SBCAW-PSO: A Sine-Based Chaotic Adaptive Inertia Weight Particle Swarm Optimization

Yu-Huei Cheng, *Member, IAENG*, Che-Nan Kuo

***Abstract*—Many strategies for adapting inertia weight of the particle swarm optimization (PSO) have been proposed. The inertia weight of PSO plays a crucial role in the ability of exploration and exploitation. In this paper, we propose a sine-based chaotic map and use it to chaotically adapt inertia weight in PSO. We call this method as SBCAW-PSO. Ten unimodal and multimodal benchmark functions are used to evaluate the SBCAW-PSO. The experimental results show that using the sine-based chaotic map can effectively adapt inertia weight of PSO to improve the search results.**

***Index Terms*—Chaos, inertia weight, particle swarm optimization (PSO), sine-based chaotic map**

I. Introduction

PARTICLE swarm optimization (PSO) is a popular population-based algorithm [1]. Many real-world applications have been effectively solved by PSO, 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, there are many variants of the PSO are proposed and most of them depend on the inertia weight for adjust their search ability with exploration and exploitation. The inertia weight is able to balance the global and local search. A large inertia weight facilitates a global search while a small inertia weight facilitates a local search [9].

In the original PSO, no inertia weight is embedded in PSO. The fixed inertia weight PSO [10] (here we called it as "FW-PSO") 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. However, the fixed inertia weight PSO is not very effective for tracking nonlinear dynamic systems most real-world applications. Therefore, we proposed a novel chaotic inertia weight PSO for tracking nonlinear dynamic systems. It tends to automatic control the global and local search ability according to the chaotic value. The method have been tested and compared with the FW-PSO on unimodal and multimodal benchmark functions.

Manuscript received December 6, 2013. This work is partly supported by the National Science Council (NSC) in Taiwan under grant NSC102-2221-E-464-004-.

Yu-Huei Cheng is with the Department of Digital Content Design and Management, Toko University, Chiayi, Taiwan (corresponding author to provide phone: +886-9-19832183; e-mail: yuhuei.cheng@gmail.com).

Che-Nan Kuo is with the Department of Digital Content Design and Management, Toko University, Chiayi, Taiwan (e-mail: fkikimo@hotmail.com).

II. Methods

A. Particle swarm optimizations

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 based on 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. After that, these particles are 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 $X_i = (x_{i1}, x_{i2}, \ldots, x_{iN})$. The personal best position of the ith particle is represented as $pbest_i = (p_{i1}, p_{i2}, \ldots, p_{iN})$. The global best position found from all the particles is represented as $gbest = (g_1, g_2, \ldots, g_N)$. The velocity of the ith is represented as $V_i = (v_{i1}, v_{i2}, \ldots, v_{iN})$. 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

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

where g is the current generation; X is the position of the particle.

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

The global best position is found by

$$gbest(g+1) = \min(\forall pbest(g+1)) \quad (2)$$

where $\min(\forall X_i)$ represents the function for get the minimum X_i.

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

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

where g is the current generation; c_1 and c_2 denote the acceleration coefficients; r_1 and r_2 are the uniform random values in the range (0, 1).

The current position of each particle is updated using

$$X_i(g+1) = X_i(g) + V_i(g+1) \quad (4)$$

(b) Fixed inertia weight PSO (FW-PSO)

The original PSO introduces a parameter called inertia weight (w) to the updating equation of the velocity [10]. The influence for using fixed inertia weight has been estimated for the performance of PSO. 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 "FW-PSO" to represent the fixed inertia weight in this paper. The new updating equation of the velocity is given as follows:

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

where g is the current generation; w is inertia weight; c_1 and c_2 are the acceleration coefficients; r_1 and r_2 are the uniform random values in the range (0, 1); V is the velocity of the particle; X is the position of the particle.

