Particle Swarm Optimization with Adaptive Inertia Weight

Sameh Kessentini and Dominique Barchiesi

***Abstract*—In this paper, a new PSO algorithm with adaptive inertia weight is introduced for global optimization. The objective of the study is to balance local search and global search abilities and alternate them through the algorithm progress. For this, an adaptive inertia weight is introduced using a feedback on particles' best positions. The inertia weight keeps varying to alternate exploration and exploitation. Tests are carried on a set of thirty test functions (the CEC 2014 benchmark functions) and compared with other settings of inertia weight. Results show that the new algorithm is very competitive mainly when increasing the dimension of the search space.**

***Index Terms*—Algorithms, exploration and exploitation, inertia weight, particle swarm optimization.**

I. Introduction

Particle swarm optimization (PSO) was first introduced by Kennedy and Eberhart in 1995 [1] and imitates the swarm behavior to search the globally best solution. In this method, particles of the swarm move in a multidimensional search space looking for a potential solution. When moving, each particle is guided by its own experience and collaboration with neighbor swarm particles. This technique attracted a high level of interest because of its simplicity and its encouraging results in many fields.

The basic PSO [1] is not the best tool to solve all engineering problems as it is slow in some cases and converges to local optima in some others (e.g. in the field of plasmonics [2]). To improve the PSO performance, different variants of the algorithm were developed with the main objective of balanced exploration-exploitation [3]-[6].

The inertia weight, introduced in 1998 [7], plays a key role in the PSO process, because it is a crucial tool to balance the exploration and exploitation. We introduce a new setting of this parameter. The inertia weight is dynamically adjusted using a feedback on the particles' best positions to *alternate* exploration and exploitation during the algorithm process. Our algorithm is compared with other settings of inertia weight -based on previous comparative studies- that are GPSO [3], Sugeno [4], APSO [5], and AIWPSO [8]. The tests are carried using the CEC 2014 benchmark functions [9] and show a great potential of the new setting.

The remainder of the paper is organized as follows. Section II provides an overview of the PSO and related work. In Section III, the new algorithm is fully described. Section IV presents the simulation results and their discussion. Finally, we conclude in Section V with a brief discussion and a summary of results.

Manuscript received December 26, 2014; revised April 22, 2015.

S. Kessentini is with the Department of Mathematics, Faculty of Science of Sfax, University of Sfax, Route de Soukra km 4-BP 802, 3038 Sfax, Tunisia (e-mail: samehkessentini@gmail.com).

D. Barchiesi is with the Project Group for Automatic Mesh Generation and Advanced Methods - Gamma3 Project (UTT-INRIA), University of Technology of Troyes, 12 rue Marie Curie - BP 2060, 10010, Troyes Cedex, France (e-mail: dominique.barchiesi@utt.fr).

II. Background

The PSO is basically a cooperative method where, at step t, the vector of decision variables, N being the number of particles and D being the search space dimension, is considered as an i^{th} particle in motion during the algorithm. Each position $x_i(t)$ represents a potential solution of the optimization problem. Then, the particles of the swarm communicate good positions to each other and adjust their own positions and velocities $V_i(t) = \left(V_i^j(t)\right)_{1\le j\le D}$ at each step following

$$V_i^j(t+1) = \omega V_i^j(t) + r_1^j c_1\left(p_i^j(t) - x_i^j(t)\right) + r_2^j c_2\left(g^j(t) - x_i^j(t)\right), \quad (1)$$

$$x_i^j(t+1) = x_i^j(t) + V_i^j(t+1), \quad (2)$$

where r_1^j and r_2^j are independent uniform random variables generated between 0 and 1, $p_i(t)$ is the best position of particle i i.e. its best experience, $g(t)$ is the global best of the swarm, ω is the inertia weight, and c_1 and c_2 are the acceleration coefficients. Equation (1) is used to evaluate the particle new velocity using its previous one, the distances between its current position and its best position, and the distance between its current position and the global best. Equation (2) is used to update the position of the particle using its previous position and its new velocity.

The success of PSO depends on values taken by the inertia weight that was introduced by Shi and Eberhart in 1998 [7]. Without the first term of (1), the search will be reduced to a local search. If the inertia weight takes large values (other terms of this equation are almost omitted), the algorithm keeps exploring new spaces and then the convergence is delayed. Therefore, the inertia weight must be adjusted for a better exploration-exploitation trade-off.

