An Improved Particle Swarm Optimization Algorithm Based on Velocity Updating

Jinglei Guo Zhijian Wu Zhejun Wu
State Key Lab of Software Engineering
Wuhan University
Wuhan, China
guojinglei@mail.ccnu.edu.cn, zjwu9551@sina.com jayily1011@163.com

***Abstract*—The particle swarm optimization is a stochastic optimization technique for finding optimal regions of complex problems through the interaction of individuals in the swarm. In this paper the search trajectory of particle is analyzed. Based on the behavior of each particle, the factors which affect the convergence and the convergence rate are discussed. Furthermore, an improved particle swarm optimization algorithm is proposed based on the new velocity updating equation. The new algorithm is applied to some benchmark problems, the numerical experiments show that the new algorithm has better performance than the standard PSO and PSO with inertia weight.**

***Keywords*—convergence precision, particle trajectory, convergence rate, particle swarm, social weight**

I. INTRODUCTION

Particle swarm optimization (PSO) is an evolutionary computation technique inspired by natural swarm behavior of searching food [1]. The PSO algorithm is first introduced by Eberhart and Kennedy [2]. In PSO, a member in the swarm, called a particle, represents a potential solution which is a point in the search space. Each particle moves with an adaptive velocity and stores the best position for the search space that has ever visited. The particle adjusts trajectory towards its own previous best position and the best previous position attained by any member of the whole swarm. The PSO algorithm performs well on many optimization problems. However, PSO is easy to relapse into local optimum in solving complex optimization problems and shows the slow convergence velocity in the late stage. In [3], Eberhart and Shi adopt an inertia weight in PSO algorithm. PSO with inertia weight shows superior performance on the convergence velocity. But it is also easy to run into local optimum. A PSO with a constriction factor is introduced by Clerc and Kennedy [4]. Some researchers investigated hybrid algorithm by combing PSO with other search techniques such as mutation, crossover, and so on [5]-[10]. With the accumulation of experience, PSO has been successfully applied to many optimization problems in engineering [11]-[13].

This paper is organized as follows. Firstly, section Ⅱ introduces the standard PSO and the PSO with inertia weight. Then, an improved PSO algorithm is proposed based on the analysis of particle trajectory in section Ⅲ. In section Ⅳ some numerical experiments are conducted. Finally, the discussion and conclusion are given in section Ⅴ.

II. PARTICLE SWARM OPTIMIZATION ALGORITHM

Each particle in swarm has its own velocity and position in different time. D is the dimension of the search space. The formula of updating position and velocity are described as follows:

$$\begin{aligned} v_{t+1} = v_t + c_1 * rand1 * (pbest_t - x_t) \\ + c_2 * rand2 * (gbest_t - x_t) \end{aligned}, \tag{1}$$

$$x_{t+1} = x_t + v_{t+1}, \tag{2}$$

where $v_t = \{v_{t1}, v_{t2}, \cdots, v_{tD}\}$ and $x_t = \{x_{t1}, x_{t2}, \cdots, x_{tD}\}$ are the velocity and position of the ith particle at time t; $pbest_t$ is the best previous position that the ith particle has ever visited; $gbest_t$ is the best position discovered by the whole position. c_1 and c_2 are the cognitive parameter and social parameter respectively, both are positive number. $rand1$ and $rand2$ are random number in the range $[0,1]$. The velocity equation has three elements: the current velocity of particle, the recognition ability which presents itself experience of the particle and the social part which shows the ability of information sharing in swarm.

Since the PSO has been put forward, some researchers commit to improving the performance of PSO algorithm. In [3], Eberhart and Shi introduce inertia weight ω into (1), the velocity formula is updated as

$$\begin{aligned} v_{t+1} = \omega v_t + c_1 * rand1 * (pbest_t - x_t) \\ + c_2 * rand2 * (gbest_t - x_t) \end{aligned}. \tag{3}$$

The inertia weight ω is used to balance the global and local search abilities. A large inertia weight is more appropriate for global search, and a small inertia weight facilitates local search. In [3], ω decreases linearly from 0.9 to 0.4.