(c) The sine-based chaotic adaptive inertia weight PSO (SBCAW-PSO)

In this paper, we proposed a chaotic map. The chaotic map use sine function to get the chaotic behaviours. The chaotic map is shown in eq. (6).

$$X(n+1) = \begin{cases} [\sin(X(n) \times 4\pi) + 1]/4, n \le N/d \\ [\sin(X(n) \times 4\pi) + 1]/4 + 0.5, \text{otherwise} \end{cases} \quad (6)$$

where n is the current number of iteration; sin is the sine trigonometric function; π is a mathematical constant with the ratio of a circle's circumference to its diameter (approximately equal to 3.14159); N is the total iterations; d is division number of the total iterations.

We use the sine-based chaotic map to update the inertia weight value of the PSO. The inertia weight is changed by generation updating and is calculated as

$$w(g+1) = \begin{cases} [\sin(w(g) \times 4\pi) + 1]/4, g \le G/d, \\ \qquad d = 2 \text{ or } 4, w(g) \in (0,1) \\ [\sin(w(g) \times 4\pi) + 1]/4 + 0.5, \text{otherwise} \end{cases} \quad (7)$$

where w is inertia weight; g is the current generation; G is the total iterations; d is division number of the total generation, here we set d = 2 or d = 4.

TABLE I
MODALITY, GLOBAL OPTIMUM, SEARCH SPACE AND INITIAL RANGES OF THE 10 TEST FUNCTIONS

Modality	Function f	Global optimum	Search space	Initial range
unimodal	f_1	0	$[-1.0, 1.0]^N$	$[-1.0, 1.0]^N$
	f_2	0	$[-2.048, 2.048]^N$	$[-2.048, 2.048]^N$
Multimodal with many local optimums	f_3	0	$[-32.768, 32.768]^N$	$[-32.768, 16.0]^N$
	f_4	0	$[-10.0, 10.0]^N$	$[-10.0, 10.0]^N$
	f_5	0	$[-10.0, 10.0]^N$	$[-10.0, 10.0]^N$
	f_6	0	$[-5.12, 5.12]^N$	$[-5.12, 5.12]^N$
	f_7	0	$[-0.5, 0.5]^N$	$[-0.5, 0.5]^N$
Multimodal with a few local optimums	f_8	0	$[-5.0, 15.0]^N$	$[-5.0, 15.0]^N$
	f_9	0	$[-5.0, 5.0]^N$	$[-5.0, 5.0]^N$
	f_{10}	0	$[-2.0, 2.0]^N$	$[-2.0, 2.0]^N$

N is the size of dimensions.

B. Benchmark functions

Ten benchmark functions [11-14] including unimodal and multimodal problems was used for evaluating the FW-PSO and SBCAW-PSO. Table 1 lists the 10 benchmark functions and their modality, global optimum, search space and initial ranges. These functions are divided into three parts, i.e., unimodal functions, multimodal functions with many local optimums, and multimodal functions with a few local optimums. They are described as follows.

(a) Unimodal functions

Unimodal functions are rather easy to optimize. However, when the problem dimensions are high, the optimization will become difficult. In this study, f_1 and f_2 are unimodal functions, they are listed below.

1) Hyperellipsoid

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

2) Rosenbrock variant function

$$f_2(x_1, x_2) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2 \quad (9)$$

C. Multimodal functions with many local optimums

Functions $f_3 \sim f_7$ are multimodal functions with many local optimums. They look to be the most difficult category of problems for many optimization methods. These functions are listed as follows.

1) Ackley's function

$$\begin{aligned} f_3(x) = -20\exp\left(-0.2\sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}\right) \\ -\exp\left(\frac{1}{N}\sum_{i=1}^{N}\cos(2\pi x_i)\right) + 20 + e \end{aligned} \quad (10)$$

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

2) Colville

$$\begin{aligned} f_4(x_1,x_2,x_3,x_4) &= 100(x_2 - x_1^2)^2 + (1-x_1)^2 \\ &+ 90(x_4 - x_3^2)^2 + (1-x_3)^2 \\ &+ 10.1((x_2-1)^2 + (x_4-1)^2) \\ &+ 19.8(x_2-1)(x_4-1) \end{aligned} \tag{11}$$