A large number of inertia weight settings were proposed. These approaches can be classified in four main groups: constant [7], random [10], time varying, and adaptive inertia weights. The most famous time varying law may be the linear decreasing of inertia weight [3]. Different other time varying laws were used such as sigmoid [11], simulated annealing [12], Sugeno function [4], exponential decreasing law [13], [14], and logarithmic decreasing law [15]. Then, the adaptive approaches were introduced with motivation a better control

of the population diversity by adaptive adjustment of the inertia weight using feedbacks of the process (e.g. the best fitness achieved [16], the number of updated best positions [8], or the distance between particles [5]).

Many comparative studies were conducted to benchmark different settings of inertia weight. Bansal *et al.* [17] compared a set of fifteen relatively recent and popular inertia weight strategies and found that constant and linear decreasing inertia weight minimize the error, whereas other laws are better using other criteria. Nickabadi *et al.* suggested a new adaptive law, and compared it with different other settings of the inertia weight including constant, random, linear time varying, nonlinear time varying, and adaptive setting [8]. Their results show the superiority of the adaptive law they suggested. To end with, Arasomwan and Adewumi [18] covered another set of settings and showed that with good experimental setting, the linear decreasing law will perform competitively with similar variants.

The most common, these approaches present decreasing inertia weight. However, other schemes can be of interest. For instance, Malik *et al.* got better performance for sigmoid increasing law compared with sigmoid decreasing law [11].

III. PSO with Adaptive Inertia Weight

The new PSO algorithm, denoted w-PSO, introduces a new adaptive parametric setting. The new algorithm is easy to implement as the acceleration coefficients are constant and the inertia weight is dynamically updated using a simple feedback on the particles' best positions.

Many theoretical studies focused on the convergence related parameterization of PSO. It was demonstrated that the acceleration coefficients should obey $c_1 + c_2 < 4(1+\omega)$ [19], [20]. On the other hand, Martínez and Gonzalo showed that the sum of acceleration coefficients must be less than 4 [21], and recommended $c_1 = c_2$ to maximize the second order stability region. Moreover, equal values of acceleration coefficients gives the same weight to all the optima (global and local ones), and may avoid attraction to local optima during the exploitation phase. Therefore, we set the accelerations to the same value:

$$c_1 = c_2 = 1.5. \tag{3}$$

The local and global search are balanced in this algorithm via an adaptive inertia weight. A theoretical study [19], assuming time varying parameters, showed that the inertia weight should be between 0 and 1. Empirically [3], the inertia weight was recommended to vary in the range [0.4, 0.9]. We let ω vary in this range. First, we introduce d, a vector of K elements (K being a constant), each of which is defined as the maximum value (*max*) of the standard deviation (*std*) of $p^j(t)$ at each step of the algorithm

$$d(k) = \max_{1\le j\le D}\left\{std\left(p^j(t)\right)\right\}, \quad 1 \le k \le K, \tag{4}$$

where k=(t *modulus* K), and $p^j(t) = \left(p_i^j(t)\right)_{1\le i\le N}$ is the vector regrouping the j^{th} components of all $p_i(t)$. Then, ω is varied following

$$\omega(t) = 0.9 - 0.4\frac{d(k)}{\max_{1\le k\le K}\{d(k)\}} \tag{5}$$

Introducing $d(k)$ is obviously intended to get an indicator of the algorithm progress. Fig. 1 shows the variation of ω using the test function f_1 from CEC 2014 benchmark functions with K=1000.

When the particles' best positions get closer to each other, ω increases to reverse the trend and enable more exploration. Then, every K steps, ω decreases which may help more exploitation. In this way, the governing law of ω helps alternation of exploration and exploitation, which may improve the quality of the solution without using additional mechanisms (e.g. local search).

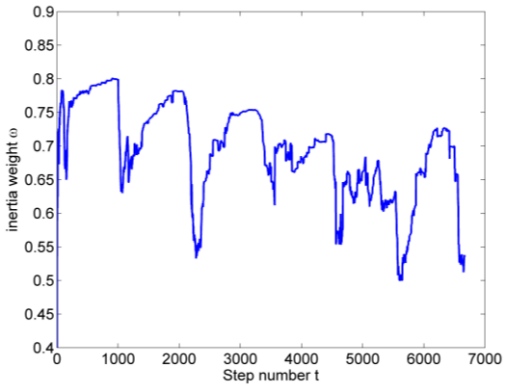


Fig. 1. Variations of the inertia weight as a function of the step number t.

