Proc. of The Third Intl. Conf. On Advances in Applied Science and Environmental Technology - ASET 2015
Copyright © Institute of Research Engineers and Doctors, USA .All rights reserved.
ISBN: 978-1-63248-084-2 doi: 10.15224/ 978-1-63248-084-2-33

Double Inertia Weight-Based Particle Swarm Optimization

[Yu-Huei Cheng*, Che-Nan Kuo, and Ching-Ming Lai]

Abstract—**Particle swarm optimization (PSO) is a well-known and popular swarm intelligence algorithm. The inertia weight of a PSO plays the crucial role in the ability of exploration and exploitation. Many strategies for adapting the inertia weight of PSO have been proposed. In this study, we use two inertia weights to improve the global and local search of PSO. Nine benchmark functions with 10 dimensions for unimodal functions, multimodal functions with many local optima, and multimodal functions with a few local optima is used as the test functions. We compare two inertia weight PSOs with the proposed method. The results show the proposed method is useful for improve the search ability of PSO.**

Keywords—**benchmark functions, double inertia weight, particle swarm optimization (PSO)**

I. Introduction

Particle swarm optimization (PSO) is a popular population-based algorithm [1], and many real-world problems are effectively solved by it. For examples, EMG (electromyogram) signal classification [2], image filter [3], decoupling control for temperature of reheating furnace [4], harmonic filters [5], Ultrawideband (UWB) Antenna Synthesis [6], flow shop scheduling [7], learning to play games [8] and so on. In PSOs, how the local and global search is controlled will influence performance on finding the optimal solution directly. In many variants of the PSOs, the inertia weight is considered significant to adjust their search ability with exploration and exploitation. The inertia weight can balance the global and local search. A large inertia weight facilitates a global search while a small inertia weight facilitates a local search [9].

Yu-Huei Cheng
Dept. of Mobile Technology, Toko University
Taiwan

*corresponding author

Che-Nan Kuo
Dept. of Animation and Game Design, Toko University
Taiwan

Ching-Ming Lai
Dept. of Vehicle Engineering, National Taipei University of Technology
Taiwan

In the past, several manners for associating with the inertia weight of PSO were proposed. The fixed inertia weight PSO [10] is first introduced into the original particle swarm optimizer by Shi and Eberhart. They performed different chosen inertia weight to illustrate the impact of this parameter on the performance of PSO. And then, Shi and Eberhart use linearly decreasing inertia weight PSO [11], which decreases the inertia weight from a relatively large value to a small value through iterations. It tends to own more global search ability at the beginning of iterations and more local search ability around the end of iterations. However, the fixed or linearly decreasing inertia weight PSO is not very effective for tracking nonlinear dynamic systems most real-world applications.

In this study, we propose two inertia weights to improve the global and local search of PSO. The above two inertia weight controlled PSOs are also performed. The three methods have been tested and comparisons each other on nine benchmark functions with 10 dimensions for unimodal functions, multimodal functions with many local optima, and multimodal functions with a few local optima.

II. Methods

The PSO was inspired by the social behaviors of a bird flock or fish school. The PSO first generates a population of random solutions called particles. Each particle has its own velocity and position. Following that, it searches for optimal solution by updating generations through the update equations for the velocities and the positions of particles. Before updating the velocities and the positions, all particles are evaluated by an objective function and compared with the previous positions to gain the personal best positions, and compared with each other to gain the global best position. In each generation, the current velocities will be updated according to the previous positions, the personal best positions and the global best position. Each particle then moves to a new position according to its current velocity and its previous position. The personal and global best positions particle is always be improved by generation to generation, and thus lead other particles accelerates in the direction to move.

A. *Original PSO*

The original PSO was proposed by Eberhart and Kennedy [1]. Let N is dimensions for an optimization problem for search space. Four characteristics are described as follows:

The position of the ith particle is represented as $X_i = (x_{i1}, x_{i2}, \dots , x_{iN})$. The personal best position of the ith particle is represented as $pbest_i = (p_{i1}, p_{i2}, \dots , p_{iN})$. The global best position found from all the particles is represented as $gbest =$

Proc. of The Third Intl. Conf. On Advances in Applied Science and Environmental Technology - ASET 2015
Copyright © Institute of Research Engineers and Doctors, USA .All rights reserved.
ISBN: 978-1-63248-084-2 doi: 10.15224/ 978-1-63248-084-2-33

($g_1, g_2, \dots, g_N$). The velocity of the *i*th particle is represented as $V_i = (v_{i1}, v_{i2}, \dots, v_{iN})$. The value of velocity V_i is restricted to the range of $[-V_{\max}, V_{\max}]$ to prevent particles from moving out of the search space.

In original PSO, each particle in the swarm is iteratively updated according to the aforementioned characteristics. Assume the objective function of an optimization problem is defined as objective(X_i) and it is minimized.