The PSO is prone to fall into the local optimum position with too small velocity and probably skips over the global best position with a too large velocity. Thus, the velocity of particle is limited between a low bound $v_{\min}$ and an upper bound $v_{\max}$ in some PSO algorithms.

III. Particle Trajectory Analysis and Improved PSO Algorithm

Eberhart and Shi propose an improved method in which inertia weight is gradually reduced during evolution, but the theory analysis of the decreasing inertia weight is not mentioned. In this section, the particle trajectory and convergence about PSO are discussed.

Let $\varphi_1 = c_1 * rand1$, $\varphi_2 = c_2 * rand2$, $\varphi = \varphi_1 + \varphi_2$ and $p = (\varphi_1 pbest_t + \varphi_2 gbest_t)/(\varphi_1 + \varphi_2)$. Equations (2) and (3) can be translated into

$$v_{t+1} = \omega v_t + \varphi(p - x_t)\,. \tag{4}$$

By using (4), (2) can be changed to

$$x_{t+1} = x_t + \omega v_t + \varphi(p - x_t)\,. \tag{5}$$

Then, let $y_t = p - x_t$. Equations (4) and (5) can be transformed into

$$v_{t+1} = \omega v_t + \varphi y_t\,,$$

$$y_{t+1} = -\omega v_t + (1-\varphi) y_t\,. \tag{6}$$

So, (6) can be described in matrix form

$$H_{t+1} = B \cdot H_t\,, \tag{7}$$

where $H_t = \begin{bmatrix} v_t \\ x_t \end{bmatrix}$, and $B = \begin{bmatrix} \omega & \varphi \\ -\omega & 1-\varphi \end{bmatrix}$.

Suppose that we know the *i*th particle's initial state H_0 . Through iteration method, (7) can be deduced to

$$H_{t+1} = B^{t+1} H_0\,. \tag{8}$$

A. PSO Convergence Analysis

Firstly, the definition of the sequence convergence in mathematics should be discussed.

Definition 1: Let $\{a_n\}$ be a sequence and let A be a fixed number. If for any given $\varepsilon > 0$, there is a positive integer N such that the inequality $|a_n - A| < \varepsilon$ is true for all $n > N$, then we say the sequence has the limit A , which is denoted as $\lim_{n\to\infty} a_n = A$. The sequence is also said to converge to A .

Theorem 1: A $n \times n$ matrix X is diagonalizable if and only if X has n distinct eigenvalues.

Suppose that e_1 and e_2 are the distinct eigenvalues of B . There exists a matrix $Y_{2\times 2}$ which make the flowing formula true.

$$YBY^{-1} = L = \begin{bmatrix} e_1 & 0 \\ 0 & e_2 \end{bmatrix} \tag{9}$$

Let $S_t = YH_t$, (7) and (8) can be changed to

$$S_{t+1} = L \cdot S_t\,, \tag{10}$$

$$S_{t+1} = L^{t+1} \cdot S_0\,. \tag{11}$$

From the definition 1 and theorem 1, theorem 2 is deduced.

Theorem 2: $\lim_{t\to\infty} S_t = 0$ iff $\|e_i\| < 1$ for all eigenvalues (e_i) of $B = \begin{bmatrix} \omega & \varphi \\ -\omega & 1-\varphi \end{bmatrix}$.

The characteristics of matrix B are described as follows.

1) The determinant value of matrix B is equal to ω , $jetB = \omega$.

2) The eigenvalues of B are $e_{1,2} = \dfrac{(\omega + 1 - \varphi) \pm \sqrt{\Delta}}{2}$, where $\Delta = (\omega + 1 - \varphi)^2 - 4\omega$.

a) If $\Delta \ge 0$ ($|\omega + 1 - \varphi| \ge 2\sqrt{w}$), the eigenvalues of B are two real numbers.

b) If $\Delta < 0$ ($|\omega + 1 - \varphi| < 2\sqrt{w}$), the eigenvalues of B are two complex numbers.

