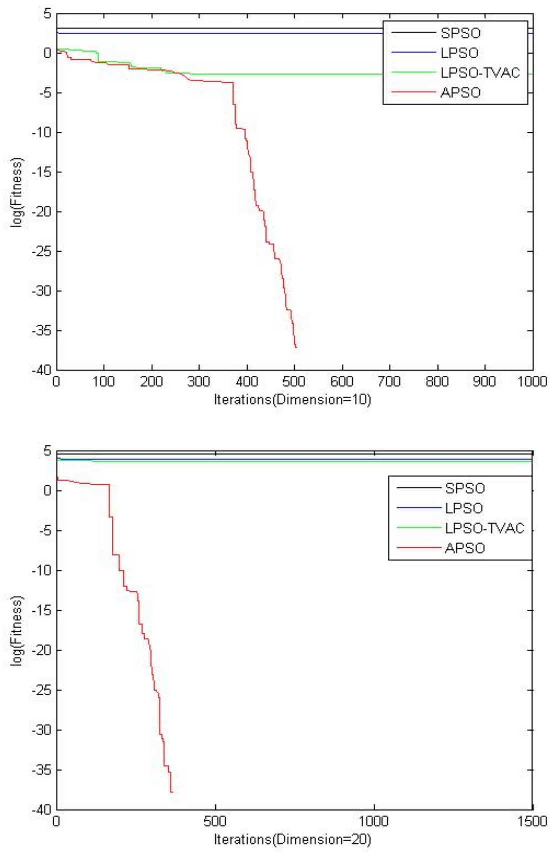
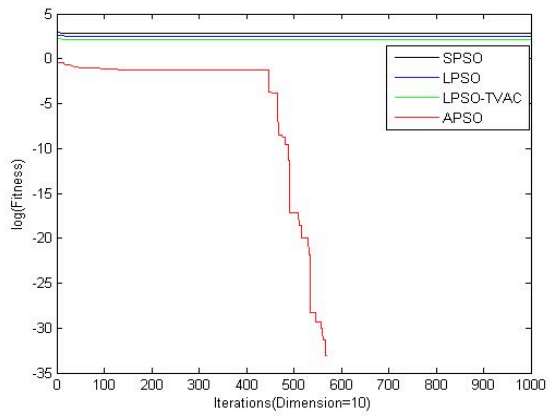


Fig.5. Evolution of logarithmic average fitness of Rastrigrin function for SPSO, LPSO, LPSO–TVAC and APSO.



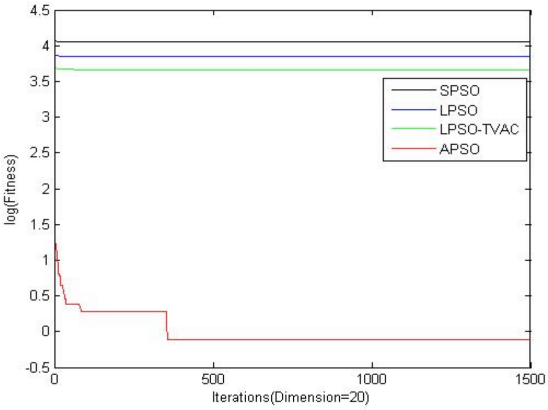


Fig.6. Evolution of logarithmic average fitness of Non continuous Rastrigrin function for SPSO, LPSO, LPSO－TVAC and APSO.

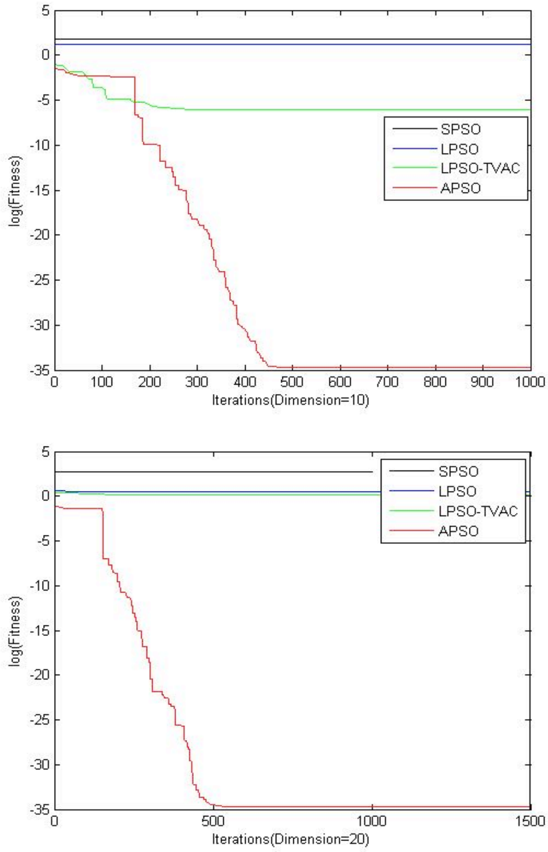


Fig.7. Evolution of logarithmic average fitness of Ackely function for SPSO, LPSO, LPSO－TVAC and APSO.

