TEST Engineering & Management

May – June 2020
ISSN: 0193-4120 Page No. 2146 - 2149

A Study on Weightless Particle Swarm Optimization with a Globally Best Particle

Idris AbdLatiff
Electrical Technology Section
Universiti Kuala Lumpur British Malaysian Institute
Selangor, Malaysia
idris@unikl.edu.my

Article Info
Volume 83
Page Number: 2146 - 2149
Publication Issue:
May - June 2020

Article History
***Article Received**: 11August 2019*
***Revised**: 18November 2019*
***Accepted**: 23January 2020*
***Publication**: 10 May2020*

Abstract:

Particle Swarm Optimization (PSO) has been established as an efficient computational intelligencetool since its introduction. Much of the improvement made on the particle swarm algorithm centered on the effect of a parameter called the inertia weight. In this study, the effect of the absence of inertia weight on the performance of PSO algorithm has been analyzed. A new term called the global particle has been introduced in the velocity update equation. This term has been able to compensate the absence of the inertia weightin order to maintain the convergence ability of the algorithm. Test results with standard objective functions demonstrate the necessity of having the inertia weight and justify the effort and research spent on developing many variants based on this important parameter. The results also show where the inertia weight term can be omitted to save computational cost.

Keywords: *particle swarm, convergence,inertia weight, global search*

I. INTRODUCTION

PARTICLE swarm optimization (PSO) was introduced by James Kennedy and RusselEberhart in 1995 [1][2] as a population-based search method based on the behavior of birds flocking and fish schooling. PSO and scores of other swarm-based algorithms such as Ant Colony Optimization, Bacterial Foraging and the more recent ones such as Bats Algorithm and Grasshopper Optimization, form a family of Swarm Intelligence optimization tools [3]. These nature-inspired or bio-inspired heuristics were developed based on the methods adopted by creatures in nature to survivethe hostile environment, find food sources, and avoid the threat of predators.

PSOin particular is made up of a population of solutions called particles. These particles move in the search space of the objective function to find the optimum point, which in most cases are the minima of the functions.In this paper, the goal is to minimize the objective functions, without loss of generality. The main process in the PSO algorithm involves two equations: velocity update and position update as described by equations (2) and (3) respectively.

If the *i*th particle in a *d*-dimensional search space be

represented by

$$X_i = (x_{i1}, x_{i2}, ..., x_{id}) \quad (1)$$

then, each particle can thus be governed by

$$v_{id} = wv_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}) \quad (2)$$

and

$$x_{id} = x_{id} + v_{id} \quad (3)$$

where$w$ is the inertia weight,v_i is the velocity of particle i, x_i is the position of particle i, c_1 is the cognition factor, c_2 is the social factor, r_1 and r_2 are uniformly distributed random numbers between 0 and 1, p_i is personal best (*pbest*), i.e. the best position of particle i so far, and p_g is global best (*gbest*), i.e. the best position of the particle in the population so far.

II. THE INERTIA WEIGHT

The inertia weight has gained much attention due

May – June 2020
ISSN: 0193-4120 Page No. 2146 - 2149

to its effect on the both the exploration and exploitation phases of the search. Generally, a large inertia weight is applied at the beginning of the search. Once the minimum has been located and the particles start converging, small inertia weight is applied to exploit or fine tune the solution for high accuracy. Thus various schemes have been devised such as constricted inertia weight [4], time-decreasing inertia weight [5], fuzzy-controlled inertia weight [6] and adaptive inertia weight [7].

Another good strategy for efficient convergence and achieving good balance between exploration and exploitation is by varying both the inertia weight and acceleration coefficients with [8]. In this case, all three parameters are linearly decreased from their initial values to predetermined final values as given by equations (4), (5), and (6).

$$w_t = (w_1 - w_2)\frac{\max\_t - t}{\max\_t} + w_2 \tag{4}$$

$$c_{1t} = (c_{1f} - c_{1i})\frac{t}{\max\_t} + c_{1i} \tag{5}$$

$$c_{2t} = (c_{2f} - c_{2i})\frac{t}{\max\_t} + c_{2i} \tag{6}$$

In this study, both acceleration coefficientsor learning factors used are linearly decreasing from an initial value of 2 as given by equation (7).

$$c_{1,2} = 2\times(1-(iter/iteration\_\max)) \tag{7}$$

III. Initial Test Results

Table I shows the list objective test functions used in this study as well as the results. The test done employed 20 particles with the maximum number of iterations is set to 100. The convergence of particle is defined as the Spread parameter as described in [7]. Essentially, the Spread is the distance of the two farthest particles in opposite direction for each dimension. This is also taken as the termination criteria which is set at $1\text{x}10^{-6}$. The number of runs is 50 times and the results are averaged.