3) The product of eigenvalues is equal to ω , i.e. $e_1 \cdot e_2 = \omega$.

B. PSO Convergence Rate

In PSO algorithm, the eigenvalues $e_{1,2}$ have great influence on convergence rate of algorithm.

Definition 2: Let $A = \{x_k\}$ and $B = \{y_k\}$ be sequence with limit ς and ξ respectively. If $\lim_{n\to\infty} (|y_n - \xi|)/(|x_n - \varsigma|) = 0$, that is said that B converges faster than A .

From the above preliminary discussion, theorem 2 is deduced.

Theorem 2: The PSO algorithm converges faster when the eigenvalue $\|e_i\| \to 0$.

Proof: Suppose that $S'_{t+1} = L'^{t+1} \cdot S_0$ and $S_{t+1} = L^{t+1} \cdot S_0$. So

$$(S'_{t+1}-0)/(S_{t+1}-0)=L'^{n+1}/L^{n+1}=\begin{pmatrix} e_1'^{n} & 0 \\ 0 & e_2'^{n} \end{pmatrix}\cdot\begin{pmatrix} e_1^{n} & 0 \\ 0 & e_2^{n} \end{pmatrix}^{-1} = \begin{pmatrix} (e_1'/e_1)^n & 0 \\ 0 & (e_2'/e_2)^n \end{pmatrix}. \quad (12)$$

From definition 2 we know S'_{t+1} converges faster than S_{t+1} iff $\|e_i'/e_i\|<1$. That is $\|e_i'\|<\|e_i\|$.

Theorem 3: When $\omega \to 0$, the PSO gets higher convergence rate.

Proof: From theorem 2 we know the PSO algorithm converge fast if e tends to 0. If $(\omega+1-\varphi)^2 \to \Delta$, $e_{1,2} \to 0$. That is $(\omega+1-\varphi)^2 \to (\omega+1-\varphi)^2 - 4\omega$. We can get $\omega \to 0$.

The inertia weight ω decreases linearly from a large value (0.9) to a small one (0.4) during the PSO run. At the beginning of the run, PSO has more globally search ability because of the large inertia weight, whereas at the end of the process, it has more local search ability and has high efficiency in convergence because of the small inertia weight. From above discussion, we proposed an improving velocity formula,

$$\begin{aligned} v_{t+1} &= \omega v_t + c_1 * rand1 * (pbest_t - x_t) \\ &+ c_2 * (1-\omega/2) * (gbest_t - x_t) \end{aligned} \quad (13)$$

In (13), we bring a new social weight $(1-\omega/2)$ to the social part. When the inertia weight decreases, the social weight increases. The small social weight makes the global best position $gbest_t$ have minor impact on the velocity updating. So at the beginning run, PSO with large inertia weight and small social weight is conductive to global search. At the end of the run, the large social weight makes the best position's information have a great effect on the swarm search behavior. So PSO with small inertia weight and large social weight facilitates information sharing and improves the convergence rate at the end of the run.

The structure of the PSO with social weight (SWPSO) is described as follows.

Begin

Initialize particle swarm

{ randomly generate the position (x_i) of each particle and the associated velocity (v_i);

$pbest_i$ is equal to x_i for each particle i;

calculate the best position of the swarm ($gbest$);

}

while not {stopping criterion} do

{ update the velocity of each particle by using (13);

update the position of each particle by using (2), and limit the position between $[x_{\min}, x_{\max}]$;

calculate the $pbest_i$ for each particle i;

calculate the $gbest$ of the swarm;

}

End

IV. Experiment Result and Discussion

A. Test Function

In order to demonstrate the performance of the new algorithm SWPSO proposed in this paper, the numerical experiments are conducted on the following five benchmark functions.

