A Novel PSO with Piecewise-Varied Inertia Weight

Qin Quande
School of Business Administration
South University of Technology
Guangzhou, PR China
qinquande@gmail.com

Li Li
College of Management
Shenzhen University
Shenzhen, PR China
llii318@163.com

Li Rongjun
School of Business Administration
South University of Technology
Guangzhou, PR China
lirongjun1016@hotmail.com

***Abstract*—To choose the appropriate value of inertia weight can improve the performance of PSO by means of making a good balance between exploration and exploitation in search process. This paper presents a novel inertia weight variation method based on a piecewise function，in which there are two parts: one is nonlinear decreasing to enhance the explorative ability; the other is linear decreasing just as standard PSO algorithm. The two key parameters in the proposed PSO algorithm (PSO-PIW) are identified through two experimental simulations. The results of several benchmark functions tested demonstrate good performance of PSO-PIW in solving multimodal problems compared with some other inertia weight variation methods in PSO.**

Keywords-Particle swarm optimization; Inertia weight; Piecewise function; PSO-PIW; Multimodal problems

I. INTRODUCTION

Particle swarm optimization (PSO) is a kind of stochastic optimization technique, motivated from the social behaviors of animals such as fish schooling and bird flocking [1,2]. In original PSO algorithm, the trajectory of each particle in the search space is adjusted by dynamically altering its velocity, according to its own experience and the experience of its informants [3].

Comparing with other stochastic optimization methods, PSO has comparable or superior search performance for many hard optimization problems with faster convergence rate. It has already been widely used as a problem-solving method in many areas. However, Angeline [4] shows that the PSO has difficulties in keeping balance between exploration and exploitation. This indicates that PSO algorithm is prone to lose the diversity. Many researches have pointed out the defects of PSO and provided some improved methods, for instance, tuning the parameters in the velocity and position update equations of PSO [5,6,7], designing different swarm topologies [8,9], combining PSO with other evolutionary optimization operators [10], incorporating bio-inspired mechanisms into the standard PSO [11] and adopting new learning strategies[12,13].

It is generally recognized that how to set the value of inertia weight (IW) counts much for the performance of PSO algorithm. In this paper, we present here a new approach to improve the performance after a predefined number of generations in PSO, by means of proposing a novel variation strategy for IW based on piecewise function. There are two parts in the piecewise function to set the variation of IW, one is nonlinear decreasing in order to enhance the explorative ability of PSO, and the other is linear decreasing as the same as standard PSO.

II. BASIC PSO AND ITS VARIANTS

The original PSO defines each particle as a potential solution of the problem in a D-dimension search space with a velocity which is dynamically adjusted according to its own experience and that of other particles. The position vector of the i_{th} particle is represented as $x_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$ in the D-dimensional space, where $x_{id} \in [l_d, u_d]$, $d = 1,2,\cdots D$, l_d, u_d are the lower and upper bounds of the d_{th} dimension, respectively. The velocity vector for i_{th} particle is represented as $v_i = (v_{i1}, v_{i2}, \ldots, v_{iD})$, which is clamped to a maximum velocity, $V_{\max}$. Each particle remembers its own previous best position, $pBest$, so far. With regard to the i_{th} particle, we denote as $pBest_i = (pbest_i^1, pbest_i^2, \ldots, pbest_i^D)$. $gBest$ is the swarm previous best position. During the iteration time, the i_{th} particle is manipulated according to the following equations:

$$v_i(t+1) = \omega v_i(t) + \varphi_1 c_1 (pBest_i - x_i(t)) + \varphi_2 c_2 (gBest - x_i(t)) \quad (1)$$

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

where φ_1 and φ_2 are random numbers, uniformly distributed with the interval $[0,1]$. c_1 and c_2 , usually set to 2.0, which determine the balance between the influence of the individual's knowledge and that of the swarm, called acceleration coefficients. ω , called inertia weight, is to assist with the balance between exploration and exploitation. The value of ω is initially set to 0.9, reducing linearly to 0.4 during an optimization run [14]. Hereafter, in this paper, this version of PSO is referred to as standard PSO (SPSO).

Inspired by Clerc's constriction factor concept [7], Eberhart and Shi [15] proposed a random IW factor, which is set to change according to the following equation:

$$\omega = 0.5 + rand()/2 \quad (3)$$

where rand() is a uniformly distributed random number within range [0,1].In the remainder of this paper, this method is referred to as random weight method (PSO-RIW).