IV. Simulation Results

The algorithm is tested using a set of thirty reference test functions recently introduced, and compared with four other settings of the inertia weight.

A. Reference Test Functions

Tests are carried using the CEC 2014 test functions (please refer to [9] for a full description):

- Functions f_1 to f_3 are unimodal functions.
- Functions f_4 to f_{16} are simple multimodal functions.
- Functions f_{17} to f_{22} are hybrid functions.
- Functions f_{23} to f_{30} are composition functions.

For each of these functions, the search space is $[-100,100]^D$ (D being the dimension search space). The values of the optimal solution are 100 to 3000 (by step of 100) for functions f_1 to f_{30}, respectively.

B. Inertia Weight Laws Used for Comparison

Based on previous comparative studies [8], [17], [18] (cf. section II), we choose the following time varying and adaptive laws for comparison:

- GPSO with linear decreasing law [3]:

  $\omega = 0.9 - 0.5(t/T) \in [0.4, 0.5]$, where t is the current step number and T is the maximal number of steps.
- Sugeno [4]:

  $\omega = 0.4 + 0.5\dfrac{1-(t/T)}{1+s(t/T)} \in [0.4, 0.5]$, where s is a constant greater than -1 and that is fixed to 10
- APSO with adaptive law [5]:

$\omega = \frac{1}{1+1.5\exp(-2.6f)} \in [0.4, 0.5]$, where f is the evolutionary factor computed using the distance between particles.

- AIWPSO with adaptive law [8]:

$\omega = S(t)/N \in [0,1]$, where N is the population *size* and $S(t)$ is the number of improved best positions at step t.

The chosen algorithms are tested using Matlab codes with the following parameters: the acceleration *coefficients* are set to 1.5, the population size N is set to 75, and the velocity of the global best $g(t)$ is regenerated each time its quality is improved. The parameter K introduced in our algorithm is set to 1000.

C. Comparison Criteria

We carry thirty realizations for each algorithm using the test functions in dimensions D=10 and D=50. The solutions are evaluated after carrying a number of function evaluations *FEs* given by the following law as recommended in Ref. [9]:

$$FEs = N \times T = N \times 10000 \times \frac{D}{N} = 10000 \times D. \quad (6)$$

The mean value and the standard deviation are computed and then algorithms are scored according to the number of times they yield the best results [9] (score varies between 0 and 30) in both dimensions 10 and 50.

D. Results and Discussion

The mean value and the standard deviation (giving the dispersion from average) of the best solutions are reported in Table I-Table II.

Comparing the performance of the algorithms on search spaces of dimension D=10 and D=50, we find that the quality of the solution (by the different settings) is deteriorated for some functions when increasing the dimension of search space. Such behavior may be avoided by increasing the number of function evaluations for D=50 or using additional mechanisms (e.g. hybridization) to these simple PSO algorithms.

TABLE I: THE MEAN VALUE ±STANDARD DEVIATION OF THE BEST SOLUTIONS FOR FIVE PSO ALGORITHMS IN DIMENSION D=10 (U IS THE MULTIPLICATION FACTOR OF THE VALUE)

