2009 International Joint Conference on Computational Sciences and Optimization

A Particle Swarm Optimizer with Multi-Stage Linearly-Decreasing Inertia Weight

Jianbin Xin [1, 2], Guimin Chen [1], Yubao Hai [1]
[1] School of Mechatronics, Xidian University, Xi'an 710071, China
[2] School of Electrical Engineering, Xi'an Jiaotong University, Xi'an 710049, China
xin_jianbin@126.com; gmchen@mail.xidian.edu.cn

Abstract

The inertia weight is often used to control the global exploration and local exploitation abilities of particle swarm optimizers (PSO). In this paper, a group of strategies with multi-stage linearly-decreasing inertia weight (MLDW) is proposed in order to get better balance between the global and local search. Six most commonly used benchmarks are used to evaluate the MLDW strategies on the performance of PSOs. The results suggest that the PSO with W_5 strategy is a good choice for solving unimodal problems due to its fast convergence speed, and the CLPSO with W_5 strategy is more suitable for solving multimodal problems. Also, W_5-CLPSO can be used as a robust algorithm because it is not sensitive to the complexity of problems for solving.

1. Introduction

The Particle Swarm Optimizer (PSO) is an evolutionary computation technique first introduced by Kennedy and Eberhart in 1995 [1]. Because of its simple mechanism and high performance, PSO has been used in many engineering fields such as neural network training, reactive power compensation [2] and task scheduling [3].

As an important parameter in PSO, the inertia weight is often used to balance global exploration and local exploitation of the searching process. PSO with linearly-decreasing inertia weight (LDW-PSO) [4] was recommended due to its good performance over a large number of optimization problems. Nonlinearly-decreasing strategies of inertia weight also have been proposed and evaluated [5-7]. However, the evaluation of these nonlinear strategies was drawn based on relatively short runs (e.g., 1000 or 2000 iterations) on classic benchmarks. It's should be noted that some algorithms might outperform others by finding a locally best region of a multimodal function in a short run. Furthermore, recent PSO variants, including the Comprehensive Learning Particle Swarm Optimizer (CLPSO) [8] and the Fully Informed Particle Swarm Optimizer (FIPSO) [9], have shown remarkable performance in consistently finding the global optimum. But their superiority always reveals after 3000 or more iterations. Therefore, the effects of nonlinear strategies on PSOs in relatively long runs should be evaluated.

The present work proposes a new group of nonlinear strategies called multi-stage linearly-decreasing inertia weight (MLDW), for the purpose of easily refining the decreasing process of the inertia weight. Six commonly used benchmarks are used to evaluate MLDW for 5000 iterations.

2. Particle Swarm Optimizers

2.1. LDW-PSO

PSO is a population-based algorithm. Each individual in the population is called a particle. In a d-dimension search space, the position vector and the velocity vector of the ith particle can be represented as $X_i = (x_{i1}, x_{i2}, \ldots, x_{id})$ and $V_i = (v_{i1}, v_{i2}, \ldots, v_{id})$ respectively. The best position found by the ith particle so far is denoted as $P_i = (p_{i1}, p_{i2}, \ldots, p_{id})$, and the best position found by the whole swarm as $P_g = (p_{g1}, p_{g2}, \ldots, p_{gd})$. The velocity and position of the ith particle in the kth dimension are updated as follows:

$$v_{ik} = wv_{ik} + c_1R_1()*(p_{ik} - x_{ik}) + c_2R_2()*(p_{gk} - x_{ik}) \quad (1)$$

$$x_{ik} = x_{ik} + v_{ik} \quad (2)$$

where w is the inertia weight, c_1 and c_2 are constants known as acceleration coefficients, which are often fixed at 2, and $R_1()$ and $R_2()$ are two random numbers in the range [0, 1].

In LDW-PSO, w is given as

$$w = (w_s - w_e)(t_{\max} - t)/t_{\max} + w_e \quad (3)$$

where $t_{\max}$ donates the maximum number of allowable iterations, t represents the current iteration times, and w_s and w_e are the initial and final values of the inertia

978-0-7695-3605-7/09 $25.00 © 2009 IEEE
DOI 10.1109/CSO.2009.420

weight, respectively. The performance of LDW-PSO can be improved significantly when w_s=0.9 and w_e=0.4 [4].