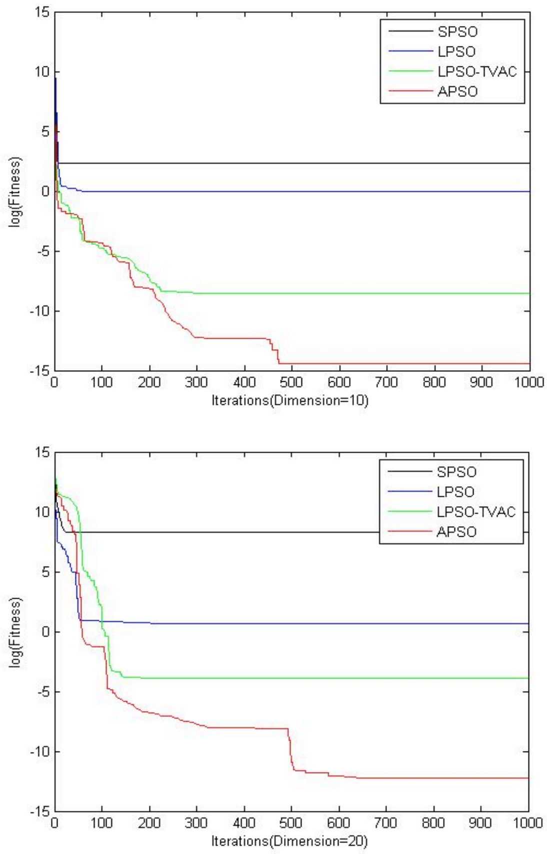


Fig.8. Evolution of logarithmic average fitness of Penalized function for SPSO, LPSO, LPSO－TVAC and APSO.

5. Conclusions

In this paper, a novel PSO algorithm is proposed. The main procedure of this new variant of PSO is presented. This new approach can enhance diversity by mutation. The new adaptive PSO is discussed in comparison with SPSO, LPSO, and LPSO-TVAC through empirical simulations with well-known benchmark functions from the standard literature. Results have shown that the novel PSO is a promising method with good global convergence performance.

Acknowledgements

This work was supported by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD), Social Science Foundation of Chinese Ministry of Education (No.12YJC630271), Natural science fund for colleges and universities in Jiangsu Province (No.13KJB120008) and China Natural Science Foundation (No. 71273139).

References

1. Kennedy,J., Eberhart,R.C., “Particle swarm optimization”, Proc. IEEE International Conference on Neural Networks IV, Perth, Australia, IEEE Service Center, Piscataway, NJ (1995), pp. 1942–1948
2. Zhao,Y.X., Zu,W., Zeng,H.T., “A modified particle swarm optimization via particle visual modeling analysis”, *Computers & Mathematics with Applications* 57(11-12),2009,pp. 2022-2029
3. Wang,Y., Li,B., Weise,T., Wang,J.Y., Yuan,B., Tian,Q.j., “Self-adaptive learning based particle swarm optimization” *Information Sciences* 181(20),2011,pp.4515-4538.
4. Liang,J.J., Qin,A.K., Suganthan,P.N., Baskar,S., “Comprehensive learning particle swarm optimizer for global optimization of multimodal functions”, *IEEE Trans. Evol. Comput* 10 (3) ,2006,pp.281-295.
5. Tang,Y., Wang,Z.D., Fang,J.A., “Feedback learning particle swarm optimization”, *Applied Soft Computing* 11(8),2011,pp.4713-4725.
6. Yu,Y.Z.,Ren,X.Y.,Du,F.S.,Shi,J.J.,“Application of Improved PSO Algorithm in Hydraulic Pressing System Identification”, *Journal of Iron and Steel Research*, International 19(9), 2012, pp.29-35.

7. Jia,D.L., Zheng,G.X., Qu,B.Y., Muhammad K.K.,"A hybrid particle swarm optimization algorithm for high-dimensional problems", *Computers & Industrial Engineering* 61(4), 2011, pp.1117-1122
8. Chen,D.B.,Zhao,C.X, "Particle swarm optimization with adaptive population size and its application", *Applied Soft Computing* 9(1), 2009, pp.39-48.
9. Wang,X., Tang,L., "A discrete particle swarm optimization algorithm with self-adaptive diversity control for the permutation flowshop problem with blocking", *Applied Soft Computing* 12(2),2012,pp.652-662.
10. Zhu,H.H., Wang,Y., Wang,K., Chen,Y., Particle Swarm Optimization (PSO) for the constrained portfolio optimization problem, *Expert Systems with Applications* 38(8), 2011,pp.10161-10169.
11. Tripathi,P.K., Bandyopadhyay.S., Pal,S.K., "Multi-objective particle swarm optimization with time variant inertia and acceleration coefficients", *Information Science* 177(22),2007,5033-5049.