TABLE I

No	Name	Dimension	Success Rate
F1	Sphere	30	0%
F2	Rastrigin	30	0%
F3	Griewank	30	0%
F4	Ackley	30	0%
F5	Schwefel	30	0%
F6	Rosenbrock	2	20%
F7	Michalewicz	2	0%
F8	Easom	2	10%
F9	Weierstrass	10	0%

Initial test results obtained by simply removing the inertia weight term highlight the importance and significance of the parameter. Without the inertia weight, the algorithm failed to find the global minimum in almost all test functions. Furthermore, the particles also failed to find local minima in F2, F3, F4, F5, F7 and F9. On F1, even with a single minimum, the particles seems unable to converge. Success rates of 20% and 10% obtained in F6 and F8 respectively were probably due to the landscape of the test function. Nevertheless, the results were not encouraging. For better illustration, the profile of first five functions are shown in Figure 1.

IV. Global Particle

To improve the convergence of the PSO algorithm without the inertia weight, a third term was added to the velocity update equation as follows

$$v_{id} = wv_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}) \ldots.$$

$$+ c_3 r_3 (P_G - x_{id}) \tag{8}$$

In equation (8), c_3 is called the global factor and P_G is called the globally best particle [9]. It is a virtual particle created to enable particles to escape from local minima and premature convergence. This is useful in the exploitation phase where the global best p_{gd} hardly moves. Adding this term to the velocity replacing the inertia weight has resulted in a much better performance as shown in Table II. The results obtained in Table II employed the same parameter setting as in the initial test except with the presence of the globally best particle.

Success rate of 100% was achieved in test function F1, F3, F4 and F9. This is a remarkable improvement from 0% success. The success rates in F2 and F8 of 86% and 80% respectively are also encouraging. However there seems to be a limitation to this method as the poor results were obtained in

May – June 2020
ISSN: 0193-4120 Page No. 2146 - 2149

F5, F6 and F7. A common factor among the test functions with good results is that the global minimum is located at the origin which is not the case in F5, F6 and F7. The exception is the function F8 which is quite easy since there is no local minima.

The following observation can be drawn from the results.In this study, the inertia weight has been removed to save the computation cost of the PSO algorithm. The calculation of the inertia weight can be simple in the case of fixed or linearly decreasing values, but can be time consuming in the case of adaptive or fuzzy-based calculation. Thus, in its place another term was added where the parameter does not require much computation time. This method has produced good results in a certain type of test functions. Therefore, a useful knowledge of the problem can be valuable and enable us to simplify the algorithm without sacrificing its efficiency.

V. Conclusion

In this study, the inertia weight has been removed and replaced with another term. The results were mixed. They proved that successful convergence depends on the landscape of the test function or the nature of the problem. The test results not only illustrate the importance of the inertia weight but also enable us to omit the parameter and simplify the algorithm provided that valuable information of the problem is available.

References

[1] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Networks*, Perth, Australia, 1995, pp. 1942-1948.

[2] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micro Machine and Human Science*, Nagoya, Japan, 1995, pp. 39-43.

[3] A. Slowik and H. Kwasnicka, "Nature inspired methods and their industry applications – swarm intelligence algorithms," in *IEEE Transactions on Industrial Informatics*, vol. 4(3), pp. 1004-1015, Mar 2018.

[4] M. Clerc, "The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization," in *Proc. Congr. Evolutionary Computation*, WashingtonDC, 1999, pp. 1951-1957.

[5] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in *Proc. Congr. Evolutionary Computation*, La Jolla, CA, 2000, pp. 84-88.

[6] Y. Shi and R. C. Eberhart, "Fuzzy adaptive particle swarm optimization," in *Proc. Congr. Evolutionary Computation*, Seoul, South Korea, 2001, pp. 101-106.

[7] I. AbdLatiff and M. O. Tokhi,"Fast Convergence Strategy for Particle Swarm Optimization using Spread Factor," in *Proc. IEEE Int. Conf. Evolutionary Computation*, Trondheim, Norway, 2009, pp. 2693-2700.