1) Sphere function

$$f_1(x) = \sum_{i=1}^{n} x_i^2 . \quad (14)$$

2) Rosenbrock function

$$f_2(x) = \sum_{i=1}^{100} (100(x_{i+1}-x_i)^2 + (x_i - 1)^2) . \quad (15)$$

3) Rastrigrin function

$$f_3(x) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10) . \quad (16)$$

4) Greiwank function

$$f_4(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1 . \quad (17)$$

5) Schaffer function

$$f_5(x) = 0.5 - \frac{(\sin\sqrt{x^2+y^2})^2 - 0.5}{(1.0+0.001(x^2+y^2))^2} . \quad (18)$$

For the above five functions, the dimension, search range and the optimal value are listed in table I.

TABLE I. The Parameters Used in Benchmark Function

Function Name	**Dimension** n	**Range** $[x_{min}, x_{max}]$	**Optimal Value**
Sphere function f_1	30	[-100,100]	0
Rosenbrock function f_2	30	[-2.048,2.048]	0
Rastrigrin function f_3	30	[-5.12,5.12]	0
Griewank function f_4	30	[-600,600]	0
Schaffer function f_5	2	[-100,100]	1

B. *Parameter Settings and Experimental Results*

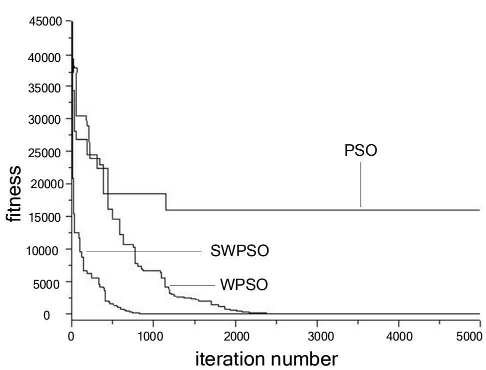
We use two different methods to evaluate the algorithm's performance: 1) assessing the convergence precision and the convergence rate with the fixed generation of evolution; 2) assessing the generation of evolution for achieving the precision which is set ahead.

1) The convergence precision and rate analysis with fixed generation

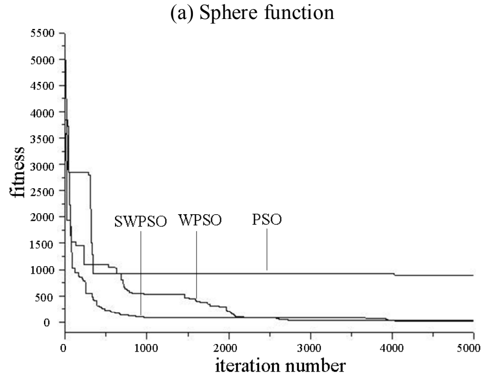
These experiments' goal is to compare three PSO algorithms including the SWPSO, PSO with inertia weight (WPSO)[3] and standard PSO[2]. The population size is set to 30 and the maximum number of iterations is set to 5000. The cognitive parameter c_1 and social parameter c_2 are set to 2. The inertia weight ω deceases linearly from 0.9 to 0.4. In all experiments, the algorithms are run 30 times, the mean values of optimal solutions for 30 runs are presented in table Ⅱ. Figure 1 shows the fitness evolutionary curves of five Benchmark functions for three algorithms.

TABLE II. THE MEAN VALUE IN FIVE BENCHMARK FUNCTIONS

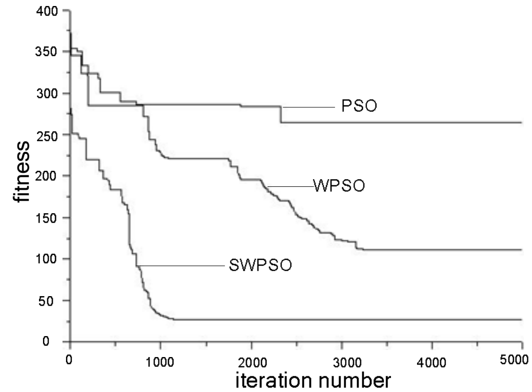
Function Name	PSO	WPSO	SWPSO
Sphere function f_1	15948.073267	0.000000	0.000000
Rosenbrock function f_2	888.126245	57.282964	12.1022187
Rastrigrin function f_3	264.111239	57.7570860	27.271351
Griewank function f_4	141.592531	0.028655	0.006781
Schaffer function f_5	1.000000	1.000000	1.000000



