Journal of Zhejiang University SCIENCE A
ISSN 1673-565X (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

θ-PSO: a new strategy of particle swarm optimization*

Wei-min ZHONG[1,2], Shao-jun LI[1,2], Feng QIAN[‡1,2]

([1]State Key Laboratory of Chemical Engineering, East China University of Science and Technology, Shanghai 200237, China)

([2]Automation Institute, East China University of Science and Technology, Shanghai 200237, China)

E-mail: {wmzhong, lishaojun, fqian}@ecust.edu.cn

Received May 31, 2007; revision accepted Aug. 4, 2007; published online May 2, 2008

Abstract: Particle swarm optimization (PSO) is an efficient, robust and simple optimization algorithm. Most studies are mainly concentrated on better understanding of the standard PSO control parameters, such as acceleration coefficients, etc. In this paper, a more simple strategy of PSO algorithm called $\boldsymbol{\theta}$-PSO is proposed. In $\boldsymbol{\theta}$-PSO, an increment of phase angle vector replaces the increment of velocity vector and the positions are decided by the mapping of phase angles. Benchmark testing of nonlinear functions is described and the results show that the performance of $\boldsymbol{\theta}$-PSO is much more effective than that of the standard PSO.

Key words: Particle swarm optimization (PSO), Phase angle, Benchmark function
doi:10.1631/jzus.A071278 **Document code:** A **CLC number:** TP301.6

INTRODUCTION

Particle swarm optimization (PSO) is one of the evolutionary computation techniques introduced by Kennedy and Eberhart in 1995 (Kennedy and Eberhart, 1995; Eberhart and Kennedy, 1995). It is a population-based search algorithm and is initialized with a population of random solutions, named particles. PSO makes use of a velocity vector to update the current position of each particle in the swarm. The velocity vector is updated based on the history information gained by the swarm. And the positions of the swarm are updated to search for better positions according to the updated velocity vector (Shi and Eberhart, 1998; Clerc, 1999; Trelea, 2003). Recently, many researchers have studied the performance of PSO, mostly about the basic control parameters, such as the acceleration coefficients, inertia weight, velocity clamping, and swarm size (Kennedy and Eberhart, 2001; Zhang *et al*., 2005; Lee and Chen, 2007; Fan and Zahara, 2007; Ho *et al*., 2007). From these empirical studies, it can be concluded that PSO is sensitive to control parameters, but few studies are involved in the basic mechanism. In the standard PSO, velocity is an important parameter and is dynamically adjusted according to the historical behaviors of the particle and its companions. In this paper, we put forward a new PSO algorithm called $\boldsymbol{\theta}$-PSO, which is based on the phase angle vector but not the velocity vector. In $\boldsymbol{\theta}$-PSO, an increment of phase angle vector $\Delta\boldsymbol{\theta}$ replaces velocity vector $\boldsymbol{v}$ and the positions are adjusted by the mapping of phase angles. Benchmark testing of nonlinear functions is described in detail and $\boldsymbol{\theta}$-PSO appears to be a promising approach of function optimization.

This paper is organized as follows: the standard PSO and $\boldsymbol{\theta}$-PSO algorithms are described in Section 2; in Section 3, benchmark functions are tested and optimization results are discussed; some conclusions and views are put forward in Section 4.

$\boldsymbol{\theta}$-PARTICLE SWARM OPTIMIZATION

In order to illustrate our new $\boldsymbol{\theta}$-PSO, we first

‡ Corresponding author
* Project supported by the National Natural Science Foundation of China (Nos. 60625302 and 60704028), the Program for Changjiang Scholars and Innovative Research Team in University (No. IRT0721), the 111 Project (No. B08021), the Major State Basic Research Development Program of Shanghai (No. 07JC14016) and Shanghai Leading Academic Discipline Project (No. B504) of China

introduce the standard algorithm. PSO is a stochastic optimization approach, which maintains a swarm of candidate solutions.

Standard algorithm

The standard PSO is described in vector notation as follows (Kennedy and Eberhart, 1995; 2001):

$$\begin{aligned}\boldsymbol{v}_i(t+1)=\omega\boldsymbol{v}_i(t)+c_1\boldsymbol{r}_1(t)(\boldsymbol{p}_i(t)-\boldsymbol{x}_i(t))\\+c_2\boldsymbol{r}_2(t)(\boldsymbol{p}_\mathrm{g}(t)-\boldsymbol{x}_i(t)),\end{aligned}\quad(1)$$

$$\boldsymbol{x}_i(t+1)=\boldsymbol{x}_i(t)+\boldsymbol{v}_i(t+1),\quad(2)$$

for i=1, 2, …, s. Where s is the swarm size, c_1 and c_2 the acceleration coefficients, ω the inertia weight, $\boldsymbol{r}_1(t)$ and $\boldsymbol{r}_2(t)\sim U(0, 1)$, $\boldsymbol{x}_i(t)$ the position of particle i at time t, $\boldsymbol{v}_i(t)$ the velocity of particle i at time t, $\boldsymbol{p}_i(t)$ the personal best solution of particle i at time t, and $\boldsymbol{p}_\mathrm{g}(t)$ the global best solution at time t.