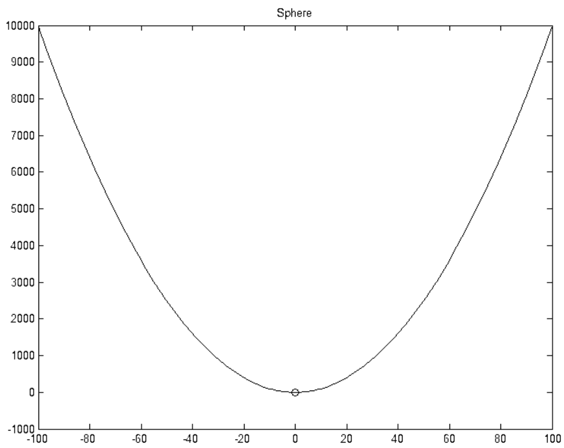
[8] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self- organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Trans. Evolutionary Computation*, vol. 8, pp. 240-255, June 2004.

[9] I. AbdLatiff, *PhD Thesis*, The University of Sheffield, UK, 2011.

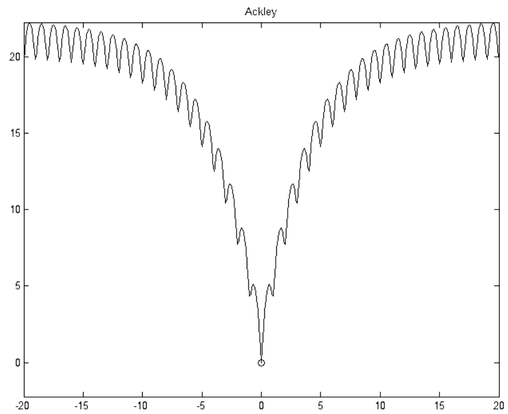
Table II

No	Iterations	Best Fitness	Mean Fitness	Spread	Success Rate
F1	55	3.954E-35	7.522E-12	5.620E-07	100
F2	58	4.480E-08	1.421E-07	1.444E-06	86
F3	58	1.325E-12	1.454E-12	5.998E-07	100
F4	53	2.522E-15	6.916E-07	5.566E-07	100
F5	75	3.938E-04	3.938E-04	5.077E-06	52
F6	100	6.255E-05	8.238E-04	4.331E-03	52
F7	100	-1.80	-1.38	5.739E-03	10
F8	69	-1.00	-1.00	4.930E-05	80
F9	51	2.732E-05	5.470E-04	4.904E-07	100

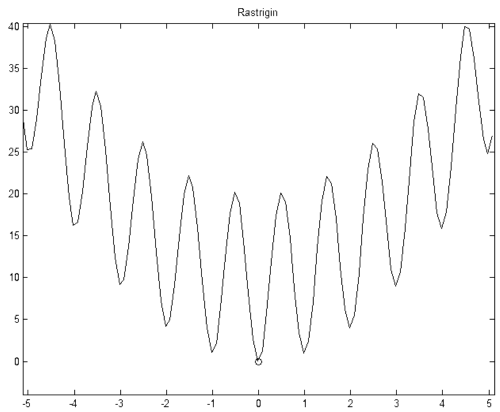
May – June 2020
ISSN: 0193-4120 Page No. 2146 - 2149



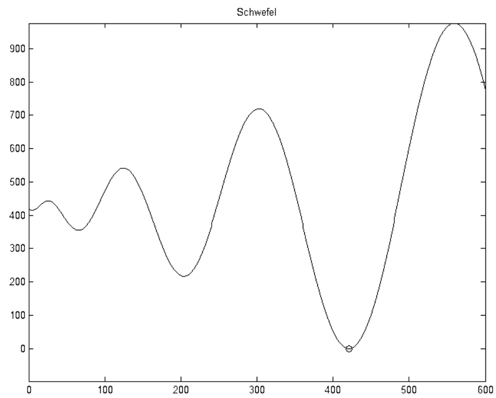
(a) Sphere, *x* = [-100,100]



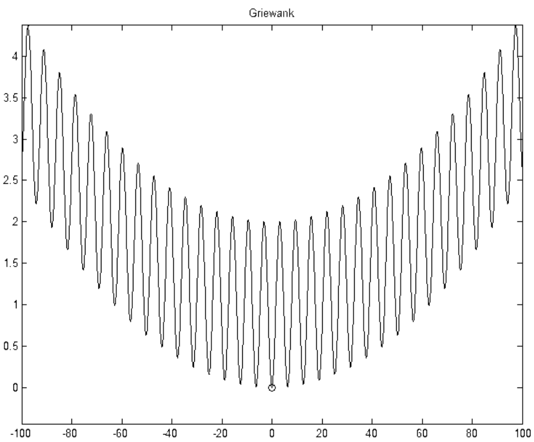
(d) Ackley,*x* = [-20, 20]



(b) Rastrigin, *x* = [-5.12, 5.12]



(e)Schwefel, *x* = [0, 600]



(c) Griewank, *x* = [-100, 100]