3) Levy

$$\begin{aligned} f_5(x) &= \frac{\pi}{N}\Big(k\sin^2(\pi\, y_1) \\ &+ \sum_{i=1}^{N-1}\Big[(y_i-a)^2(1+k\sin^2(\pi\, y_{i+1}))\Big] \\ &+ (y_n-a)^2\Big) \\ y_i &= 1+\frac{1}{4}(x_i-1),\ k=10,\ a=1 \end{aligned} \tag{12}$$

4) Rastrigin's function

$$f_6(x) = \sum_{i=1}^{N}(x_i^2 - 10\cos(2\pi x_i) + 10) \tag{13}$$

5) Rastrigin's function

$$\begin{aligned} f_7(x) &= \sum_{i=1}^{N}\left(\sum_{k=0}^{k\max}\Big[a^k\cos\Big(2\pi b^k(x_i+0.5)\Big)\Big]\right) \\ &- N\sum_{k=0}^{k\max}\Big[a^k\cos\Big(2\pi b^k\cdot 0.5\Big)\Big], \\ a &= 0.5,\ b=3,\ k\max = 20. \end{aligned} \tag{14}$$

D.Multimodal functions with a few local optimums

Functions $f_8 \sim f_{10}$ are as well multimodal functions, but they only comprise a few local optimums. They are listed as follows.

1) Rastrigin's function

$$\begin{aligned} f_8(x_1,x_2) &= (x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6)^2 \\ &+ 10(1-\frac{1}{8\pi})\cos x_1 + 10 \end{aligned} \tag{15}$$

2) Rastrigin's function

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

3) Rastrigin's function

$$\begin{aligned} f_{10}(x_1,x_2) &= \Big[1+(x_1+x_2+1)^2(19-14x_1+3x_1^2-14x_2+6x_1x_2+3x_2^2)\Big] \\ &\times\Big[30+(2x_1-3x_2)^2(18-32x_1+12x_1^2+48x_2-36x_1x_2+27x_2^2)\Big] \end{aligned} \tag{17}$$

III. Results

The experiments compared the SBCAW-PSO with the FW-PSO on the 10 benchmark functions with 10 dimensions, i.e., N is set to 10. The two PSOs were implemented using 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 PSOs, we use bound terminal, 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 two methods, i,e., the number of iterations (30000), the particle swarm size (10), the inertia weight w (based on different adaptive approaches), and the constriction factors c_1 and c_2 (2 and 2). Each method was run 30 times and their mean values, standard deviation, and average running time for the results were calculated.

B. Experimental results

Table 2 presents the mean, standard deviation, and average running time of 30 runs for the FW-PSO and SBCAW-PSO on the 10 benchmark functions with 10 dimensions. The best results are shown in bold fonts.

In this unimodal functions for f_1, SBCAW-PSO performed the best mean, variation, and standard deviation when d is set to 4. The SBCAW-PSO performed the shortest run time and the better mean, standard deviation, and average running time than FA-PSO when d is set to 2. In this unimodal functions for f_2, SBCAW-PSO performed the best mean, variation, standard deviation, and average running time when d is set to 4. In this multimodal functions with many local optimums, SBCAW-PSO performed the best mean, variation, and standard deviation for f_4, f_5 and f_6 except the mean in f_5 when d is set to 2. The SBCAW-PSO performed the best mean, variation, and standard deviation for f_3 and f_7 when d is set to 4. In this multimodal functions with a few local optimums, SBCAW-PSO performed the best mean, variation, and standard deviation for all functions f_8, f_9, and f_{10}. Although the FA-PSO performed shortest run time, the mean, variation, and standard deviation are the worst. From the above results, we get the SBCAW-PSO performed better results than FW-PSO on the 10 benchmark functions for unimodal and multimodal problems with many local optimums and with a few local optimums.

In this paper, the sine-based chaotic map can be seen as an algorithmic component that provides an improved performance. It changes inertia weight according to the chaotic adaption by iterations and therefore there are more opportunities to find out optimal solution.

IV. Conclusion

In this paper, we propose SBCAW-PSO that uses a novel sine-based chaotic map to adapt the inertia weight of PSO. Ten unimodal and multimodal benchmark functions are used to evaluate the SBCAW-PSO. The experimental results show that using the sine-based chaotic map can effectively adapt inertia weight of PSO to improve the search results.