The particle position $\boldsymbol{x}_i(t+1)$ is updated using its current value and the newly computed velocity $\boldsymbol{v}_i(t+1)$, which is determined by the values of $\boldsymbol{v}_i(t)$, $\boldsymbol{x}_i(t)$, $\boldsymbol{p}_i(t)$, $\boldsymbol{p}_\mathrm{g}(t)$ and coefficients ω, c_1 and c_2.

***θ*-PSO algorithm**

In $\boldsymbol{\theta}$-PSO, the increment of phase angle replaces the increment of velocity and the position is decided by the mapping of the phase angle. $\boldsymbol{\theta}$-PSO can be described in vector notation as follows:

$$\begin{aligned}\Delta\boldsymbol{\theta}_i(t+1)=\omega\Delta\boldsymbol{\theta}_i(t)+c_1\boldsymbol{r}_1(t)(\boldsymbol{\theta}_{i\mathrm{b}}(t)-\boldsymbol{\theta}_i(t))\\+c_2\boldsymbol{r}_2(t)(\boldsymbol{\theta}_\mathrm{g}(t)-\boldsymbol{\theta}_i(t)),\end{aligned}\quad(3)$$

$$\boldsymbol{\theta}_i(t+1)=\boldsymbol{\theta}_i(t)+\Delta\boldsymbol{\theta}_i(t+1),\quad(4)$$

$$\boldsymbol{x}_i(t)=f(\boldsymbol{\theta}_i(t)),\quad(5)$$

$$F_i(t)=\mathit{fitnessvalue}(\boldsymbol{x}_i(t)),\quad(6)$$

with $\theta_{ij}\in(\theta_{\min}, \theta_{\max})$, $\Delta\theta_{ij}\in(\Delta\theta_{\min}, \Delta\theta_{\max})$, $x_{ij}\in(x_{\min}, x_{\max})$ and f being a monotonic mapping function, i=1, 2, …, s and j=1, 2, …, n. Where we assume the global optimal particle is not on the boundary; s, c_1, c_2, ω, $\boldsymbol{r}_1(t)$, $\boldsymbol{r}_2(t)$, and $\boldsymbol{x}_i(t)$ are the same as those in Eqs.(1) and (2); n is the dimension of the problem; $\boldsymbol{\theta}_i(t)$ the phase angle of particle i at time t; $\Delta\boldsymbol{\theta}_i(t)$ the increment of particle i's phase angle at time t; $\boldsymbol{\theta}_{i\mathrm{b}}(t)$ the phase angle of the personal best solution of particle i at time t; $\boldsymbol{\theta}_\mathrm{g}(t)$ the phase angle of the global best solution at time t; $F_i(t)$ the fitness value of particle i at time t, which is decided by the function *fitnessvalue*; $F_{i\mathrm{b}}(t)$ the personal best fitness value of particle i at time t; $F_\mathrm{g}(t)$ the global best fitness value at time t.

In this paper, we set $\theta_{ij}\in(-\pi/2, \pi/2)$, $\Delta\theta_{ij}\in(-\pi/2, \pi/2)$ and

$$f(\theta_{ij})=\frac{x_{\max}-x_{\min}}{2}\sin\theta_{ij}+\frac{x_{\max}+x_{\min}}{2}.\quad(7)$$

The $\boldsymbol{\theta}$-PSO algorithm can be summarized as follows:

```
Create and initialize an n-dimensional swarm (phase angle
  θi(1));
Repeat t=1, 2, …, iteration number;
for each particle i=1, 2, …, s
  if t=1
    calculate xi(1) using Eq.(5);
    calculate the fitness value Fi(1) using Eq.(6);
    Fib(1)=Fi(1); θib(1)=θi(1);
    Fg(1)=Fi(1); θg(1)=θi(1);
  else if t>1
    update the increment of the phase angle Δθi(t) using
      Eq.(3) and limit Δθi(t) to (Δθmin, Δθmax);
    update θi(t) using Eq.(4) and limit θi(t) to (θmin, θmax);
    update xi(t) using Eq.(5);
    update the fitness value Fi(t) using Eq.(6);
    if Fi(t)<Fib(t)
      Fib(t)=Fi(t); θib(t)=θi(t);
    end
    if Fi(t)<Fg(t)
      Fg(t)=Fi(t); θg(t)=θi(t);
    end
  end
end      // until the stopping condition is true
```