The personal best position of each particle is found by

$$pbest_i(g+1) = \begin{cases} pbest_i(g), \text{if objective}(X_i(g+1)) \geq \text{objective}(pbest_i(g)) \\ X_i(g+1), \text{otherwise} \end{cases} \tag{1}$$

where *i* is a serial number of the particle; *g* is the current generation; $pbest_i(g)$ is the personal best position of the *i*th particle in the current generation; $X_i(g)$ represents the position of the *i*th particle in the current generation; objective($X_i(g)$) is the objective function for getting objective value of a particle $X_i(g)$; $pbest_i(g+1)$ is the personal best position of the *i*th particle in the next generation.

The global best position is found by

$$\begin{aligned} &gbest(g+1) = pbest_k(g+1), \\ &if \;\; pbest_k(g+1) \; is \; \min(\forall objective(pbest(g+1))) \end{aligned} \tag{2}$$

where *g* is the current generation; $pbest_k(g+1)$ is a position with the minimum objective value; $\min(\forall \text{objective}(pbest(g+1)))$ represents a function for getting the minimum objective value of $pbest_i(g+1)$ from all $pbest(g+1)$ in the next generation; $gbest(g+1)$ is the global best position found from all the particles in the next generation.

The equation for the new velocity of every particle is defined as

$$\begin{aligned} V_i(g+1) = V_i(g) &+ c_1 \times r_{1i}(g) \times [pbest_i(g) - X_i(g)] \\ &+ c_2 \times r_{2i}(g) \times [gbest(g) - X_i(g)] \end{aligned} \tag{3}$$

where *i* is a serial number of the particle; *g* is the current generation; c_1 and c_2 denote the acceleration coefficients; $r_{1i}(g)$ and $r_{2i}(g)$ are the uniform random values of the *i*th particle in the range (0, 1) in the current generation; $V_i(g)$ is the velocity of the *i*th particle in the current generation; $pbest_i(g)$ is the personal best position of the *i*th particle in the current generation; $gbest(g)$ is the global best position found from all the particles in the current generation; $X_i(g)$ represents the position of the *i*th particle in the current generation; $V_i(g+1)$ is the velocity of the *i*th particle in the next generation.

The current position of each particle is updated using

$$X_i(g+1) = X_i(g) + V_i(g+1) \tag{4}$$

where $X_i(g)$ represents the position of the *i*th particle in the current generation; $V_i(g+1)$ is the velocity of the *i*th particle in the next generation; $X_i(g+1)$ represents the position of the *i*th particle in the next generation.

B. *Fixed inertia weight PSO*

The original PSO introduces a parameter called inertia weight (*w*) to the updating equation of the velocity [10]. The influence of using fixed inertia weight has been estimated. The fixed inertia weight was considered to be chosen a good area on the range [0.9, 1.2] [10]. In this study, we use the abbreviation of "FWPSO" to represent the fixed inertia weight PSO. The new updating equation of the velocity is given as follows:

$$\begin{aligned} V_i(g+1) = w \times V_i(g) &+ c_1 \times r_{1i}(g) \times [pbest_i(g) - X_i(g)] \\ &+ c_2 \times r_{2i}(g) \times [gbest(g) - X_i(g)] \end{aligned} \tag{5}$$

where *w* is inertia weight; the other variables and functions are the same as above-mentioned.

C. *Linearly decreasing inertia weight PSO*

A large inertia weight facilitates a global search while a small inertia weight facilitates a local search. Linearly decreasing inertia weight PSO (LDWPSO) by linearly decreasing the inertia weight from a relatively large value ($w_{\max}$ = 0.9) to a small value ($w_{\min}$ = 0.4) through iterations (i.e., generations) to control the global search and local search. It had been reported to be better than the fixed inertia weight PSO on the benchmark problem of Schaffer's F6 function [11] and four non-linear testing functions [9]. The decreasing inertia weight is calculated as

$$w(g+1) = w_{\max} - \frac{w_{\max} - w_{\min}}{G} \times g \tag{6}$$

where $w_{\max}$ is the maximum inertia weight value; $w_{\min}$ is the minimum inertia weight value; *G* is the maximum number of generations; *g* is the current generation.

D. *Double inertia weight PSO*

In order to control the global and local search, we propose double inertia weight PSO (DWPSO) here. The DWPSO use two inertia weights, one is focused on the global search (w_{global} = 0.9), and the other is focused on the local search (w_{local} = 0.4). At the beginning iterations, the large inertia weight is enabled for global search, and at the ending iterations, the small inertia weight is enabled for local search. The double inertia weight is set according to

$$w(g+1) = \begin{cases} w_{global}, \text{ when } g > G/2 \\ w_{local}, \text{ otherwise} \end{cases} \tag{7}$$

where w_{global} means the large inertia weight for global search; w_{global} means the small inertia weight for local search; *g* is the current generation; *G* is the maximum number of the preset generations; $w(g+1)$ represents the inertia weight in the next generation.