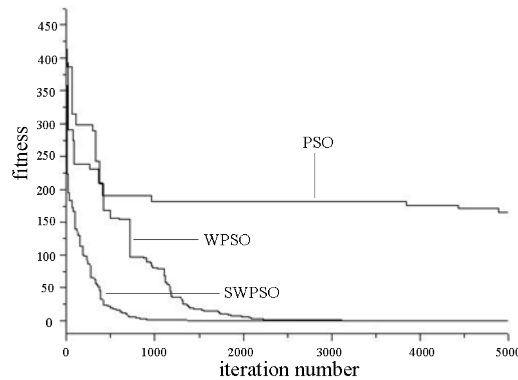
(a) Sphere function



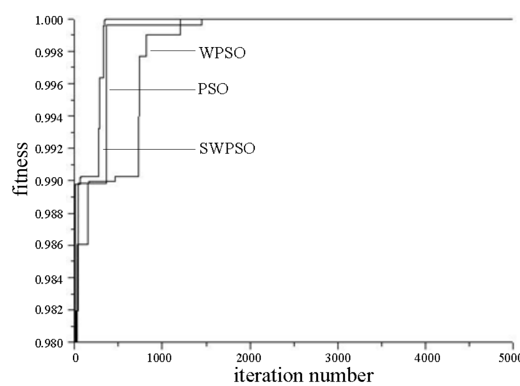
(b) Rosenbrock function



(c) Rastrigrin function



(d) Greiwank function



(e) Schaffer function

Fig 1 The convergence characteristics of test functions

By analyzing the values in table Ⅱ and the convergence curves in fig 1 , we can conclude that the SWPSO has the best performance on the convergence rate and convergence precision among the PSO, WPSO and SWPSO for the above five functions.

2) The convergence rate analysis with fixed accuracy

In these experiments, the population size is set to 30 and the maximum number of iterations is set to 10,000. When the required accuracy is achieved the optimization process is terminated. The required precisions about Benchmark functions are listed in table Ⅲ.In table Ⅲ, f^* is the optimal value, f is the experimental value. The parameters are same as the experiment in 1). Each approach (PSO, WPSO, EPSO) runs 20 times for all test functions. Table Ⅳ lists the average, maximum and minimum number of iterations needed for each approach and gives the successful rate(SR), where *SR=the number of run in which precision is achieved/the total number of run.*

TABLE III. THE REQUIRED ACCURACY OF THE FIXED ACCURACY EXPERIMENTS

Function Name	**Required Accuracy**
Sphere function f_1	$\lvert f-f^*\rvert < 0.01$
Rosenbrock function f_2	$\lvert f-f^*\rvert < 100$
Rastrigrin function f_3	$\lvert f-f^*\rvert < 125$
Griewank function f_4	$\lvert f-f^*\rvert < 0.05$
Schaffer function f_5	$\lvert f-f^*\rvert < 0.0001$

TABLE IV. THE RESULTS FOR THE FIXED ACCURACY EXPERIMENTS

Benchmark Code		**Iterative Times**			***SR***
		Mean	***Min***	***Max***	
f_1	PSO	10000	10000	10000	0%
	WPSO	5375	5265	5517	100%
	SWPSO	1221	1192	1287	100%
f_2	PSO	10000	10000	10000	0%
	WPSO	4500	3369	10000	90%
	SWPSO	765	591	784	100%
f_3	PSO	10000	10000	10000	0%
	WPSO	4364	3732	4931	100%
	SWPSO	933.5	765	1102	100%
f_4	PSO	10000	10000	10000	0%
	WPSO	7703	6639	10000	90%
	SWPSO	2087	1746	2375	100%
f_5	PSO	1038	1038	1038	100%
	WPSO	897	50	1260	100%
	SWPSO	216	216	216	100%

From table Ⅳ, it can be seen that SWPSO achieves the target precision for all five Benchmark functions, but PSO obtains the target precision only for Schaffer function. SWPSO uses the least iteration times for the same goal precision and the same function among PSO, WPSO and SWPSO.