OPTIMIZATION EXPERIMENTS

The $\boldsymbol{\theta}$-PSO algorithm [Eqs.(3)~(6)] and the standard PSO algorithm [Eqs.(1)~(2)] are used for optimization of several benchmark functions, named "camel", "Levy f3", "Jason", "Sphere", "Griewank" and "Rosenbrock". Two sets of parameters recommended in published papers with w=0.6, c_1=c_2=1.7 (Trelea, 2003) and w=0.729, c_1=c_2=1.494 (Clerc, 1999) are used. The functions, the number of dimensions, the admissible range of the input $\boldsymbol{x}$, and the optimum and acceptable tolerance [the three functions in (Trelea, 2003) are selected here and the tolerances of "Sphere", "Griewank" and "Rosenbrock" in (Trelea, 2003) are 0.01, 0.1 and 100, respectively] are summarized in Table 1. The maximum number of iterations is fixed to 10000. Each optimization experiment is run 20 times with a random initial value of $\boldsymbol{\theta}$. Swarm sizes of s=20 and 40 particles are tested, respectively. During the optimization process, $\boldsymbol{\theta}$ and $\Delta\boldsymbol{\theta}$ are restricted within $(-\pi/2, \pi/2)$. Eq.(7) guarantees the particles not be able to "fly" outside the region defined by $(x_{\min}, x_{\max})$. And the optimization performance is listed in Table 2. Furthermore, another group of experiments are conducted using "Jason" with w=0.6, c_1=c_2=1.7 to illustrate the effect of the dimension. Each optimization experiment is also run 20 times with a random initial status. The information and results are listed in Table 3.

Table 1 List of optimization test functions

Function	Formula	Dimension n	Admissible range of $\boldsymbol{x}$	Global optimal fitness value	Admissible optimization tolerance
Camel	$f_1=(4-2.1x_1^2+x_1^4/3)x_1^2+x_1x_2+(-4+4x_2^2)x_2^2$	2	(−100, 100)	−1.0316	0.0001
Levy f3	$f_2=\sum_{i=1}^{5}\big(i\cos((i-1)x_1+i)\big)\cdot\sum_{i=1}^{5}\big(i\cos((i+1)x_2+i)\big)$	2	(−100, 100)	−176.5418	0.0001
Jason	$f_3=\sum_{i=1}^{n}(x-i)^2$	10	(−100, 100)	0	0.0001
Sphere	$f_4=\sum_{i=1}^{n}x_i^2$	30	(−100, 100)	0	0.0001
Griewank	$f_5=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos(x_i/\sqrt{i})+1$	30	(−600, 600)	0	0.1
Rosenbrock	$f_6=\sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	30	(−30, 30)	0	20

Table 2 Optimization performance of θ-PSO and the standard PSO

Function	Algorithm	Swarm size s	Number of iterations to achieve the goal				Success rate (×100%)	
			w=0.6, c_1=c_2=1.7		w=0.729, c_1=c_2=1.494		w=0.6, c_1=c_2=1.7	w=0.729, c_1=c_2=1.494
			Minimum	Average	Minimum	Average		
Camel	$\boldsymbol{\theta}$-PSO	20	33	45	44	67	1	1
	PSO	20	38	69	69	113	1	1
	$\boldsymbol{\theta}$-PSO	40	26	40	40	62	1	1
	PSO	40	32	60	54	103	1	1
Levy f3	$\boldsymbol{\theta}$-PSO	20	66	162	65	189	1	1
	PSO	20	60	185	84	305	1	1
	$\boldsymbol{\theta}$-PSO	40	50	148	77	156	1	1
	PSO	40	69	152	80	150	1	1
Jason	$\boldsymbol{\theta}$-PSO	20	119	147	152	170	1	1
	PSO	20	172	210	255	348	1	1
	$\boldsymbol{\theta}$-PSO	40	104	114	139	151	1	1
	PSO	40	153	184	266	326	1	1
Sphere	$\boldsymbol{\theta}$-PSO	20	533	598	362	734	1	1
	PSO	20	670	853	1198	1365	1	1
	$\boldsymbol{\theta}$-PSO	40	352	406	266	683	1	1
	PSO	40	577	716	855	1188	1	1
Griewank	$\boldsymbol{\theta}$-PSO	20	343	512	385	564	1	0.95
	PSO	20	463	1350	751	948	0.8	0.75
	$\boldsymbol{\theta}$-PSO	40	231	334	263	356	1	1
	PSO	40	375	461	558	742	1	0.9
Rosenbrock	$\boldsymbol{\theta}$-PSO	20	223	376	328	402	1	1
	PSO	20	792	3258	2645	4637	0.5	0.6
	$\boldsymbol{\theta}$-PSO	40	194	283	272	325	1	1
	PSO	40	569	2268	866	3087	0.55	0.65