Chatterjee and Siarry [6] proposed a nonlinear function modulated IW adaptation for improved performance of PSO

978-1-4244-6928-4/10/$26.00 ©2010 IEEE

algorithm (PSO-NLIW). In this development, the IW of current iteration is calculated based on the following equation:

$$\omega = \left((iter_{\max} - iter)^n / (iter_{\max})^n \right) \times (\omega_{start} - \omega_{end}) + \omega_{end} \tag{4}$$

where n is the nonlinear modulation index and the implication of other parameters are the same as the Eq. (4).

III. PROPOSED METHOD

It is universally acknowledged that a larger IW facilitates global exploration and a smaller one tends to facilitate local exploration to fine-tune the current search area [5]. Accordingly, we firstly want to develop a method, in which the value of IW is higher in early iterations and lower in later iterations than SPSO, to modulate the variation of IW. The segments of (A+C) in Figure 1 deliver our initial idea. Through some experimental simulations, the performance is below expectation because the IW decreases rapidly between early and later iterations, which may lead to the excessive oscillations. Consequently, the linear decreasing method (described as B in Figure 1) is used to substitute for as the segment of C in Figure 1. As a result, the piecewise-varied IW is to follow a piecewise function, which is described as (A+B) in Figure 1. The proposed model is called as PSO-PIW in the paper.

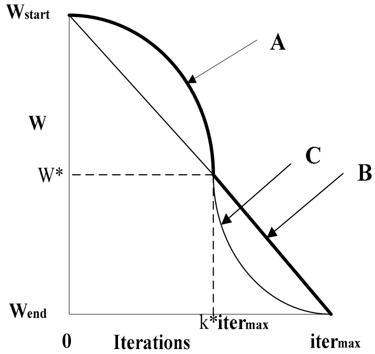

Figure1. Proposed methods of inertia weight variation

Tangent function, $y = \tan(x)$, is periodic function and in every period the result y increases nonlinearly along with the independent variable x. In this paper we attempt to use tangent function to simulate the shape of the segment of A. in Figure 1. Now the IW, ω, is set to change according to the following the piecewise equations:

$$\omega = \begin{cases} \tan\left(\pi/4 \times \left(1 - iter/(k \times iter_{\max})\right)^a \right) \times (\omega_{start} - \omega^*) + \omega^* & iter \le (k \times iter_{\max}) \quad (5) \\ (\omega_{start} - \omega_{end}) \times (iter_{\max} - iter)/iter_{\max} + \omega_{end} & iter > (k \times iter_{\max}) \quad (6) \end{cases}$$

where a is the nonlinear modulation index, $iter_{\max}$ is the maximum number of allowable iterations, and $iter$ is the current iteration number. ω_{start} and ω_{end} denote the initial and final values of the IW, respectively; k is a constant denotes the percent of $iter_{\max}$ to adjust the IW variation based on Eq. (5), ω^* is the corresponding IW of the iterations of $k \times iter_{\max}$ numbers. The constant, $\pi/4$, in the Eq. (5) is to guarantee the ω distributed within range $[\omega_{start}, \omega_{end}]$.

IV. PARAMETER ANALYSIS OF PSO-PIW

A. Benchmarks and experimental setting

A set of well-known benchmarks, which are commonly adopted to test the performance of an algorithm in evolutionary optimization, are used to select the proper parameters of PSO-PIW in terms of solution quality. All benchmarks used and their parameters setting are given in Table1

Table 1 Benchmarks for simulation and parameter setting

Function	Mathematical representation	Range of search	Range of initialization
Sphere	$f_1(x) = \sum_{i=1}^{n} x_i^2$	$(-100,100)^n$	$(50,100)^n$
Rosenbrock	$f_2(x) = \sum_{i=1}^{n} (100(x_{i+1} - x_i)^2 + (x_i - 1)^2)$	$(-30,30)^n$	$(15,30)^n$
Rastrigin	$f_3(x) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10)$	$(-5.12,5.12)^n$	$(2.56,5.12)^n$
Griewank	$f_4(x) = (1/4000)\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(x_i/\sqrt{i}) + 1$	$(-600,600)^n$.	$(300,600)^n$

f_1 and f_2 are both unimodal functions. f_1 is easy to solve and it is hard to find the optimal solution of f_2 as it has a narrow valley from the perceived local optima to the global optimum; therefore, it can be treated as a multimodal problem. Compared with the first two functions, f_3 and f_4 are multimodal functions, which add the difficulty in solving. f_3 is a complex multimodal problem with a large number of local optima. It is easy to fall into a local optimum in attempting to solve the problem. An interesting phenomenon of f_4 is that it is more difficult for lower dimensions than higher dimensions [16].

