Proceedings of the International MultiConference of Engineers and Computer Scientists 2015 Vol I,
IMECS 2015, March 18 - 20, 2015, Hong Kong

Evaluation of Sine-Based Chaotic Strategy for Adapting Inertia Weight of Particle Swarm Optimization

Yu-Huei Cheng, *Member, IAENG*

***Abstract*—Particle swarm optimizations (PSOs) have been applied to many fields. In PSOs, the inertia weight is an important parameter for performing global search and local search in solution search. In this paper, we use a sine-based chaotic map to chaotically adapt inertia weight of PSO based on different shift deviations (*sd*) to perform five multimodal benchmark functions with many local optima. The experimental results show that the method using *sd* 0.3-0.5 can get better results for multimodal benchmark functions with many local optima. Furthermore, the method is superior to PSO with inertia weight. This study is useful to help us to set appropriate shift deviation for multimodal benchmark functions with many local optima when using sine-based chaotic map to adapt inertia weight of PSO.**

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

I. Introduction

PARTICLE swarm optimization (PSO) was proposed by Eberhart and Kennedy in 1995 [1]. Many variants of the PSO had been proposed and applied to real-world applications, for examples, learning to play games [2], flow shop scheduling [3], Ultrawideband (UWB) Antenna Synthesis [4], harmonic filters [5], decoupling control for temperature of reheating furnace [6], image filter [7], EMG (electromyogram) signal classification [8], and so on.

In PSOs, the inertia weight plays a crucial role to determine the search ability for exploration and exploitation. A large inertia weight facilitates a global search while a small inertia weight facilitates a local search [9]. Therefore, there are many variants of the PSO are proposed which depend on the inertia weight.

In the original PSO, none of inertia weight is embedded in PSO [1] and thus it performed worse efficiency. The inertia weight was first introduced by Shi and Eberhart in 1998 [10]. They performed different chosen inertia weight to illustrate its impact on the performance of PSO. Although it improved the efficiency of the original PSO, the constant inertia weight is incapable of balancing the ability between global search and local search. Furthermore, the constant inertia weight PSO is also ineffective for tracking nonlinear dynamic systems in most real-world applications. In 2014, we propose SBCAW-PSO [11] which uses a sine-based chaotic strategy to chaotically adapt the inertia weight of PSO. In this study, we performed different chosen shift deviations (*sd*) to adjust sine-based chaotic strategy. Five multimodal benchmark functions with many local optima are used to test the performance. It tends to automatic control the global and local search ability according to the chaotic value.

II. Materials and methods

A. The sine-based chaotic map with shift deviation

In this paper, we proposed a chaotic map with shift deviation. The chaotic map use sine function to get the chaotic behaviours. It can exhibit aperiodic behavior that depends sensitively on the initial conditions and rendering long-term prediction impossible. This is helpful for tracking nonlinear dynamic systems. The chaotic map is shown in the following:

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

where *sd* is the shift deviation; *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.

B. Particle swarm optimization

The social behaviors of a bird flock or fish school inspire the PSO. In the PSO algorithm, a population of random solutions is first generated. We call the population as "particles". In these particles, each particle has its own velocity and position. In each generation, all particles are evaluated by an objective function. These particles are then compared with the previous positions to gain the personal best positions, and compared with each other to gain the global best position. The current velocities are also updated according to the previous positions, the personal best positions and the global best position. After that, each particle moves to a new position according to its current velocity and its previous position. The optimal solution is thus searched by generation to generation based on the update equations of the velocities and the positions of particles.

Manuscript received January 6, 2015; revised January 17, 2015. This work is partly supported by the Ministry of Science and Technology (MOST) in Taiwan under grant MOST101-2221-E-464-001-, MOST102-2221-E-464-004-, and MOST103-2221-E-464-004-.

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

1) Original PSO

Eberhart and Kennedy proposed the original PSO in 1995 [1]. We consider N is dimensions for an optimization problem for search space, and the characteristics of the original PSO 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.