Table 3 Optimization performance using function "Jason" with w=0.6, c_1=c_2=1.7

Dimension n	Swarm size s	Algorithm	Admissible range of $\boldsymbol{x}$	Stopping condition		Number of iterations to achieve the goal		Success rate (×100%)
				Maximum number of iterations	Tolerance	Minimum	Average	
20	40	$\boldsymbol{\theta}$-PSO	(−100, 100)	10000	0.0001	211	256	1
		PSO				277	314	1
30	40	$\boldsymbol{\theta}$-PSO	(−100, 100)	10000	0.0001	320	438	1
		PSO				527	642	1
40	40	$\boldsymbol{\theta}$-PSO	(−100, 100)	10000	0.0001	554	806	1
		PSO				807	923	1
50	40	$\boldsymbol{\theta}$-PSO	(−100, 100)	10000	0.0001	974	1244	1
		PSO				1008	1376	1
60	40	$\boldsymbol{\theta}$-PSO	(−100,100)	10000	0.0001	2014	2539	1
		PSO				2674	2976	0.3
70	40	$\boldsymbol{\theta}$-PSO	(−100, 100)	10000	0.0001	2101	3194	1
		PSO				–	–	0
100	40	$\boldsymbol{\theta}$-PSO	(−200, 200)	10000	0.0001	4946	5890	1
200	40	$\boldsymbol{\theta}$-PSO	(−300, 300)	30000	0.1	19156	22952	0.9
300	100	$\boldsymbol{\theta}$-PSO	(−400, 400)	40000	1	19879	24558	0.8
400	150	$\boldsymbol{\theta}$-PSO	(−500, 500)	50000	10	18725	27635	0.8
500	150	$\boldsymbol{\theta}$-PSO	(−600, 600)	60000	10	32873	43062	0.65
600	150	$\boldsymbol{\theta}$-PSO	(−700, 700)	60000	100	40070	56039	0.55

Comparison between $\boldsymbol{\theta}$-PSO and PSO

All analyses and conclusions are based on the setting optimization conditions in this paper.

Tables 2 and 3 show that the performance of $\boldsymbol{\theta}$-PSO is much better than that of the standard PSO. For all six benchmark functions except "Levy f3", the minimum number of iterations, the average number of iterations and success rates are all better than those of the standard PSO. Therefore, $\boldsymbol{\theta}$-PSO is much time-saving compared with the standard PSO. For "Levy f3", the results of $\boldsymbol{\theta}$-PSO and the standard PSO are almost the same. For complex and multi-minima functions "Griewank" and "Rosenbrock", the minimum and average number of iterations and success rates of $\boldsymbol{\theta}$-PSO are improved dramatically in general compared with those of the standard PSO. Therefore, $\boldsymbol{\theta}$-PSO can jump out of the local minima much more easily than the standard PSO. And $\boldsymbol{\theta}$-PSO is much more effective in dealing with high-dimensional cases than the standard PSO according to the results shown in Table 3. Standard PSO cannot solve the 70-dimensional "Jason" with the tolerance of 0.0001 within 10000 iterations using 40 particles. But $\boldsymbol{\theta}$-PSO can deal well with 100-dimensional "Jason" with the tolerance of 0.0001 within 10000 iterations at 100% success rate using 40 particles. Moreover, $\boldsymbol{\theta}$-PSO can even do 600-dimensional "Jason" with much more particles and a bigger tolerance. The results of comparison between $\boldsymbol{\theta}$-PSO and the standard PSO are attractive. The algorithm performance is dramatically improved for the introduced phase angle $\boldsymbol{\theta}$.

Effect of the swarm size s

Like the standard PSO algorithm, in most cases the number of iterations of $\boldsymbol{\theta}$-PSO decreases when the swarm size s increases, as indicated by the minimum and average numbers of iterations listed in Table 2. For the complex functions "Griewank" and "Rosenbrock", the success rate increases when s increases. But a large s will cost more time, and more function evaluations are needed per iteration. The best results are obtained with 40 particles in both $\boldsymbol{\theta}$-PSO and PSO. The results of $\boldsymbol{\theta}$-PSO with 40 particles are better than those of PSO with 20 particles.

