Proceedings of the 7th
World Congress on Intelligent Control and Automation
June 25 - 27, 2008, Chongqing, China

The Inertia Weight Self-Adapting In PSO*

Dong, Chen[1,3]*; Wang, Gaofeng*[2] *and Chen, Zhenyi*[4]
[1]*Computer School, Wuhan University, Hubei 430072, China*
[2] *Institute of Microelectronics and Information Technology, Wuhan University, Hubei 430072, China*
[3] *College of Mathematics and Computer Science, Fuzhou University, Fujian 350002, China*
[4] *Fuzhou Electric Power Bureau, Fujian 350009, China*
E-mail: gaofeng@whu.edu.cn, dongchen@fzu.edu.cn

Abstract **- The particle swarm optimization algorithm (PSO) has successfully been applied to many engineering optimization problems. However, most of the existing improved PSO algorithms work well only for small-scale problems. In this new self-adaptive PSO, a special function, which is defined in terms of the particle fitness and swarm size, is introduced to adjust the inertia weight adaptively. In a given generation, the inertia weight for particles with good fitness is decreased to accelerate the convergence rate, whereas the inertia weight for particles with inferior fitness is increased to enhance the global exploration abilities. When the swarm size is large, a smaller inertia weight is utilized to enhance the local search capability for fast convergence rate. If the swarm size is small, a larger inertia weight is employed to improve the global search capability for finding the global optimum. This novel self-adaptive PSO can greatly accelerate the convergence rate and improve the capability to reach the global minimum for large-scale problems. Moreover, this new self-adaptive PSO exhibits a consistent methodology: a larger swarm size leads to a better performance.**

Index Terms - PSO, Self-adapting, Swarm size, Inertia weight

I. INTRODUCTION

The Particle Swarm Optimization algorithm (PSO) was originally introduced by Eberhart and Kennedy in 1995 [3]. In order to make the PSO more efficient, various improvements over the original PSO have been made. Shi and Eberhart put forward the conception of inertia weight [5], in order to control the search better, and this method called standard-PSO. Before long, Shi and Eberbart depicted the linearly decreasing inertia weight PSO [7]. They found that if the inertia weight change form 0.9 to 0.4 with time, can accelerate the convergence rate. Carlisle and Dozier worked on parameters of the PSO, and found a set of parameters worked well in most problems: c_1 is 2.8, c_2 is 1.3, swarm size is 30, and V_{max} is X_{max} [2].

However, most of the existing improved PSO algorithms work well only for small-scale problems. When the problem scale gets bigger, the large swarm size will significantly reduce the convergence rate of the existing improved PSO algorithms and trap the searching into local minima. In this work, a new self-adaptive PSO is proposed for effectively tackling large-scale problems.

II. STANDARD PSO

In a D-dimensional solution space, the position of the i-th particle is denoted as the D-dimensional vector: $X_i = (X_{i1}, X_{i2},\ldots, X_{iD})$. Moreover, the velocity of the i-th particle also be defined by a D-dimensional vector: $V_i = (V_{i1}, V_{i2}, \ldots, V_{iD})$.

The position of a particle represents a possible solution to the optimization problem under study. The fitness value is an indictor of the quality of the particle position as a solution candidate to the optimization problem under study.

Denote the best position so far of the i-th particle as $p_i = (p_{i1}, p_{i2}, \ldots, p_{iD})$ and the index of the best particle position among $\{p_i\}$ in the population (i.e., the best position so far of the population) as g. Then, the particle positions and velocities can be updated according to the equations (1):

$$\begin{aligned} V_{id} &\leftarrow w\cdot V_{id} + c_1\cdot rand()\cdot(p_{id}-X_{id}) + c_2\cdot rand()\cdot(p_{gd}-X_{id}) \\ X_{id} &\leftarrow X_{id}+V_{id} \end{aligned} \quad (1)$$

where w is the inertia weight, *rand()* is random function whose value is between 0 and 1, and c_1 and c_2 are both positive constants, which are called as acceleration coefficients. A maximum allowable velocity, V_{max}, is a prescribed constant that is often used for limiting the velocity of the particles.