2.2. CLPSO

In CLPSO [8], the velocity of particle i in the kth dimension is updated as

$$v_{ik} = wv_{ik} + c_3 R_3() * (p_{F_i(k)k} - x_{ik}) \qquad (4)$$

where $F_i(k)$ defines which particles' p_{ik} should particle i follow in the kth dimension.

3. MLDW strategies

The inertia weight offers PSOs a convenient way to control between exploration and exploitation. In order to find a strategy that can balance the search better between exploration and exploitation over the iterations than LDW does, a group of MLDW is presented, which can be expressed as

$$w = \begin{cases} (w_s - w_m)(t_1 - t)/t_1 + w_m & 0 \le t \le t_1 \\ w_m & t_1 < t \le t_2 \\ (w_m - w_e)(t_{\max} - t)/(t_{\max} - t_2) + w_e & t_2 < t \le t_{\max} \end{cases} \qquad (5)$$

where t_{max}=5000, w_s=0.9, w_e=0.4, and w_m, t_1 and t_2 and the multi-stage parameters The multi-stage parameters of six selected MLDW strategies which are referred to as W_1-W_6, are listed in Table 1. Figure 1 plots the curves of these strategies.

Table 1. Parameters for selected MLDW strategies

	W_1	W_2	W_3	W_4	W_5	W_6
w_m	0.8	0.8	0.65	0.65	0.5	0.5
t_1	1000	2000	1000	2000	1000	2000
t_2	4000	3000	4000	3000	4000	3000

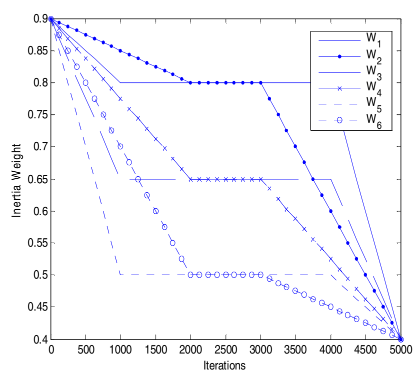

Figure 1. The inertia weight curves over iterations

In the following sections, LDW-PSO is referred to as W_0-PSO while PSO with MLDW as W_j-PSO, and CLPSO with LDW as W_0-CLPSO while CLPSO with MLDW as W_j-CLPSO.

4. Test

4.1. Benchmarks

Six commonly used benchmarks were adopted to evaluate the performance of algorithms.

(1) Sphere function:

$$f_1(x) = \sum_{i=1}^{d} x_i^2$$

(2)Rosenbrock function:

$$f_2(x) = \sum_{i=1}^{d} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right]$$

(3) Rastrigin function:

$$f_3(x) = \sum_{i=1}^{d} [x_i^2 - 10\cos(2\pi x_i) + 10]$$

(4) Griewank function:

$$f_4(x) = \frac{1}{4000} \sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

(5) Schaffer's f6 function:

$$f_5(x) = 0.5 - \frac{(\sin\sqrt{x_1^2 + x_2^2})^2 - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}$$

(6) Ackley function:

$$f_6(x) = -20\exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\right) - \exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)\right) + 20 + e$$

The parameters for each benchmark are given in Table 2.

4.2. Experimental setting

The population size was set at 30 and the symmetric initialization method was used. Each variant of the PSOs was tested 30 times on each benchmark. The optimization results at the 1000th iteration, the 3000th iteration, and the 5000th iteration were recorded, respectively.

4.3. Results

In order to combine the results from different functions, we standardized the results to the same scale using the method proposed in [9]. Because all the benchmarks are minimization problems, negative standardized values are better than average, while positive ones are worse than average. Table 3 and 4 list the standardized results for W_j-PSO and W_j-CLPSO, respectively.