Proc. of The Third Intl. Conf. On Advances in Applied Science and Environmental Technology - ASET 2015
Copyright © Institute of Research Engineers and Doctors, USA .All rights reserved.
ISBN: 978-1-63248-084-2 doi: 10.15224/ 978-1-63248-084-2-33

III. Benchmark functions

Nine benchmark functions including unimodal and multimodal problems was used for evaluating the inertia weight PSOs. Table I lists the nine benchmark functions and their modality, global optimum, search space and initial ranges. These functions are divided into three parts, i.e., A. unimodal functions, B. multimodal functions with many local optima, and C. multimodal functions with a few local optima. They are described below.

TABLE I. MODALITY, GLOBAL OPTIMUM, SEARCH SPACE AND INITIAL RANGES OF NINE TEST FUNCTIONS

Modality	Function *f*		Global optimum	Search space	Initial range
Unimodal	f_1	Hyperellipsoid	0	$[-1.0, 1.0]^N$	$[-1.0, 1.0]^N$
	f_2	Quartic	0	$[-1.28, 1.28]^N$	$[-1.28, 1.28]^N$
	f_3	Step	0	$[-100.0, 100.0]^N$	$[-100.0, 50.0]^N$
Multimodal with many local optima	f_4	Griewank	0	$[-600.0, 600.0]^N$	$[-600.0, 300.0]^N$
	f_5	Levy	0	$[-10.0, 10.0]^N$	$[-10.0, 10.0]^N$
	f_6	Schaffer f6	0	$[-100.0, 100.0]^N$	$[-100.0, 50.0]^N$
Multimodal with a few local optima	f_7	Branin	0.398	$[-5.0, 15.0]^N$	$[-5.0, 15.0]^N$
	f_8	Goldstein-price	0	$[-2.0, 2.0]^N$	$[-2.0, 2.0]^N$
	f_9	Six-hump camel-back	-1.0316285	$[-5.0, 5.0]^N$	$[-5.0, 5.0]^N$

Note: N is the size of dimensions.

A. *Unimodal functions*

Unimodal functions are rather easy to optimize. In this study, $f_1 \sim f_3$ are unimodal functions, they are listed below.

(1) f_1 - Hyperellipsoid

$$f_1(x) = \sum_{i=1}^{N} i^2 x_i^2 \tag{8}$$

(2) f_2 - Quartic function

$$f_2(x) = \sum_{i=1}^{N} (ix^4 + random[0,1]) \tag{9}$$

(3) f_3 - Step function

$$f_3(x) = \sum_{i=1}^{N} (\lfloor x_i + 0.5 \rfloor)^2 \tag{10}$$

B. *Multimodal* functions with many local optima

Functions $f_4 \sim f_6$ are multimodal functions with many local optima. They look to be the most difficult category of problems for many optimization methods. These functions are listed as follows.

(4) f_4 - Griewank's function

$$f_4(x) = \frac{1}{4000}\sum_{i=1}^{N} x_i^2 - \prod_{i=1}^{N} \cos(\frac{x_i}{\sqrt{i}}) + 1 \tag{11}$$

(5) f_5 - Levy

$$f_5(x) = \frac{\pi}{N}\left(k\sin^2(\pi\, y_1) + \sum_{i=1}^{N-1}\left[(y_i - a)^2(1 + k\sin^2(\pi\, y_{i+1}))\right] + (y_n - a)^2 \right)$$

$$y_i = 1 + \frac{1}{4}(x_i - 1),\ k = 10,\ a = 1 \tag{12}$$

(6) f_6 - Schaffer f6

$$f_6(x) = 0.5 + \frac{(\sin\sqrt{x_1^2 + x_2^2})^2 - 0.5}{\left[1 + 0.001(x_1^2 + x_2^2)\right]^2} \tag{13}$$

C. *Multimodal* functions with a few local optima

Functions $f_7 \sim f_9$ are as well as multimodal functions, but they only comprise a few local optima. They are listed as follows.

(7) f_7 - Branin function

$$f_7(x_1, x_2) = (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 + 10(1 - \frac{1}{8\pi})\cos x_1 + 10 \tag{14}$$

(8) f_8 - Goldstein-Price function

$$f_8(x_1, x_2) = \left[1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\right] \times \left[30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)\right] \tag{15}$$

(9) f_9 - Six-hump Camel-back

$$f_9(x_1, x_2) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4 \tag{16}$$

IV. Experiments

We compared the proposed inertia weight PSO to two inertia weight PSOs on the nine benchmark functions which include unimodal functions, multimodal functions with many local optima, and multimodal functions with a few local optima in 10 dimensions, i.e., *N* is set to 10. The three PSOs were implemented in JAVA. The experiments were executed on Pentium 4 CPU 3.4 GHz with 1GB of RAM on Microsoft Windows XP SP3 professional operating system. On the boundary process of these PSOs, we use bound terminal process, i.e., when the particles are over shot, the positions of the particles will be reset to the maximum limit of the search range.