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

$$pbest_i(g+1) = \begin{cases} pbest_i(g), \\ \quad \text{if objective}(X_i(g+1)) \geq \text{objective}(X_i(g)) \\ X_i(g+1), \text{otherwise} \end{cases} \quad (2)$$

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

The global best position is found by

$$gbest(g+1) = \min(\forall pbest(g+1)) \quad (3)$$

where $\min(\forall pbest(g+1))$ represents the function for get the minimum $pbest$(g+1).

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

$$\begin{aligned} V_i(g+1) = V_i(g) + c_1 \times r_{1i}(g) \times [pbest_i(g) - X_i(g)] \\ + c_2 \times r_{2i}(g) \times [gbest(g) - X_i(g)] \end{aligned} \quad (4)$$

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 of between 0 and 1; V is the velocity of the particle; X is the position of the particle.

The current position of each particle is updated by

$$X_i(g+1) = X_i(g) + V_i(g+1) \quad (5)$$

2) PSO with inertia weight

The original PSO is not effective for solution search. Therefore, Shi and Eberhart introduce a parameter called inertia weight (w) to the velocity updating equation [10]. They had estimated the influence for using inertia weight for the performance of PSO. The chosen inertia weight range [0.9, 1.2] was considered to be a good area [10]. The velocity updating equation for the PSO with inertia weight is given as follows:

$$\begin{aligned} V_i(g+1) = w \times V_i(g) + c_1 \times r_{1i}(g) \times [pbest_i(g) - X_i(g)] \\ + c_2 \times r_{2i}(g) \times [gbest(g) - X_i(g)] \end{aligned} \quad (6)$$

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 of between 0 and 1; V is the velocity of the particle; X is the position of the particle.

TABLE I
GLOBAL OPTIMUM, SEARCH SPACE AND INITIAL RANGES OF THE FIVE BENCHMARK FUNCTIONS.

Function f	Global optimum	Search space	Initial range
f_1	20	$[-32.768, 32.768]^N$	$[-32.768, 16.0]^N$
f_2	0	$[-10.0, 10.0]^N$	$[-10.0, 10.0]^N$
f_3	0	$[-100.0, 100.0]^N$	$[-100.0, 50.0]^N$
f_4	0	$[-1.0, 1.0]^N$	$[-1.0, 1.0]^N$
f_5	0	$[-0.5, 0.5]^N$	$[-0.5, 0.5]^N$

N is the size of dimensions.

3) The sine-based chaotic map with shift deviation chaotically adaptive inertia weight PSO

The original PSO and PSO with inertia weight are always very ineffective for tracking nonlinear dynamic systems in most real-world applications. Therefore, we use the sine-based chaotic map with shift deviation to chaotically update the inertia weight value of the PSO. The inertia weight is changed by generation updating and is calculated as follows:

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

where w is inertia weight; g is the current generation; G is the total iterations.

C. Benchmark functions

Five benchmark functions [12-15] for multimodal problems with many local optimums was used for evaluating the proposed method. They look to be the most difficult category of problems for many optimization methods. Table I lists their global optimum, search space and initial ranges. The five benchmark functions are shown as follows:

1) Ackley's function

$$\begin{aligned} f_1(x) = -20\exp\left(-0.2\sqrt{\frac{1}{N}\sum_{i=1}^{N} x_i^2}\right) \\ -\exp\left(\frac{1}{N}\sum_{i=1}^{N}\cos(2\pi x_i)\right) + 20 + e \end{aligned} \quad (8)$$

2) Colville

$$\begin{aligned} f_2(x_1, x_2, x_3, x_4) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 \\ + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 \\ + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) \\ + 19.8(x_2 - 1)(x_4 - 1) \end{aligned} \quad (9)$$

3) Schaffer f6

$$f_3(x) = 0.5 + \frac{(\sin\sqrt{x_1^2 + x_2^2})^2 - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2} \quad (10)$$

4) Sum of different powers

$$f_4(x) = \sum_{i=1}^{N} |x_i|^{i+1} \tag{11}$$

5) Weierstrass

$$f_5(x) = \sum_{i=1}^{N} \left(\sum_{k=0}^{k\,\max} \left[a^k \cos\left(2\pi b^k (x_i + 0.5)\right) \right] \right) - N \sum_{k=0}^{k\,\max} \left[a^k \cos\left(2\pi b^k \cdot 0.5\right) \right], \tag{12}$$

$a = 0.5$, $b = 3$, k max = 20.