III. INERTIA WEIGHT, FITNESS VALUE AND SWARM SIZE

The inertia weight w is utilized to adjust the influence of the previous velocity on the current velocity, and to balance between global and local exploration abilities of the "flying particle" [6,8]. At the end of an optimization procedure, the diversity in the population is small, and the velocities of particles decreases to zero gradually. To prevent this situation, the utilization of different inertia weights for different particles would play a vital role [7].

In the PSO model, it is believed that the particle with a better fitness value is closer to the global optimum than those with inferior fitness value. For the particle with a better fitness, it might need stronger local exploration ability to search through its local surrounding region for the optimal solution. On the other hand, a particle with inferior fitness would need stronger global exploration ability so that it can quickly move to the particles with better fitness, which can increase the probability of finding the global optimum and

* This work was supported by the National Natural Science Foundation of China under the National Outstanding Young Scientist Award No. 60788402 and the Natural Science Foundation of Fujian Province of China (NO.S0750006).

accelerate the convergence of the PSO.

For particles with better fitness, lower inertia weights can be used, which is beneficial to accelerating the convergence of the PSO. For particles with inferior fitness and thus far away from the best particle position, higher inertia weights will be utilized, which can enhance their global exploration abilities and thus is beneficial for these particles to escape from local optima [7].

The swarm size is one of the vital parameters in the PSO. When the swarm size is too small, the PSO may have too few particles to effectively cover the search space for finding the global optimum. On the other hand, if the swarm size is too large, the PSO will have too many particles that will slow down the convergence rate [10].

A remedy for the above issues can be implemented by adjusting the inertia weight according to the swarm size. When the swarm size is large, a smaller inertia weight can be utilized to enhance the local search capability for fast convergence rate. If the swarm size is small, a larger inertia weight can be employed to improve the global search capability for finding the global optimum.

In summary, there exists an underlying relation among the inertia weight, fitness, and swarm size, which could be used for accelerating the convergence of the PSO. Specifically, define an self-adaptive inertia weight function for a particle in terms of its fitness value and the swarm size as equation (2):

$$W = \left[3 - \exp\left(-S/200\right) + \left(R/100\right)^2\right]^{-1} \quad (2)$$

Where S is the total number of the particles in the population, i.e., the swarm size, and R denotes the fitness rank of the given particle.

IV. The New Self-Adapting PSO Algorithm

By inserting the inertia weight function defined by (2) into the PSO updating equation (1), an enhanced PSO with a self-adaptive capability is attained. The performance of this new PSO is greatly improved by the self-adaptive feature. Different particles in a swarm may be assigned to different values for the inertia weight to update their velocities in terms of their respective fitness values. The algorithm can be summarized as follows:

1) Initialize the particle swarm. Initialize m particles with random positions and velocities within the search space.

2) Calculate the fitness value of each particle.

3) Compare the fitness values at the present position with its best position p_i so far for each particle: If the fitness value at the present position is better than that at its best position, set the present position as the best position so far for this particle. Otherwise, keep the best position unchanged for this particle.

4) Compare the fitness value of each particle with the fitness value at the global best position p_g within the population: If the fitness value of a particle is better than that at the global best position, set the present position of this particle as the best global position so far for the population. Otherwise, keep the original best global position for the population.

5) Calculate the self-adaptive inertia weight for each particle within the population by using equation (2) in terms of the fitness value of this particle and the swarm size of the population.

6) Update the velocity and position of each particle by using equation (1).

7) Check the termination criterion is satisfied: If the criterion is met, then stop. Otherwise, go to step (2).

V. Experiments And Discussions

The performance of this new self-adaptive PSO (SA-PSO) algorithm will be evaluated in comparison with the linearly decreasing inertia weight PSO (LD-PSO) algorithm.