A. *Parameter Settings*

Five main parameters were set for the PSO-based methods, i,e., the number of iterations (10000), the particle swarm size

Proc. of The Third Intl. Conf. On Advances in Applied Science and Environmental Technology - ASET 2015
Copyright © Institute of Research Engineers and Doctors, USA .All rights reserved.
ISBN: 978-1-63248-084-2 doi: 10.15224/ 978-1-63248-084-2-33

(10), the different inertia weight *w* setting approaches, and the constriction factors c_1 and c_2 (2 and 2). Each method was run 30 times and the results for their mean values, standard deviation, and average running time were calculated.

B. *Experimental results*

Table II presents the mean, standard deviation, and average running time of 30 runs for the three inertia weight PSOs on the nine benchmark functions with 10 dimensions in the unimodal functions, the multimodal functions with many local optima, and the multimodal functions with a few local optima. Because only the inertia weight is handled, the results of these PSOs are approximative.

On the unimodal functions, for f_1, the FWPSO performed the worst result for mean value and but the best average run time; the LDWPSO performed the worst result for variance, standard deviation and average run time; the proposed DWPSO performed the best result for mean value, variance, and standard deviation. For f_2, the FWPSO performed the worst result for mean value, variance, standard deviation and average run time; the LDWPSO performed the secondary result for mean value, variance, standard deviation and average run time; the proposed DWPSO performed the best result for mean value, variance, standard deviation and average run time. For f_3, the three methods performed the perfect result for mean value, variance and standard deviation. The LDWPSO had the better average run time than the FWPSO and the proposed DWPSO.

On the multimodal functions with many local optima, for f_4, the FWPSO performed the worst result for variance and standard deviation, but the best average run time; the LDWPSO performed the secondary result for mean value, variance, standard deviation, and average run time; the proposed DWPSO performed the better mean value, variance and standard deviation. For f_5, the three methods performed the same result for mean value, variance and standard deviation. The proposed DWPSO had the better average run time than the FWPSO and LDWPSO. For f_6, the FWPSO performed the best result for mean value, variance and standard deviation; the LDWPSO performed the worst result for mean value, variance and standard deviation; the proposed DWPSO performed the secondary result for mean value, variance and standard deviation, and had the best average run time.

On the multimodal functions with a few local optima, for f_7, the three methods had the same result for mean value. The FWPSO performed the worst result for variance and standard deviation, but the best average run time. The LDWPSO and the proposed DWPSO in addition to the same mean value, they had the same result for variance and standard deviation. For f_8, the FWPSO performed the best result for variance and standard deviation. The LDWPSO performed the same mean value as the FWPSO, and the average run time is the worst one. The DWPSO performed the best mean value and the average run time.

TABLE II. COMPARISONS OF SEARCH RESULTS FOR THE MEAN, VARIATION, STANDARD DEVIATION, AND AVERAGE RUN TIME IN 30 RUNS FOR THE THREE INERTIA WEIGHT PSOS ON THE NINE BENCHMARK FUNCTIONS WITH 10 DIMENSIONS FOR UNIMODAL FUNCTIONS, MULTIMODAL FUNCTIONS WITH MANY LOCAL OPTIMA, AND MULTIMODAL FUNCTIONS WITH A FEW LOCAL OPTIMA.