Table1 lists the ranges of their search, range of initialization and mathematical representation of benchmarks used. This paper follows Eberhart and Shi's suggestion [17] that it is good to limit the maximum velocity $V_{\max}$ to the upper value of the range of search space $X_{\max}$, that is $V_{\max} = X_{\max}$. It should be noted that asymmetric initialization method introduced by Angeline [18] is adopted here, in which the population is initialized only in a portion of the search space. This is to prevent a center-seeking optimizer from accidentally finding the global optimum as the benchmarks used in this paper at the origin of the search space. All experiments carried out with a population size of 40 as van den Bergh [19] suggested that there is a slight improvement of optimal value with increasing swarm size.

B. Parameter analysis

From Eq. (5), it can be seen that the value of a is related to the smoothness of the curve and the performance of PSO-PIW. In order to identify the best value of a, Eq. (5) is used to modulate the variation of IW for the entire search process, which means that $k=1$, $\omega^*=\omega_{end}$ in Eq. (5). Therefore, IW changes into the following equation:

$$\omega=\tan\left((\pi/4)\times(1-iter/iter_{end})^{a}\right)\times(\omega_{start}-\omega_{end})+\omega_{end} \qquad (7)$$

The values of ω_{start} and ω_{end} are set in accordance with SPSO. On the assumption that $iter_{\max}$ equates 4,000, the graph of Eq. (7) is plotted with the different values of a as described in Figure 2.

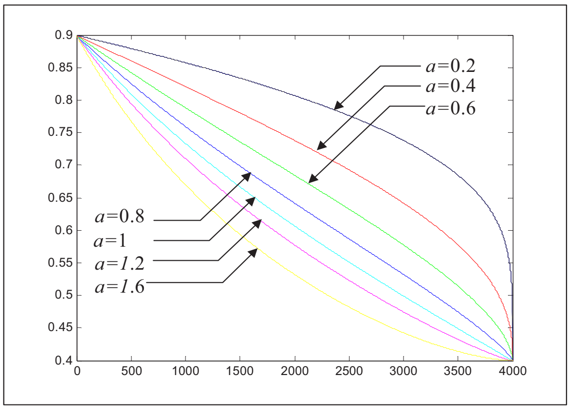

Figure 2 Variations of IW for different values of a

It can be seen from Figure 2 t when a is equal to 0.2, 0.4, 0.6 and 0.8, respectively, the shape of the graph, which conforms to the requirements in view of the analysis of Section 3, is convex, while a is equal to 1, 1.2 and 1.6, the shape of the graph is concave.

Rastrigin is employed to conduct experiment to choose the best value of a and k in Eq. (5). The value of a is confined within range [0.2, 2] in the experiment and choose the typical value of a: 0.2, 0.6, 0.8, 1, 1.2 and 2. In the experiment, the parameters are set as follows: the dimension of problem, n, is set to 20; c_1 and c_2 are both set to 2.0; IW varied based on Eq. (7). A total of 30 runs for each value of a are conducted. The experimental results (i.e., the mean and the standard deviations) of the Rastrigin are listed in Table 2.

Table 2 Experimental results of f_3 based on different value of a

value of a	Indicators	value of a	Indicators	value of a	Indicators
a =0.2	19.7992 (6.8487)	a =0.6	10.093 (3.3786)	a =0.8	10.3675 (3.7918)
a =1	10.2083 (3.8139)	a =1.2	11.4221 (3.9246)	a =2	12.6161 (3.8895)

In Table 2, it is obvious that when a =0.6, both the mean and standard deviation are the minimum and with the increasing or decreasing the value of a, the performance of the function is getting worse. Furthermore, some experiments are also conducted with the value of a that is near 0.6 and the results demonstrated that a is equal to 0.6 is a good choice. Herein, we have identified the best value of a.