A. Experiments Settings

Three benchmark functions popularly used in the previous publications [1,4,9,10,11] will be used here for the experiments on the new self-adaptive PSO algorithm:

1) DeJong function

$$f_1(x) = \sum_{i=1}^{D} x_i^2 \quad (3)$$

2) Rosenbrock function

$$f_2(x) = \sum_{i=1}^{D} \left[100\left(x_{i+1} - x_i^2\right)^2 + \left(x_i - 1\right)^2\right] \quad (4)$$

3）Griewank function

$$f_3(x) = (\sum_{i=1}^{D} x_i^2)/4000 - \prod_{i=1}^{D} \cos\left(x_i/\sqrt{i}\right) + 1 \quad (5)$$

For these benchmark functions, different population sizes of 20, 100, and 200 are used for 1000, 1500 and 2000 the max iterations. The random approach is utilized to initialize the first generation of particles. The inertia weight is initially set as 0.9, and decreasing to 0.4 linearly over iterations for the linearly decreasing inertia weight PSO.

The neighbour particles are the two particles on the both sides. A run is terminated if the iterations reach the maximum number. Each experiment will be conducted for 20 runs and the best fitness values for all the runs are averaged to give a mean value for the best fitness values. The other parameters of the PSO are listed in Table I.

TABLE I
PARAMETERS SETTINGS

Function	*Dimension Size*	Acceleration Coefficients	Initialization Range	V_{max}
f_1	30	$c_1 = c_2 = 2$	(2.56,5.12)	10
f_2	30	$c_1 = c_2 = 2$	(15,30)	100
f_3	30	$c_1 = c_2 = 2$	(300,600)	600

B. Experiment I: DeJong Function

The experimental results on the DeJong function are listed in Table II. The DeJong function is continuous, convex and unimodal. As shown in Table II, this new self-adaptive PSO

(SA-PSO) has a much improved performance over the linearly decreasing inertia weight PSO (LD-PSO).

TABLE II
EXPERIMENTAL RESULTS FOR DEJONG FUNCTION

Swarm Size S	Generation	Mean of the Best Fitness Values	
		LD-PSO	This SA-PSO
20	1000	1.2561e-005	6.4150e-011
100	1500	6.5734e-015	6.2127e-044
200	2000	1.9798e-023	1.9626e-081

C. Experiment II：Rosenbrock Function

The experimental results on the Rosenbrock function are listed in Table III. The minimization of the Rosenbrock function is a classic optimization problem. The global optimum of the Rosenbrock function is in a long, narrow, parabolic shaped valley. To find the valley is trivial. However, the convergence to the global optimum is difficult [5]. As illustrated in Table III, this new self-adaptive PSO converges to a closer position to the global optimum and gets a better value than another algorithm.

TABLE III
EXPERIMENTAL RESULTS FOR ROSENBROCK FUNCTION

Swarm Size S	Generation	Mean of the Best Fitness Values	
		LD-PSO	This SA-PSO
20	1000	1188.8081	109.3384
100	1500	100.6765	40.9744
200	2000	63.2639	19.5096

D. Experiment III：Griewank Function

The experimental results on the *Griewank* function are listed in Table IV. The Griewank function is a multimodal function and the locations of its minima are regularly distributed [1,4,9,10,11]. This new self-adaptive PSO and the linearly decreasing inertia weight PSO give almost equal mean of the best fitness values for the Griewank function. It is worthwhile to mention that there are many times of finding optima: 0 (the global optimum) by this new self-adaptive PSO, especially on the condition: the population size of 100 and the max iterations of 1500. It seems to suggest that this new self-adaptive PSO has the ability to escape from the local optimum.

TABLE IV
EXPERIMENTAL RESULTS FOR GRIEWANK FUNCTION

Swarm Size S	Generation	Mean of the Best Fitness Values	
		LD-PSO	This SA-PSO
20	1000	0.0314	0.0138
100	1500	0.0118	0.0044
200	2000	0.0172	0.0111

From the above experiments, it can be observed that the SA-PSO has a great improvement over LD-PSO. In addition, the linearly decreasing inertia weight PSO converges at irregular rates and thus increase the instability of the algorithms.