Functions	Result	FWPSO	LDWPSO	DWPSO
f_1	Mean	6.84E+00	3.77E+00	**3.30E+00**
	Var.	1.69E+01	1.77E+01	**1.51E+01**
	Std. dev.	4.11E+00	4.21E+00	**3.89E+00**
	Avg. run time (ms)	**1106**	1421	1250
f_2	Mean	3.20E+00	1.90E+00	**1.76E+00**
	Var.	1.06E+00	3.34E-01	**5.42E-02**
	Std. dev.	1.03E+00	5.78E-01	**2.33E-01**
	Avg. run time (ms)	1534	1533	**1523**
f_3	Mean	0	**0**	0
	Var.	0	**0**	0
	Std. dev.	0	**0**	0
	Avg. run time (ms)	53	**31**	61
f_4	Mean	3.59E+00	3.66E+00	**2.66E+00**
	Var.	1.66E+01	7.51E+00	**3.64E+00**
	Std. dev.	4.07E+00	2.74E+00	**1.91E+00**
	Avg. run time (ms)	**1179**	1248	1218
f_5	Mean	4.71E-32	4.71E-32	**4.71E-32**
	Var.	1.12E-93	1.12E-93	**1.12E-93**
	Std. dev.	3.34E-47	3.34E-47	**3.34E-47**
	Avg. run time (ms)	1383	1390	**1358**
f_6	Mean	**3.89E-03**	2.11E-02	4.53E-03
	Var.	**2.34E-05**	2.08E-03	2.43E-05
	Std. dev.	**4.84E-03**	4.57E-02	4.93E-03
	Avg. run time (ms)	1682	1304	**1176**
f_7	Mean	3.98E-01	3.98E-01	**3.98E-01**
	Var.	1.68E-12	1.28E-32	**1.28E-32**
	Std. dev.	1.29E-06	1.13E-16	**1.13E-16**
	Avg. run time (ms)	**927**	991	935
f_8	Mean	-5.74E+05	-5.74E+05	**-5.73E+05**
	Var.	**5.99E+06**	6.81E+06	7.12E+06
	Std. dev.	**2.45E+03**	2.61E+03	2.67E+03
	Avg. run time (ms)	947	1084	**743**
f_9	Mean	-1.03E+00	-1.03E+00	**-1.03E+00**
	Var.	3.07E-11	4.59E-31	**4.59E-31**
	Std. dev.	5.54E-06	6.78E-16	**6.78E-16**
	Avg. run time (ms)	1031	1122	**1029**

Note: Var. is the abbreviation of "variance"; std. dev. is the abbreviation of "standard deviation"; Avg. is the abbreviation of "average". The bold represents the best results.

For f_9, the three methods had the same mean value. The FWPSO performed the worst result for variance, standard deviation; the LDWPSO and the proposed DWPSO performed the same variance and standard deviation. The proposed DWPSO had the best average run time.

V. Converging *conditions*

In order to observe the conditions for their convergence, the iterative process is shown in the study. Figures 1 to 3 respectively shows the charts for unimodal functions, multimodal functions with many local optima, and multimodal functions with a few local optima. The chart for unimodal functions as shown in Fig. 1. For the Hyperellipsoid and the Quartic functions, all the three methods did not converge on the optimal solution. The proposed DWPSO converged rapidly than the other methods in Hyperellipsoid function. The proposed DWPSO converged in the better solution than the other methods in Quartic function when the iterations reached to 10000. Furthermore, the chart show Step function was converged rapidly by all the three methods and reached the optimal solution.

The chart for multimodal functions with many local optima as shown in Fig. 2. For the Griewank, Levy, and Schaffer f6 functions, all the three methods did not converge on the optimal solution. The proposed DWPSO converged in the better solution than the other methods in Griewank and Levy functions when the iterations reached to 10000. However, for the Schaffer f6 function, the FWPSO converged in the better solution than the other methods when the iterations reached to 10000.

The chart for multimodal functions with a few local optima as shown in Fig. 3. The converged conditions were similar among the three methods for Branin, Goldstein-price, and Six-hump camel-back functions. From the Table II, we observe the proposed DWPSO generated little better solutions for Branin, Goldstein-price, and Six-hump camel-back functions than the other methods when the iterations reached to 10000.

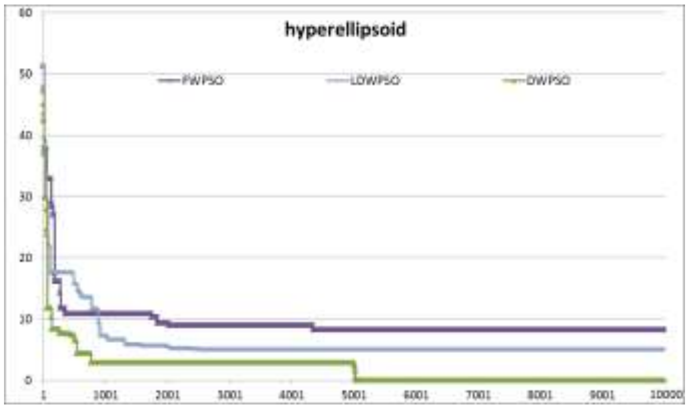
From these results, for the most converged conditions, the proposed DWPSO yield better solutions than the FWPSO and LDWPSO methods on the used unimodal functions, multimodal functions with many local optima, and multimodal functions with a few local optima.