Table 2. Parameters for the benchmarks

Function	Dimensions (d)	Initialization range	Goal
Sphere	30	(-100, 100)	0.01
Rosenbrock	30	(-30, 30)	100
Rastrigin	30	(-5.12,5.12)	100
Griewank	10	(-600, 600)	0.05
Griewank	30	(-600, 600)	0.05
Schaffer's f6	2	(-100, 100)	10^{-5}
Ackley	30	(-30, 30)	0.01

Table 3. Standardized results of W_i-PSO

	1000	3000	5000
W_1-PSO	3.521	8.434	3.977
W_2-PSO	10.398	9.417	-0.474
W_3-PSO	**-5.637**	**-3.87**	**-1.859**
W_4-PSO	0.759	-3.697	-0.884
W_5-PSO	**-8.366**	**3.780**	**-0.377**
W_6-PSO	-3.531	-3.536	-0.736
W_0-PSO	2.856	-2.968	0.354

Table 4. Standardized results of W_i-CLPSO

	1000	3000	5000
W_1-CLPSO	2.627	5.377	11.542
W_2-CLPSO	11.904	12.344	0.326
W_3-CLPSO	**-5.668**	**-3.586**	**-2.455**
W_4-CLPSO	0.03	-3.114	-1.955
W_5-CLPSO	**-7.443**	**-4.233**	**-2.95**
W_6-CLPSO	-4.055	-3.982	-2.788
W_0-CLPSO	2.606	-2.806	-1.721

It can be seen from Table 3 and 4 that W_3 and W_5 (marked in bold) improve PSO and CLPSO either at the 1000th iteration or at the 5000th iteration. Figure 2 shows the mean convergence characteristics of the selected PSO variants, i.e., W_0-PSO, W_3-PSO, W_5-PSO, W_0-CLPSO, W_3-CLPSO and W_5-CLPSO. It can be seen from Figure 2 that both W_3-PSO and W_5-PSO converge faster than W_0-PSO at the early stage for all the benchmarks. W_3 enhances the searching ability of PSO for Rastrigin and Schaffer f6. Compared with W_i-CLPSOs, W_i-PSOs can always find good results at the early stage, but are prone to be trapped in a locally optimal region in long runs. For all benchmarks, W_3-CLPSO and W_5-CLPSO surpass other CLPSO variants both in the convergence speed and search accuracy. This indicates that the inertia weight for CLPSO should be decreased quickly at the early stage. The results partially support the conclusion of Chen *et al* [6] which suggests using concave functions for decreasing strategies.

Table 5 presents the iteration times to the goal on average for five of the algorithms mentioned above. The numbers in bold indicate the corresponding success rate is 100%. We may say the performance of W_5-CLPSO is what we want because not only does it reach the goal in relatively short runs, but also its success rate is high up to 100%.