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.

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

[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.
[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] C.-Y. Lee and X. Yao, "Evolutionary programming using mutations based on the Lévy probability distribution," *IEEE Transactions on Evolutionary Computation*, vol. 8, pp. 1-13, 2004.
[12] Z. Tu and Y. Lu, "A robust stochastic genetic algorithm (StGA) for global numerical optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, pp. 456-470, 2004.
[13] J. J. Liang, A. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, pp. 281-295, 2006.
[14] S. T. Hsieh, T. Y. Sun, C. C. Liu, and S. J. Tsai, "Efficient population utilization strategy for particle swarm optimizer," *IEEE Trans Syst Man Cybern B Cybern*, vol. 39, pp. 444-56, Apr 2009.

TABLE II
SEARCH RESULT COMPARISONS FOR THE MEAN, VARIATION, STANDARD DEVIATION, AND AVERAGE RUN TIME OF 30 RUNS FOR THE FW-PSO AND SBCAW-PSO ON THE 10 TEST FUNCTIONS WITH 10 DIMENSIONS

Modality	Function f	Results	FW-PSO	SBCAW-PSO (d=2)	SBCAW-PSO (d=4)
unimodal	f_1	Mean	1.24E-001	1.26E-002	**9.96E-003**
		Var.	6.69E-003	1.63E-004	**1.28E-004**
		Std. dev.	8.18E-002	1.28E-002	**1.13E-002**
		Avg. run time (ms)	966	**830**	944
	f_2	Mean	2.00E+000	**0**	**0**
		Var.	4.27E-001	**0**	**0**
		Std. dev.	6.53E-001	**0**	**0**
		Avg. run time (ms)	1251	231	**190**
Multimodal with many local optimums	f_3	Mean	6.78E+001	4.60E-002	**4.08E-002**
		Var.	1.33E+005	1.27E-002	**6.27E-003**
		Std. dev.	3.65E+002	1.13E-001	**7.92E-002**
		Avg. run time (ms)	1020	**924**	1031
	f_4	Mean	3.20E+000	**1.75E+000**	1.83E+000
		Var.	1.06E+000	**7.67E-002**	2.32E-001
		Std. dev.	1.03E+000	**2.77E-001**	4.81E-001
		Avg. run time (ms)	1534	**1425**	1547
	f_5	Mean	3.93E+001	5.53E+000	**4.98E+000**
		Var.	7.56E+002	**3.09E+000**	3.73E+000
		Std. dev.	2.75E+001	**1.76E+000**	1.93E+000
		Avg. run time (ms)	1136	**1060**	1164
	f_6	Mean	1.24E-005	**4.58E-026**	1.41E-023
		Var.	3.44E-010	**5.92E-050**	5.93E-045
		Std. dev.	1.85E-005	**2.43E-025**	7.70E-023
		Avg. run time (ms)	962	**565**	712
	f_7	Mean	2.12E-001	7.69E-010	**2.16E-010**
		Var.	3.97E-003	2.91E-018	**4.84E-019**
		Std. dev.	6.30E-002	1.71E-009	**6.96E-010**
		Avg. run time (ms)	**999**	1258	1418
Multimodal with a few local optimums	f_8	Mean	1.28E-003	3.32E-018	**1.05E-019**
		Var.	7.03E-007	1.85E-034	**1.17E-037**
		Std. dev.	8.38E-004	1.36E-017	**3.42E-019**
		Avg. run time (ms)	**1330**	1502	1684
	f_9	Mean	7.60E+000	1.51E-001	**9.09E-003**
		Var.	7.39E-001	5.30E-001	**1.79E-004**
		Std. dev.	8.59E-001	7.28E-001	**1.34E-002**
		Avg. run time (ms)	**30380**	31477	31725
	f_{10}	Mean	4.12E+000	1.82E-006	**2.60E-007**
		Var.	4.05E+002	6.38E-011	**8.27E-013**
		Std. dev.	2.01E+001	7.99E-006	**9.10E-007**
		Avg. run time (ms)	**1077**	1215	1357

ISBN: 978-988-19252-5-1
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)