VI. Conclusion

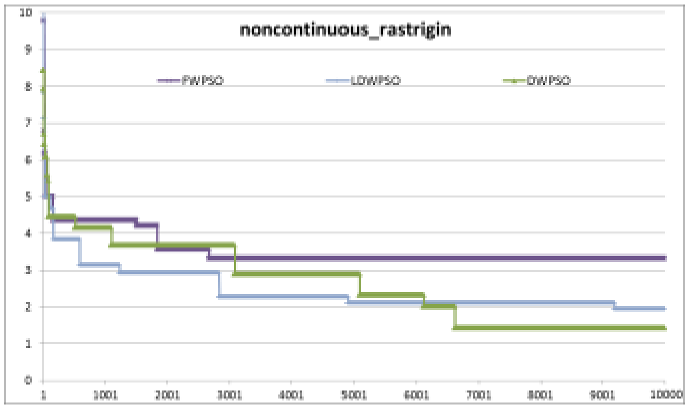
In the study, we implement three inertia weight PSOs and test them on nine benchmark functions for unimodal functions, multimodal functions with many local optima, and multimodal functions with a few local optima. The DWPSO is proposed based on two inertia weights for the improvement of the global and local search of PSO. The results show the two inertia weights are helpful for promoting the search ability in of PSO. In the future, we will test the DWPSO method in more benchmark functions to observe its search ability and propose further process for the DWPSO.

Acknowledgment

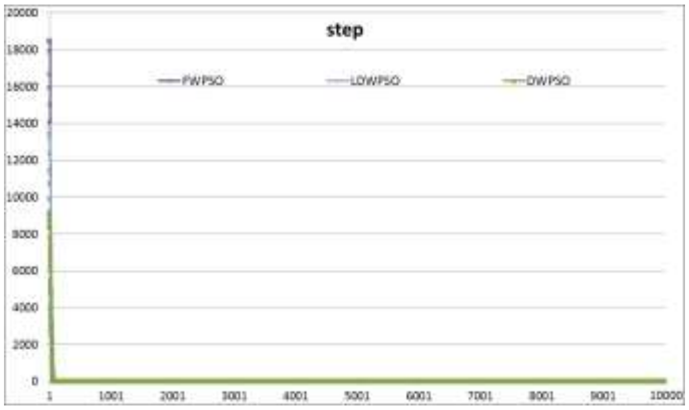
This work is partly supported by the Ministry of Science and Technology (MOST) in Taiwan under grant MOST104-2221-E-464-003.



(a) f_1: Hyperellipsoid chart for iterations of 10000

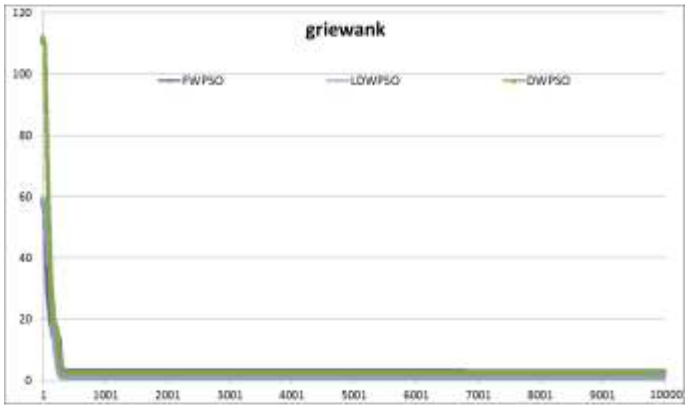


(b) f_2: Quartic chart for iterations of 10000

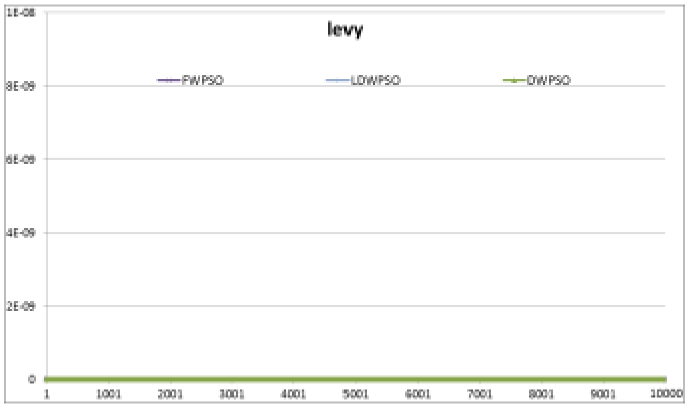


(c) f_3: Step chart for iterations of 10000

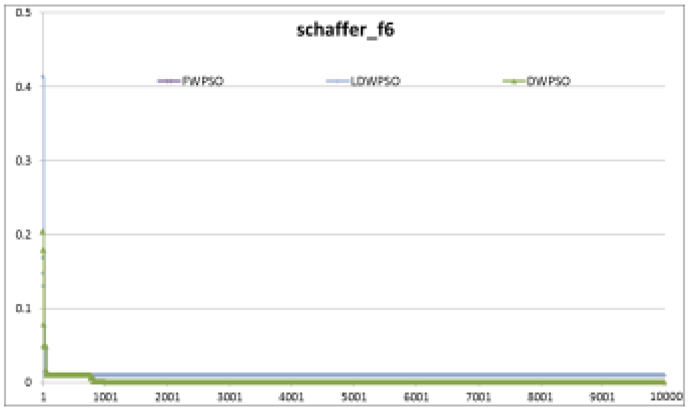
Figure 1. The chart for unimodal functions in $f_1 \sim f_3$. The x-axis is the iteration and the y-axis is the search value.