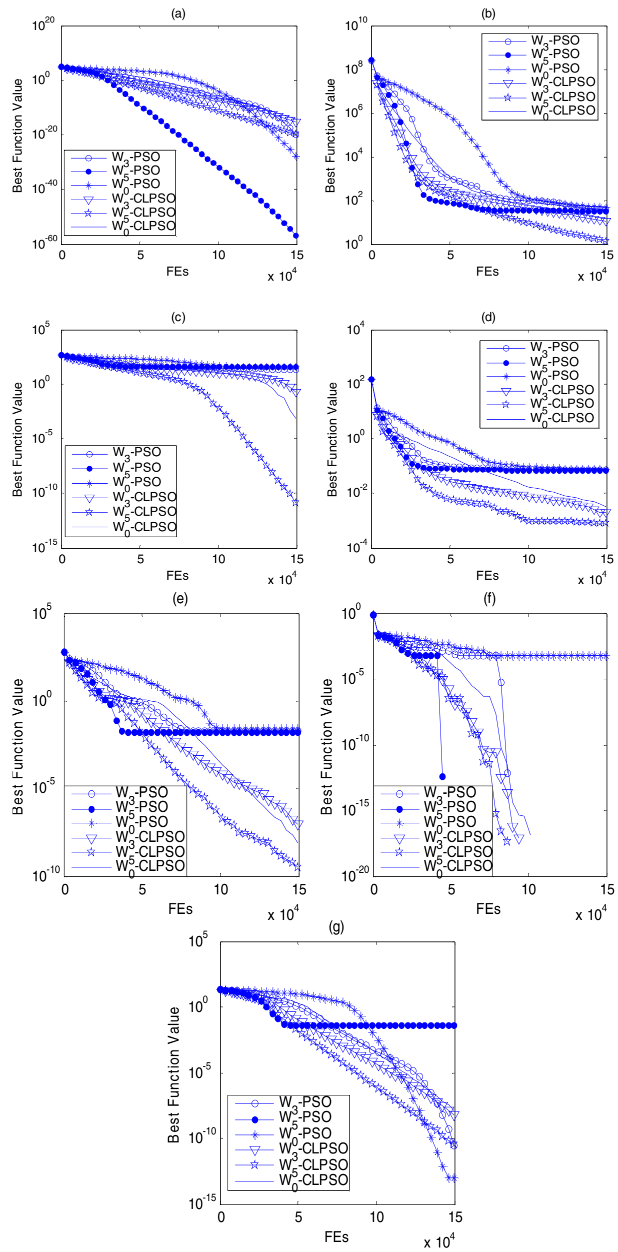

Figure 2.The mean convergence characteristics of six selected PSO variants on the benchmarks. (a) Sphere function. (b) Rosenbrock's function. (c) Rastrigin's function. (d) Griewank's function (10d). (e) Griewank's function (30d). (f) Schaffer's f6 function. (g) Ackley's function.

Generally speaking, W_5-PSO is suitable for solving unimodal problems such as the Sphere function because of its fast convergence speed, whereas W_5-CLPSO can be used to solve multimodal problems.

Table 5.The iteration times to the goal on average

	W_3-PSO	W_5-PSO	W_0-PSO	W_5-CLPSO	W_0-CLPSO
$f_1(x)$	**2187**	**1115**	**3082**	**1489**	**2520**
$f_2(x)$	**2658**	**1364**	3601	**1833**	**3145**
$f_3(x)$	**1173**	**791**	**2607**	**577**	**1228**
$f_4(x)$ (10d)	4004	3991	4670	**915**	**2554**
$f_4(x)$ (30d)	2377	1377	3421	**1494**	**2538**
$f_5(x)$	**822**	**546**	1501	**918**	**1835**
$f_6(x)$	**2535**	1363	**3215**	**1738**	**2688**

5. Conclusion

Motivated by the idea of LDW, the strategy of MLDW is proposed in this paper. The PSO with W_5 strategy is a good choice for solving unimodal problems due to its fast convergence speed. The CLPSO with W_5 strategy features relatively fast convergence speed and high search accuracy and is suitable for solving multimodal problems. Also, W_5-CLPSO can be used as a robust algorithm because it is not sensitive to the complexity of problems for solving.

Acknowledgments

The authors gratefully acknowledge the financial support from the National Natural Science Foundation of China under Grant No. 50805110 and the China Postdoctoral Science Foundation under Grant No. 20070421110. The authors also want to thank Professor P. N. Suganthan for the code he provided.

References

[1] J Kennedy, R Eberhart. *Particle swarm optimization*. International Conference on Neural Networks. Perth, Australia: IEEE, 1995, 1942-1948.

[2] R Eberhart, Y Shi. *Particle swarm optimization: developments, applications and resources*. Proceedings of the IEEE Congress on Evolutionary Computation. Piscataway, USA: IEEE, 2001, 81-86.

[3] A Salman, I Ahmad, S Al-Madani. *Particle swarm optimization for task assignment problem*. Microprocessors and Microsystems, 2002, 26 (8): 363-371.

[4] Y Shi, R Eberhart. *Empirical study of particle swarm optimization*. International Conference on Evolutionary. Computation. Washington, USA, 1999: 1945-1950.

[5] X Zhang, Y Du, G Qin. *Adaptive Particle Swarm Algorithm with Dynamically Changing Inertia Weight (in Chinese)* Journal of Xi'an Jiaotong University, 2005, 39 (19): 1039-1042.

[6] G M Chen, Q Han, J Y Jia. *Study on the Strategy of Decreasing Inertia Weight in Particle Swarm Optimization Algorithm (in Chinese)*. Journal of Xi'an Jiaotong University, 2006, 40 (1): 53-56.

[7] A Chatterjee and P Siarry. *Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization*. Computers & Operations Research, 2006, 33 (3): 859-871.

[8] J J Liang, A K Qin, P N Suganthan, and S Baskar. *Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions*. IEEE Transactions on Evolutionary Computation, 2006, 10(3): 281-295.

[9] R Mendes, J Kennedy, and J Neves. *The fully informed particle swarm: Simpler, maybe better*. IEEE Transactions on Evolutionary Computation, 2004, 8(3): 204-210.