CONCLUSIONS

In this work, based on the observation and careful analysis on the relation among the inertia weight, the fitness and the swarm size, a new self-adaptive PSO is introduced by defining an inertia weight function in terms of the fitness and the swarm size. In a given generation, the inertia weight for particles with good fitness has been decreased to accelerate the convergence rate, whereas the inertia weight for particles with inferior fitness has been increased to enhance the global exploration abilities. When the swarm size is large, a smaller inertia weight has been utilized to enhance the local search capability for fast convergence rate. If the swarm size is small, a larger inertia weight has been employed to improve the global search capability for finding the global optimum.

This new self-adaptive PSO provides a great improvement over the convergence rate on optimization problems. Moreover, this new self-adaptive PSO exhibits a consistent methodology: a larger swarm size leads to a better performance. Finally, the uniformity of the convergence rate enhances the robustness and stability of the optimization algorithm. This new method extends the PSO applicability to large-scale engineering optimization problems, such as the optimal designs of very-large-scale integration (VLSI) circuits.

Three benchmark functions have been used to test the performance of this new self-adaptive PSO. At the same conditions, the convergence rate of this new self-adaptive PSO is faster hundreds of millions times than that of the linearly decreasing inertia weight PSO. Experiments also illustrated that this new self-adaptive PSO has ability to escape from local optima.

REFERENCES

[1] Angeline, P.J. (1998), "Using selection to improve particle swarm optimization", *Proceedings of IEEE Congress on Evolutionary Computation*, Anchorage, Alaska, pp. 84-89.

[2] Carlisle, A. and Dozier, G. (2001), "An off-the-shelf PSO", *Proceedings of the Workshop on Particle Swarm Optimization*, Indianapolis, IN.

[3] Eberhart, R.C. and Kennedy, J. (1995), "A new optimizer using particles swarm theory", *Proeedings of Sixth International Symposium on Micro Machine and Human Science*, Nagoya, Japan, pp. 39~43.

[4] Kennedy, J. and Eberhart, R.C. (1997), "A discrete binary version of the particle swarm algorithm", *Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics*, Orlando, FL, pp. 4101-4109.

[5] Shi, Y.H. and Eberhart, R.C. (1998), "A modified particle swarm optimizer", *Proceedings of IEEE International conference of Evolutionary Computation*, Anchorage, Alaska, pp. 69-73.

[6] Shi, Y.H. and Eberhart, R.C. (1998), "Parameter selection in particle swarm optimization", in Eiben A, Porto V, Saravanan N, Waagen D. (Eds), *Evolutionary Programming VII*. San Diego, California,USA: Springer-Verlag, pp. 591-600.

[7] Shi, Y.H. and Eberhart, R.C. (1999), "Empirical study of particle swarm optimization", *Proceedings of IEEE Congress on Evolutionary Computation*, Washington DC, pp. 1945-1950.

[8] Shi, Y.H. and Eberhart, R.C. (2000), "Experimental study of particle swarm optimization", *Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics*, Orlando, FL.
[9] Shi, Y.H. and Eberhart, R.C. (2001), "Fuzzy adaptive particle swarm optimization", *Proceedings of IEEE Congress on Evolutionary Computation*, San Francisco, CA, pp. 101-106
[10] Van Den Bergh, F. and Engelbrecht, A.P. (2001), "Effects of swarm size on cooperative particle swarm optimizers", *Proceedings of the Genetic and Evolutionary Computation Conference*, San Francisco, California, pp. 892-899.
[11] Peer, E. S, Van den Bergh, F. and Engelbrecht, A.P. (2003), "Using neighborhoods with the guaranteed convergence PSO", *Proceedings of the IEEE Swarm Intelligence Symposium*, Indianapolis, IN, pp. 235-242.