(a) f_4: Griewank chart for iterations of 10000

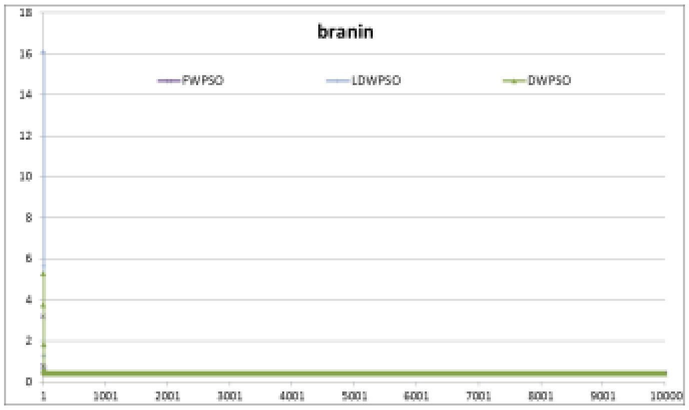


(b) f_5: Levy chart for iterations of 10000

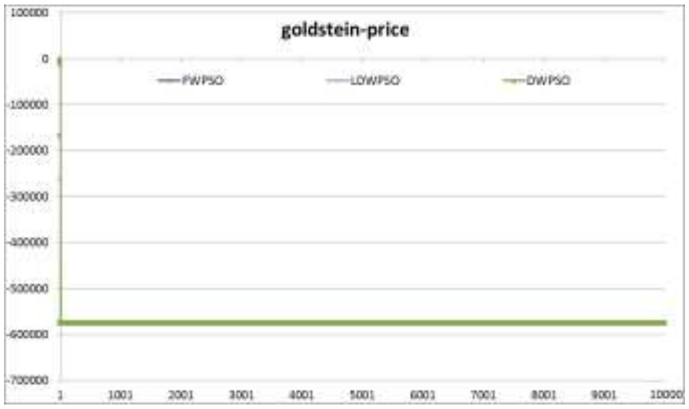


(c) f_6: Schaffer f6 chart for iterations of 10000

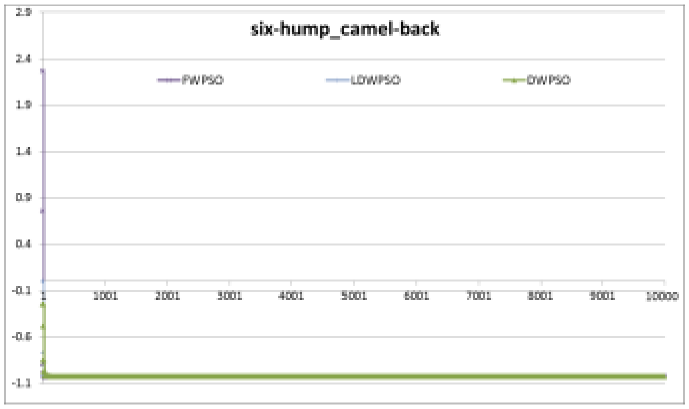
Figure 2. The chart for multimodal functions with many local optima in f_4 ~ f_6. The x-axis is the iteration and the y-axis is the search value.



(a) f_7: Branin chart for iterations of 10000



(b) f_8: Goldstein-price chart for iterations of 10000



(c) f_9: Six-hump camel-back chart for iterations of 10000

Figure 3. The chart for multimodal functions with a few local optima in f_7 ~ f_9. The x-axis is the iteration and the y-axis is the search value.

References

[1] J. Kennedy and R. Eberhart, "Particle swarm optimization," *IEEE International Conference on Neural Networks, 1995. Proceedings.*, vol. 4, pp. 1942-1948, 1995.

[2] A. Subasi, "Classification of EMG signals using PSO optimized SVM for diagnosis of neuromuscular disorders," *Comput Biol Med*, vol. 43, pp. 576-86, Jun 1 2013.

[3] H. H. Chou, L. Y. Hsu, and H. T. Hu, "Turbulent-PSO-Based Fuzzy Image Filter With No-Reference Measures for High-Density Impulse Noise," *IEEE Trans Syst Man Cybern B Cybern*, Jul 20 2012.

[4] Y.-X. Liao, J.-H. She, and M. Wu, "Integrated hybrid-PSO and fuzzy-NN decoupling control for temperature of reheating furnace," *IEEE Transactions on Industrial Electronics*, vol. 56, pp. 2704-2714, 2009.