	U	w-PSO	GPSO	Sugeno	APSO	AIWPSO
f_1	10^4	5.16 ±5.04	3.42 ±5.83	3.59 ±3.76	64.42 ±13.63	46.7 ±43.2
f_2	10^3	3.61 ±2.31	2.93 ±2.04	3.20 ±2.09	3.67 ±2.96	14.8 ±50.22
f_3	1	447 ±169	470 ±232	1733 ±1581	907 ±790	9603 ±3362
f_4	1	422 ±17	420 ±18	421 ±18	436 ±18	427 ±18
f_5	1	519 ±2.8	519 ±2.8	519 ±2.8	519 ±3.5	520 ±2.1
f_6	1	601 ±1.4	601 ±1.1	601 ±1.4	603 ±1.5	603 ±1.5
f_7	1	700.1 ±0.07	700.1 ±0.07	700.1 ±0.06	700.2 ±0.18	700.6 ±0.8
f_8	1	800.9 ±0.9	802.3 ±1.4	801.6 ±1.3	814.9 ±7.4	811.6 ±1.5
f_9	1	909.2 ±3.8	909.4 ±4.7	907.8 ±4.5	919.7 ±8.5	920.8 ±6.5
f_{10}	1	1138 ±98	1158 ±106	1131 ±99	1399 ±203	1481 ±219
f_{11}	1	1437 ±225	1440 ±106	1442 ±222	1775 ±296	1872 ±225
f_{12}	1	1200.1 ±0.1	1200.5 ±0.3	1200.2 ±0.3	1200.2 ±0.1	1200.9 ±0.3
f_{13}	1	1300.1 ±0.1	1300.2 ±0.1	1300.1 ±0.1	1300.3 ±0.1	1300.3 ±0.1
f_{14}	1	1400.1 ±0.1	1400.1 ±0.1	1400.1 ±0.1	1400.4 ±0.1	1400.3 ±0.1
f_{15}	1	1500.7 ±0.3	1501.1 ±0.5	1501 ±0.4	1501.6 ±0.7	1503.4 ±0.9
f_{16}	1	1602.3 ±0.6	1602.3 ±0.5	1602.1 ±0.5	1602.8 ±0.4	1602.8 ±0.4
f_{17}	10^3	7.47 ±12.07	5.47 ±3.27	4.32 ±2.63	6.60 ±10.95	20.77 ±8.21
f_{18}	10^3	8.91 ±7.91	11.16 ±10.4	12.75 ±10.2	10.39 ±9.2	27.42 ±24.8
f_{19}	1	1901.8 ±1	1901.7 ±0.7	1901.6 ±0.7	1902.9 ±1.4	1902.8 ±0.7
f_{20}	10^3	2.57 ±0.27	2.15 ±0.5	2.13 ±0.15	5.33 ±4.99	3.16 ±1.71
f_{21}	1	2227 ±108	2211 ±75	2198 ±61	2513 ±322	2226 ±21
f_{22}	1	2231 ±39	2218 ±25	2215 ±19	2292 ±64	2226 ±21
f_{23}	1	2629 ±8 10-5	2629 ±2 10 -12	2629 ±2 10-12	2630 ±1.7	2630 ±2.2
f_{24}	1	2520 ±6.3	2521 ±5.8	2520 ±5.1	2553 ±33	2534 ±12.6
f_{25}	1	2676 ±36	2685 ±32	2690 ±27	2696 ±15	2687 ±28
f_{26}	1	2700.1 ±0.03	2700.1 ±0.1	2700.1 ±0.05	2700.2 ±0.1	2700.3 ±0.1
f_{27}	1	2942 ±177	2990 ±142	2967 ±170	3019 ±142	3039 ±116
f_{28}	1	3274 ±77	3241 ±85	3233 ±76	3355 ±121	3301 ±71
f_{29}	10^5	6.30 ±13.96	7.09 ±16.06	4.55 ±12.43	8.24 ±20.84	3.50 ±10.58
f_{30}	10^3	4.02 ±0.33	3.85 ±0.33	3.86 ±0.28	4.71 ±0.54	4.24 ±0.40

However, in both dimensions, the best solutions are mainly found using the new algorithm w-PSO, GPSO, or Sugeno. First, in dimension D=10 and taking into account the dispersion of solutions, we find that:

- For functions f_5, f_7, f_{12}, f_{13}, f_{14}, f_{16}, and f_{26}, the five algorithms yield comparable solutions.
- For functions f_4, f_6, f_{11}, f_{19}, f_{23}, and f_{24}, w-PSO, GPSO and Sugeno yield the best solutions.
- For functions f_{22} and f_{30}, the best solutions are given by GPSO and Sugeno algorithms.
- w-PSO outperforms the other algorithms for functions f_8, f_{15}, f_{18}, f_{25}, and f_{27} as shown in Table I and Fig. 2.
- GPSO gives the best results for functions f_1 and f_2.
- AIWPSO outperforms the other algorithms for functions

f_3 and f_{29}.

- With Sugeno law, the best results are found for the remaining functions: f_9, f_{10}, f_{17}, f_{20}, f_{21} and f_{28}.

If the algorithms are scored according to the number of times they yield the best results, then w-PSO, GPSO, Sugeno, APSO and AIWPSO get, respectively, 19, 17, 21, 7 and 8.