Effect of the parameters w, c_1 and c_2

Trelea (2003) reported that PSO with w=0.6, c_1=c_2=1.7 had a higher convergence rate than that with w=0.729, c_1=c_2=1.494. In our experiments, the average numbers of iterations with w=0.6, c_1=c_2=1.7

are generally smaller for "Jason". And the success rates with w=0.6, c_1=c_2=1.7 are generally smaller than those with w=0.729, c_1=c_2=1.494. It approves that the risk of premature convergence to non-optimal points of PSO with w=0.6, c_1=c_2=1.7 is higher. But the parameters have no distinct influence on θ-PSO in the success rate, namely, θ-PSO is much more insensitive to w, c_1 and c_2 than the standard PSO.

Effect of the dimension *n*

In order to analyze the effect of dimension on θ-PSO, we may review the results of different functions with different dimensions (n) listed in Table 2 and the results of "Jason" with different n listed in Table 3. Generally, the performance is better when n is smaller. And larger success rates are achieved for the low-dimensional functions. For "Jason", when n>200, the success rate starts to decrease and the optimization time increases dramatically. In the case of n=600 and s=150, each optimization costs about 30 min, while in the case of n=100 and s=40, each optimization costs only about several seconds.

SUMMARY

The behavior and performance of the θ-PSO algorithm are tested to be compared with those of the standard PSO algorithm. Tests are carried out for two parameter sets, different swarm sizes and six benchmark functions. The results of θ-PSO are better than those of the standard PSO with using the parameters recommended by (Clerc, 1999; Trelea, 2003). The main improvements are in the optimization precision, number of iterations, optimization time, and success rate. And θ-PSO can deal with high-dimensional cases much more easily than the standard PSO. In addition, by observing the functions of "Sphere" and "Jason", we find no "origin-seeking bias" problem in the θ-PSO algorithm. In a word, θ-PSO remarkably enhances the optimization performance.

Further study focusing on the effect of dynamic behavior and on the convergence of θ-PSO is needed to clarify what are the causes of the obvious improvement. And the evolutional mechanism is another attractive point to obtain a high success rate in dealing with complex and high-dimensional cases.

References

Clerc, M., 1999. The Swarm and the Queen: Towards a Deterministic and Adaptive Particle Swarm Optimization. Proc. ICEC. Washington, DC, p.1951-1957.

Eberhart, R.C., Kennedy, J., 1995. A New Optimizer Using Particle Swarm Theory. Proc. 6th Int. Symp. on Micro Machine and Human Science. IEEE Service Center, Piscataway, NJ, p.39-43.

Fan, S.K.S., Zahara, E., 2007. A hybrid simplex search and particle swarm optimization for unconstrained optimization. *Eur. J. Oper. Res.*, **181**(2):527-548. [doi:10.1016/j.ejor.2006.06.034]

Ho, S.L., Yang, S.Y., Ni, G.Z., Wong, K.F., 2007. An improved PSO method with application to multimodal functions of inverse problems. *IEEE Trans. on Magn.*, **43**(4):1597-1600. [doi:10.1109/TMAG.2006.892108]

Kennedy, J., Eberhart, R.C., 1995. Particle Swarm Optimization. Proc. 5th Int Conf. on Neural Networks. IEEE Service Center, Piscataway, NJ, p.1942-1948.

Kennedy, J., Eberhart, R.C., 2001. Swarm Intelligence. Morgan Kaufmann Publishers, Inc., San Francisco, CA.

Lee, T.Y., Chen, C.L., 2007. Iteration particle swarm optimization for contract capacities selection of time-of-use rates industrial customers. *Energy Conv. Manag.*, **48**(4): 1120-1131. [doi:10.1016/j.enconman.2006.10.013]

Shi, Y., Eberhart, R.C., 1998. A Modified Particle Swarm Optimizer. Proc. IEEE World Congress on Computational Intelligence, p.69-73. [doi:10.1109/ICEC.1998.699146]

Trelea, I.C., 2003. The particle swarm optimization algorithm: convergence analysis and parameter selection. *Inf. Process. Lett.*, **85**(6):317-325. [doi:10.1016/S0020-0190(02)00447-7]

Zhang, L.P., Yu, H.J., Hu, S.X., 2005. Optimal choice of parameters for particle swarm optimization. *J. Zhejiang Univ. Sci.*, **6A**(6):528-534. [doi:10.1631/jzus.2005.A0528]