Proc. of The Third Intl. Conf. On Advances in Applied Science and Environmental Technology - ASET 2015
Copyright © Institute of Research Engineers and Doctors, USA .All rights reserved.
ISBN: 978-1-63248-084-2 doi: 10.15224/ 978-1-63248-084-2-33

[5] C.-N. Ko, Y.-P. Chang, and C.-J. Wu, "A PSO method with nonlinear time-varying evolution for optimal design of harmonic filters," *IEEE Transactions on Power Systems,* vol. 24, pp. 437-444, 2009.

[6] L. Lizzi, F. Viani, R. Azaro, and A. Massa, "A PSO-driven spline-based shaping approach for ultrawideband (UWB) antenna synthesis," *IEEE Transactions on Antennas and Propagation,* vol. 56, pp. 2613-2621, 2008.

[7] B. Liu, L. Wang, and Y. H. Jin, "An effective PSO-based memetic algorithm for flow shop scheduling," *IEEE Trans Syst Man Cybern B Cybern,* vol. 37, pp. 18-27, Feb 2007.

[8] L. Messerschmidt and A. P. Engelbrecht, "Learning to play games using a PSO-based competitive learning approach," *IEEE Transactions on Evolutionary Computation,* vol. 8, pp. 280-288, 2004.

[9] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in *Proceedings of the 1999 Congress on Evolutionary Computation, 1999. CEC 99.,* 1999.

[10] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *The 1998 IEEE International Conference on IEEE World Congress on Computational Intelligence*, 1998, pp. 69-73.

[11] Y. Shi and R. C. Eberhart, "Parameter selection in particle swarm optimization," in *Evolutionary Programming VII*, 1998, pp. 591-600.

About Author (s):

Yu-Huei Cheng received the M.S. degree and Ph.D. degree from the Department of Electronic Engineering, National Kaohsiung University of Applied Sciences, Taiwan, in 2006 and 2010, respectively. This author became the Member (M) of IEEE in 2012; became the Senior Member of Universal Association of Computer and Electronics Engineers (UACEE) in 2015. He crosses many professional fields including biological and medical engineering, electronic engineering, and information engineering. From 2011 to 2015, he acted an assistant professor of the Department of Network Systems, an assistant professor of the Department of Digital Content Design and Management, an assistant professor and the director of the Department of Mobile Technology, and Section Chief of Teaching Resource Section in Office of Academic Affairs, Toko University, Chiayi, Taiwan. He has rich experiences in algorithms design, computer programming, database design and management, and systems programming and design. He has published SCI journal papers more than thirty and presented conference papers more than fifty. His research interests include algorithms, bioinformatics, biomedical engineering, computational biology, computational intelligence, database, data mining, electronic circuit, evolutionary computation, fuzzy systems, information retrieval, machine learning, embedded system and solar energy.

Che-Nan Kuo was born on December 1979 in Tainan, Taiwan. He received his B.S. degree in the Department of Computer Science from the Tunghai University, Taichung, Taiwan in 2002, and the M.S. and Ph.D. degrees from the Department of Computer Science and Information Engineering at the National Cheng Kung University, Tainan, Taiwan in 2004 and 2009. Now, he is an Assistant Professor in the Department of Digital Content Design and Management, Toko University, Chiayi, Taiwan. His current research interests include interconnection networks, discrete mathematics, computation theory, graph theory, and algorithm analysis.

Ching-Ming Lai received the B.S. degree in aeronautical engineering with the honor of the top-rated prize from National Huwei University of Science and Technology, Yunlin, Taiwan, in 2004, the M.S. degree in electrical engineering from National Central University, Chungli, Taiwan, in 2006, and the Ph.D. degree in electrical engineering from National Tsing Hua University, Hsinchu, Taiwan, in 2010. From 2009 to 2012, he served as a Senior R&D Engineer with the Power SBG, Lite-ON Technology Corporation, Taipei Taiwan, where he worked on the high-efficiency and high-power-density ac/dc power supply. In 2012, he established UPE-Power Technology, Company, Ltd., Taichung, Taiwan; the company is developing switching power supplies, and power converters for renewable energy resources. In 2014, he joined the Department of Vehicle Engineering, National Taipei University of Technology, Taipei, Taiwan, where he is currently an Assistant Professor. His research interests include electric vehicles (EVs), power electronics, and high-efficiency energy power conditioning systems. Dr. Lai is a Life Member of the Taiwan Power Electronics Association and a member of the IEEE Power Electronics, IEEE Industry Applications, and IEEE Industrial Electronics Societies. He was the recipient of the Young Author’s Award for Practical Application from the Society of Instrument and Control Engineers, Japan. He received the best paper award at the 2013 IEEE International Conference on Power Electronics and Drive Systems.