Table3 Experimental results of f_3 based on different value of k

value of k	Indicators	value of k	Indicators	value of k	Indicators
k=0.3	11.7745 (3.9154)	k=0.4	10.3628 (3.8878)	k=0.5	11.0382 (2.4149)
k=0.6	9.3628 (3.8878)	k=0.7	7.7857 (2.8335)	k=0.8	8.9745 (2.9302)

Table 3 shows the performance evaluation of Rastrigin function with different value of k with the interval [0.3, 0.8]. All parameters are set to as same as that in the former experiments except the IW, which is set to change according to Eq. (5) and (6). It can be seen that the best value of k is 0.7 from Table 3.

. So far, the two key parameters in Eq. (5) are identified. The IW of PSO-PIW is adjusted based on Eq. (8) and (9).

$$\omega=\begin{cases}\tan\left(\pi/4\times(1-iter/(0.7\times iter_{\max}))^{0.6}\right)\times(\omega_{start}-\omega^*)+\omega^* & iter\le(0.7\times iter_{\max}) \quad (8)\\ (\omega_{start}-\omega_{end})\times((iter_{\max}-iter)/iter_{\max})+\omega_{end} & iter>(0.7\times iter_{\max}) \quad (9)\end{cases}$$

V. EXPERIMENTAL STUDIES

The performance of PSO-PIW is compared with SPSO, PSO-RIW as well as PSO-NLIW. The parameters used for SPSO and PSO-RIW were recommended in [15, 17]. The acceleration coefficients, c_1 and c_2 for SPSO, PSO-RIW PSO-NLIW and PSO-PIW were both set to 2.0. For PSO-RIW, $c_1=c_2=1.494$ was used. In SPSO, PSO-NLIW and PSO-PIW, a decreasing IW starting at 0.9 and ending at 0.4, that is $\omega_{start}=0.9$, $\omega_{end}=0.4$. All experiments were run 50 times. The best, worst fitness, the mean values and standard deviation of the results are presented in the Table 4, in which numbers in bold represent the comparatively best values.

From the Table 4 we can easily discover that PSO-PIW performs well for multimodal problems as it has good global search ability. However, there is no distinct evidence that PSO-PIW performs better than other PSO algorithms in solving unimodal functions.

Table 4 Experimental results of the optimal value for 50 runs

	Dimension	$iter_{max}$	Indicators	Categories of PSO compared			
				SPSO	PSO-RIW	PSO-NLIW	PSO-PIW
f_1	30	3000	Best	1.1614e-020	**1.0064e-046**	9.9199e-023	6.2725e-022
			Worst	1.1886e-016	**1.0106e-036**	1.6865e-020	6.8278e-020
			Mean	1.5731e-017	**2.0239e-040**	1.1519e-020	4.4119e-021
			Std	2.2292e-017	**1.4291e-040**	3.3865e-020	1.2657e-020
f_2	30	4000	Best	5.5646	**0.5413**	1.3192	3.6752
			Worst	327.4047	165.1063	458.7243	**135.0923**
			Mean	77.5157	**32.4511**	63.9851	35.7807
			Std	89.2552	55.6584	81.1993	**42.6215**
f_3	30	4000	Best	12.9345	52.7327	13.9294	**8.9546**
			Worst	53.7277	218.8899	41.7882	**32.8356**
			Mean	28.8091	118.0415	29.3774	**19.2390**
			Std	9.4425	38.0175	6.9043	**5.7980**
f_4	10	3000	Best	0.0172	0.0156	0.0245	**0**
			Worst	0.1501	0.1746	0.2803	**0.1227**
			Mean	0.0599	0.0723	0.0704	**0.0471**
			Std	0.0322	0.0374	0.0422	**0.0304**

In order to improve the exploration, PSO-PIW has a larger value of IW in the early iterations, which makes slow convergence. According to the “no free lunch” theorem, there is some cost for tuning the PSO-PIW to obtain better performance on multimodal problems, and the cost is the slow convergence.

VI. CONCLUSIONS

In this paper, a novel variation for IW based on piecewise function in PSO is proposed. The new strategy is to improve the performance after a predefined number of generations. PSO-PIW modulates the IW in light of a convex nonlinear function, which makes IW a big value in the early iterations, and is linear decreasing as the same as standard PSO algorithm which is to keep the exploitation. Through some experiments, the two key parameters in PSO-PIW algorithms are identified.