III. Results and Discussion

The experiments used five different *sd* values to test the sine-based chaotic map with shift deviation adaptive inertia weight PSO. Furthermore, the method is also compared to PSO with inertia weight method. The dimension of these benchmark functions is set to 10, i.e., N is 10. We implement the algorithm using JAVA for cross platform applications. The experiments were computed on Pentium 4 CPU 3.4 GHz with 1GB of RAM on Microsoft Windows XP SP3 professional operating system. For the boundary process, we perform 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 proposed method, i,e., the number of iterations (10000), the particle swarm size (10), the inertia weight w (based on *sd* from 0.0 to 0.5 with the sine-based chaotically adaption), and the constriction factors c_1 and c_2 (2 and 2). We run 30 times to calculate the mean value, variation, standard deviation, and average running time.

B. Experimental results

1) The results based on five different *sd* values for the sine-based chaotic map with shift deviation adaptive inertia weight PSO

Table II presents the mean, variation, standard deviation, and average running time of 30 runs for different *sd* on the five benchmark functions with 10 dimensions. The best results are shown in bold fonts.

For f_1, the method performed the optimal mean when s*d* was set from 0.0 to 0.5. When s*d* was set to 0.1, the method performed the best variation, and standard deviation; the average run time are slightly worse than that which when s*d* was set to 0.0. For f_2 and f_3, the method performed the best mean, variation, and standard deviation when s*d* was set to 0.3, 0.4, and 0.5. Furthermore, the least average run time was spent when the s*d* was set to 0.3. For f_4, the method performed the best mean, variation, and standard deviation when s*d* was set to 0.3; the method performed the second mean, variation, and standard deviation when s*d* was set to 0.2. The shortest average run time is fall on that when s*d* was set to 0.5. Finally, for f_5, the method performed all the best mean, variation, standard deviation, and average run time when s*d* was set to 0.4.

2) Comparison of the method to PSO with inertia weight

Here, we compare the sine-based chaotic map with shift deviation adaptive inertia weight PSO (*sd* is set to 0.3) to PSO with inertia weight method. The comparing result is shown in Table III. From the result, we get all the result of the proposed method in mean, variation, standard deviation, and average run time are superior to the PSO with inertia weight method except the average run time in f_4.

C. Discussion

From the above results, we find s*d* is set to 0.3-0.5 has the better results in the five benchmark functions except the f_1. Furthermore, the result also shows the proposed method better than the PSO with inertia weight method. The sine-based chaotic map with shift deviation can be seen as an algorithmic component that provides an improved performance of PSO. It changes inertia weight according to the chaotic adaption by iterations and give assigned inertia weight to guarantee minimum inertia weight value. Therefore, it will useful for specific problem to give more opportunities to find out optimal solution.

IV. Conclusion

A sine-based chaotic map chaotically adapts inertia weight of PSO based on different shift deviations (*sd*) is useful for improving the performance of PSO. The experimental results of the five multimodal benchmark functions with many local optima show that the method using *sd* 0.3-0.5 can get better results. Also, the method is superior to PSO with inertia weight when comparing their results. The shift deviation is useful to help us to improve the performance for specific problems with many local optima when using sine-based chaotic map to adapt inertia weight of PSO.

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] 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.

[3] 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.

[4] 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.

[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] 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.

[7] 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.

[8] 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.

[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.-H. Cheng and C.-N. Kuo, "SBCAW-PSO: A Sine-Based Chaotic Adaptive Inertia Weight Particle Swarm Optimization," in *Proceedings*

ISBN: 978-988-19253-2-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

of the International MultiConference of Engineers and Computer Scientists, 2014.

[12] 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.

[13] 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.

[14] 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.