V. CONLUSIONS

From the individual particle's point of view, this paper studies the convergence and the convergence rate of PSO. Through research work, we discover the coefficients in velocity equation play an important role in convergence. Based on the analysis of the particle trajectory, an improved velocity updating method is proposed. In this new method, a social weight is employed into velocity equation. Social weight determines the impact of the global best position. PSO with large inertia weight and small social weight has a strong global search capability. When inertia weight is small and social weight is large, PSO facilitates local search and is conductive to convergence.

Finally, the PSO with social weight is tested. In the experiments with fixed generation, SWPSO obtain remarkably higher precision than PSO and WPSO. Under the fixed target precision, SWPSO greatly reduces the iteration number. By comparing the results and the convergence graphs among these PSO algorithms, SWPSO has good search ability and converges fast. SWPSO also has good performance for balancing the global and local search.

REFERENCES

[1] Z. Michalewicz, Genetic Algortihm + Data structure = Evolution Programs, Springer-Verlag, Berlin: Heidelberg, 1996

[2] J. Kennedy, and R. Eberthart, "Particle Swarm optimization", Proceedings of the IEEE Congress on Neural Networks, 1995, pp.1942-1948

[3] R. Eberhart, and Y. Shi. "Comparing inertia weights and constriction Factor in particle swarm optimization". Proceedings of the IEEE Congress on Evolutionary Computation, San Diego, CA, 2001, pp. 84-88

[4] M.Clerc and J.Kennedy, "The particle swarm – explosion, stability, and convergence in a multidimensional complex space", IEEE Tans. Evol. Comput., vol. 6. Feb, 2002, pp.58–73.

[5] P. J. Angeline, "Using selection to improve particle swarm optimization", Proceedings of the IEEE Congress on Evolutionary Computation, Anchorage, AK, 1998, pp.84-89

[6] M. Lovbjerg, T. K. Rasmussen, and T. Krink, "Hybrid particle swarm, optimizer with breeding ans subpopulations," Proceedings of the IEEE Congress on Evolutionary Computation, 2001, pp.469-471

[7] X. Xie, W. Zhang, and Z. Yang, "A dissipative particle swarm optimizatopn," Proceedings of the IEEE Congress on Evolutionary Computation, 2002, pp.1456-1461

[8] F.Bergh and A.Engelbrecht, "A cooperative approach to particle swarm optimiation", IEEE Tans. Evol. Comput., vol. 8. Jun, 2004, pp.225–239.

[9] T. Krink, J. S. Vesterstroem, and J. Riget, "Particle swarm optimization with spatial particle extension," Proceedings of the IEEE Congress on Evolutionary Computation, Honolulu, HI, 2002, pp.1474-1479

[10] K. E. Parsopoulos and M. N. Vrahatis, "On the computation of all global minimizers through particle swarm optimization", IEEE Tans. Evol. Comput., vol. 8. Jun, 2004, pp.211–224.

[11] M. Wachowiak, R. Smolikova, Y. Zheng, J. Zurada and A. Elmaghraby, "An approach to multimodal biomedical image registration utilizing particle swarm optimiztion", IEEE Tans. Evol. Comput., vol. 8. Jun, 2004, pp.289–301.

[12] L. Messerschmidt and A. Engelbrecht, "Learning to play games using a PSO-based competitve learning approach", IEEE Tans. Evol. Comput., vol. 8. Jun, 2004, pp.280–288.

[13] Y. D. Vall, S. Mohagheghi and R. G. Harley, "Particle swarm optimization: basic concepts, variants and applications in Power Systems", IEEE Tans. Evol. Comput., vol. 12. Apr, 2008, pp.171–195.