On the other hand, in dimension D=50, the results are as follows:

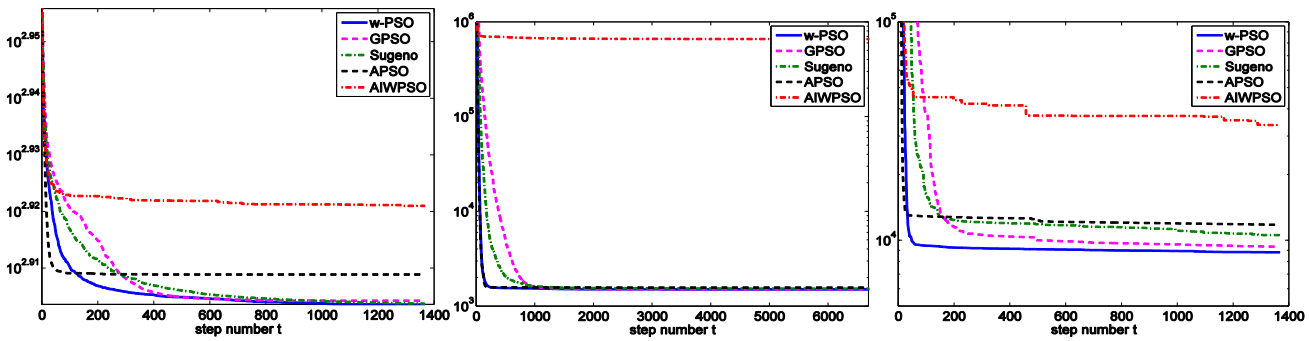
- For functions f_7, f_{13} and f_{14}, w-PSO, GPSO, Sugeno and APSO yield similar results.
- For functions f_4, f_{16} and f_{19}, the best solutions are given by w-PSO, GPSO and Sugeno algorithms.
- For functions f_8, f_{12} and f_{21}, w-PSO and Sugeno give the best solutions.
- For the function f_5, the best solutions are given by w-PSO and APSO.
- For the function f_{10} the best solutions are given by Sugeno.
- For functions f_3 and f_{22} w-PSO and GPSO guarantees the best solutions.
- GPSO gives the best solutions for functions f_9, f_{11}, f_{17}, f_{18}, f_{22}, f_{25}, f_{27}-f_{30}.
- w-PSO outperforms the other algorithms for the remaining functions: f_1, f_2, f_{15}, f_{20}, f_{24}, and f_{26} as shown in both Table II and Fig. 3.

Consequently, according to the number of times the algorithms give the best results, the algorithms get the following scores respectively: 19, 18, 10, 4, and 0 for w-PSO, GPSO, Sugeno, APSO, and AIWPSO.

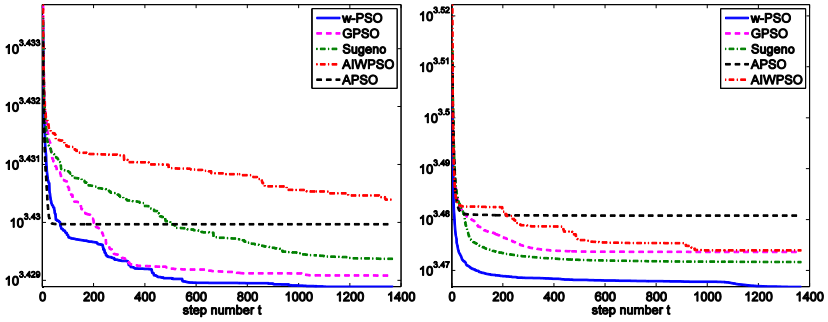
To conclude, the Sugeno law yields the best results in most cases in dimension D=10, whereas in dimension D=50, the new setting of ω outperforms the others in most cases.

TABLE II: THE MEAN VALUE ±STANDARD DEVIATION OF THE BEST SOLUTIONS FOR FIVE PSO ALGORITHMS IN DIMENSION D=50 (U IS THE MULTIPLICATION FACTOR OF THE VALUE)