Compared with SPSO, PSO-RIW and PSO-NLIW on four benchmark functions, PSO-PIW is less susceptible to premature convergence and less likely to be stuck into local optima. The preliminary results suggest that PSO-PIW performs well for multimodal problems.

Future work will focus on optimizing the performance of PSO-PIW, for example, combining a local search method such as Quasi-Newton to improve the effectiveness of solving unimodal problems and convergence rate.

REFERENCES

[1]. R.C. Eberchart, J. Kennedy, A new optimizer using particle swarm theory, in：Proceedings of the 6th International Symposium on Micromachine and Human Science, Nagoya, Japan 1995,pp.39-43.

[2]. J. Kennedy, R.C. Eberchart, Particle swarm optimization, in: Proceedings of IEEE International Conference on Neural Networks, Perth, Australia, 1995, pp.1942-1948.

[3]. R.C. Eberchart, Y. Shi, Particle swarm optimization: developments, applications and resources, in: Proceedings of the IEEE Congress on Evolutionary Computation, Piscataway, NJ，2001，pp.81-86.

[4]. P.J. Angeline, Evolutionary optimization versus particle swarm optimization and philosophy and performance difference, in: Proceedings of 7th Annual Conference on Evolutionary Programming, San Diego, USA, 1998, pp.601–610.

[5]. Y. Shi, R.C. Eberchart, A modified particle swarm optimizer, in: Proceedings of the IEEE Congress on Evolutionary Computation, Anchorage, AK, NJ, 1998, pp.69-73.

[6]. A..Chatterjee, P. Siarry, Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization, Comput. Oper. Res. 33 (2006) 859-871.

[7]. M.Clerc, J. Kennedy, The particle swarm: explosion, stability, and convergence in multidimensional complex space. IEEE Trans. Evol. Comput. 6 (2002) 58-73.

[8]. J. Kennedy, Small Worlds and Mega-minds: Effects of Neighborhood Topology on Particle Swarm Performance. In: Proceedings of the Congress on Evolutionary Computation, Piscataway, NJ 1999, pp.1931-1938

[9]. P.N. Suganthan, Particle swarm optimizer with neighborhood operator, in: Proceedings of IEEE International Congress on Evolutionary Computation, Washington, DC, USA, 1999, pp.1958–1962.

[10]. C.F. Juang, A Hybrid of Genetic Algorithm and Particle Swarm Optimization for Recurrent Network Design. IEEE Trans. Syst. Man. Cyb 34 (2) (2004) 997-1006.

[11]. S.He, Q, H. Wu etc, A particle swarm optimizer with passive congregation, Biosystems 78 (2004) 135-147.

[12]. J. J. Liang, P. N. Suganthan, A. K. Qin, S. Baskar, Comprehensive learning particle swarm Optimizer for global optimization of multimodal functions”, IEEE Trans. Evol. Comput. 10 (2006) 281-295.

[13]. R. Mendes, J. Kennedy, J. Neves, The fully informed particel swarm: Simpler, may be better, IEEE Trans, Evol. Comput. 8 (2004) 204-210.

[14]. Y. Shi, R.C. Eberhart, Parameter selection in particle swarm optimization, Lecture Notes in Computer Science 1447. Springer, Berlin, 1998, pp.591-600.

[15]. R.C. Eberchart, Y. Shi, Tracking and optimizing dynamic systems with particle swarms, in: Proceedings of the IEEE Congress on Evolutionary Computation, Seoul, Korea, 2001, pp.94-97.

[16]. D. Whitley, D. Rana, J. Dzubera, E. Mathias, Evaluating evolutionary algorithms, Artificial. Intelligence. 85, pp.245-276.

[17]. R.C. Eberhart, Y. Shi, Comparing inertia weights and constriction factors in particle swarm optimization, in: Proceedings of IEEE International Congress on Evolutionary Computation, San Diego, CA, 2000, pp. 84–88

[18]. P.J. Angeline, Using selection to improve particle swarm optimization, in: Proceedings of the IEEE International Conference on Computational Intelligence.1998, pp.84-89.

[19]. F.van den Bergh, A.P. Engelbrecht, Effect of swarm size on cooperative particle swarm optimizers, in: Proceedings of Genetic Evolutionary Computation Conference (GECC0), San Franciso, CA, 2001, pp.469-476.