[15] 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

Function f	*sd* / Results	**0.0**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**
f_1	Mean	**2.00E+01**	**2.00E+01**	**2.00E+01**	**2.00E+01**	**2.00E+01**	**2.00E+01**
	Var.	4.22E-09	**2.79E-11**	7.92E-07	1.78E-03	1.78E-03	3.16E-03
	Std. dev.	6.49E-05	**5.28E-06**	8.90E-04	4.21E-02	4.21E-02	5.62E-02
	Avg. run time (ms)	**1180**	1183	1290	1187	1188	1194
f_2	Mean	1.26E-02	3.29E-02	1.98E-04	**1.91E-05**	3.98E-04	9.02E-03
	Var.	1.63E-04	3.04E-03	2.58E-07	**2.25E-09**	1.70E-07	7.09E-05
	Std. dev.	1.28E-02	5.51E-02	5.07E-04	**4.75E-05**	4.12E-04	8.42E-03
	Avg. run time (ms)	830	854	926	**804**	808	863
f_3	Mean	3.63E-02	5.72E-03	1.30E-03	**0**	**0**	**0**
	Var.	4.87E-03	9.43E-05	1.13E-05	**0**	**0**	**0**
	Std. dev.	6.98E-02	9.71E-03	3.36E-03	**0**	**0**	**0**
	Avg. run time (ms)	1253	1089	976	**802**	813	869
f_4	Mean	3.32E-18	1.24E-14	2.63E-187	**3.07E-215**	5.07E-90	5.16E-14
	Var.	1.85E-34	4.55E-27	**0**	**0**	7.70E-178	1.58E-26
	Std. dev.	1.36E-17	6.74E-14	**0**	**0**	2.78E-89	1.26E-13
	Avg. run time (ms)	1502	1454	1739	1469	1398	**1248**
f_5	Mean	1.51E-01	4.94E-01	3.35E-01	1.68E-01	**0**	2.29E-01
	Var.	5.30E-01	6.07E-01	3.24E-01	3.08E-01	**0**	5.44E-01
	Std. dev.	7.28E-01	7.79E-01	5.69E-01	5.55E-01	**0**	7.37E-01
	Avg. run time (ms)	31477	31145	30481	25107	**21619**	30220
f_{10}	Mean	1.26E-02	3.29E-02	1.98E-04	**1.91E-05**	3.98E-04	9.02E-03
	Var.	1.63E-04	3.04E-03	2.58E-07	**2.25E-09**	1.70E-07	7.09E-05
	Std. dev.	1.28E-02	5.51E-02	5.07E-04	**4.75E-05**	4.12E-04	8.42E-03
	Avg. run time (ms)	830	854	926	**804**	808	863

ISBN: 978-988-19253-2-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE III

COMPARISON OF THE SINE-BASED CHAOTIC MAP WITH SHIFT DEVIATION ADAPTIVE INERTIA WEIGHT PSO AND PSO WITH INERTIA WEIGHT METHOD.

Function f	*sd* / Results	**PSO with inertia weight**	**Sine-based chaotic map with shift deviation adaptive inertia weight PSO (*sd* = 3)**
f_1	Mean	2.03E+01	**2.00E+01**
	Var.	4.80E-03	**1.78E-03**
	Std. dev.	6.93E-02	**4.21E-02**
	Avg. run time (ms)	1281	**1187**
f_2	Mean	1.24E-01	**1.91E-05**
	Var.	6.69E-03	**2.25E-09**
	Std. dev.	8.18E-02	**4.75E-05**
	Avg. run time (ms)	966	**804**
f_3	Mean	3.89E-03	**0**
	Var.	2.34E-05	**0**
	Std. dev.	4.84E-03	**0**
	Avg. run time (ms)	1682	**802**
f_4	Mean	1.28E-03	**3.07E-215**
	Var.	7.03E-07	**0**
	Std. dev.	8.38E-04	**0**
	Avg. run time (ms)	1330	1469
f_5	Mean	7.60E+00	**1.68E-01**
	Var.	7.39E-01	**3.08E-01**
	Std. dev.	8.59E-01	**5.55E-01**
	Avg. run time (ms)	30380	**25107**
f_{10}	Mean	2.03E+01	**2.00E+01**
	Var.	4.80E-03	**1.78E-03**
	Std. dev.	6.93E-02	**4.21E-02**
	Avg. run time (ms)	1281	**1187**

ISBN: 978-988-19253-2-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)