	U	w-PSO	GPSO	Sugeno	APSO	AIWPSO
f_1	10^7	1.54±1.16	1.63±2.07	1.73±1.26	18.37±12.77	56.41±24.57
f_2	1	6.1±6.25 103	2.46±13.3 105	2.96 ±9 107	8.3±30.1 107	3.8±18.7 108
f_3	1	621±68	621±57	1941±5835	1097±545	97044±13196
f_4	1	422±17	420±18	421±18	436±18	427±18
f_5	1	520 ±2 10-5	520.9 ±0.1	521 ±0.05	520 ±10-5	521.1 ±0.04
f_6	1	628.8 ±4.7	630.1 ±3.8	631 ±5	644 ±4.6	656.4 ±6.8
f_7	1	700.01 ±0.01	700.01 ±0.01	700.01 ±0.01	700.01 ±0.01	1382 ±146
f_8	1	837 ±19	866 ±13	837 ±11	1000 ±51	1255 ±69
f_9	1	1084 ±45	1044 ±35	1052 ±37	1183 ±51	1470 ±53
f_{10}	1	2439 ±598	3277 ±594	2368 ±484	5045 ±726	14074 ±447
f_{11}	1	6813 ±737	6613 ±789	6625 ±941	7424 ±974	14282 ±585
f_{12}	1	1200.3 ±0.1	1200.9 ±0.4	1201.3 ±0.9	1200.6 ±0.2	1203.3 ±0.3
f_{13}	1	1300.5 ±0.1	1300.6 ±0.1	1300.5 ±0.1	1300.5 ±0.1	1305.7 ±0.5
f_{14}	1	1400.4 ±0.2	1400.6 ±0.3	1400.5 ±0.3	1400.4 ±0.1	1584 ±36
f_{15}	1	1512 ±3.3	1517 ±5.2	1517 ±4.6	1587 ±33	82675 ±76149
f_{16}	1	1620.3 ±0.9	1620 ±0.9	1620.8 ±0.8	1621 ±0.6	1622 ±0.3
f_{17}	10^6	3.57 ±4.34	2.03 ±1.91	2.99 ±2.32	16.31 ±14.05	25.66 ±7.62
f_{18}	1	3656 ±1650	2936 ±899	3484 ±1886	1.74 ±6.27 106	9.26 ±6.44 108
f_{19}	1	1965 ±20	1961 ±23	1966 ±19	1979 ±31	2230 ±116
f_{20}	10^3	2.63 ±0.20	2.73 ±0.32	3.69 ±0.73	3.49 ±0.85	44.57 ±11.46
f_{21}	10^6	0.71 ±1.72	1.13 ±1.44	0.71 ±0.7	1.59 ±2.67	10.89 ±3.82
f_{22}	1	3307 ±346	2972 ±288	3074 ±276	3518 ±328	4103 ±351
f_{23}	1	2646 ±1.1	2646 ±0.5	2648 ±0.8	2700 ±43	3105 ±137
f_{24}	1	2670 ±6.4	2678 ±4.5	2675 ±5.6	2695 ±12.7	2851 ±27.2
f_{25}	1	2725 ±4.5	2722 ±3.6	2724 ±3.6	2753 ±12.3	2783 ±21.8
f_{26}	1	2757 ±50	2774 ±62	2800 ±76	2789 ±61	2798 ±73
f_{27}	1	3828 ±291	3769 ±119	3849 ±119	4304 ±114	4572 ±118
f_{28}	1	6549 ±668	5157 ±797	6178 ±1091	78960 ±787	6365 ±997
f_{29}	10^7	1.70 ±5.28	1.28 ±3.92	2.61 ±5.99	10.98 ±17.11	5.87 ±6.56
f_{30}	10^5	1.01 ±0.73	0.41 ±0.14	0.49 ±0.22	5.53 ±5.14	8.69 ±13.64

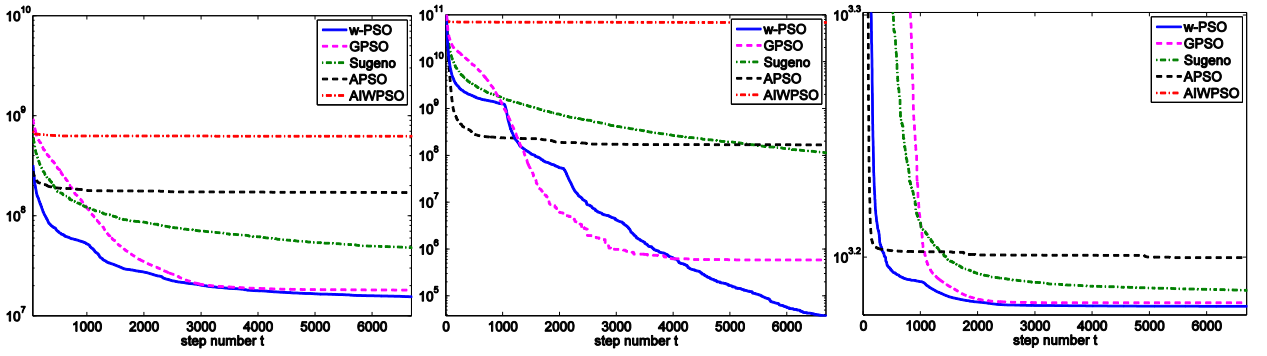


(a) function f_8. (b) function f_{15}. (c) function f_{18}.

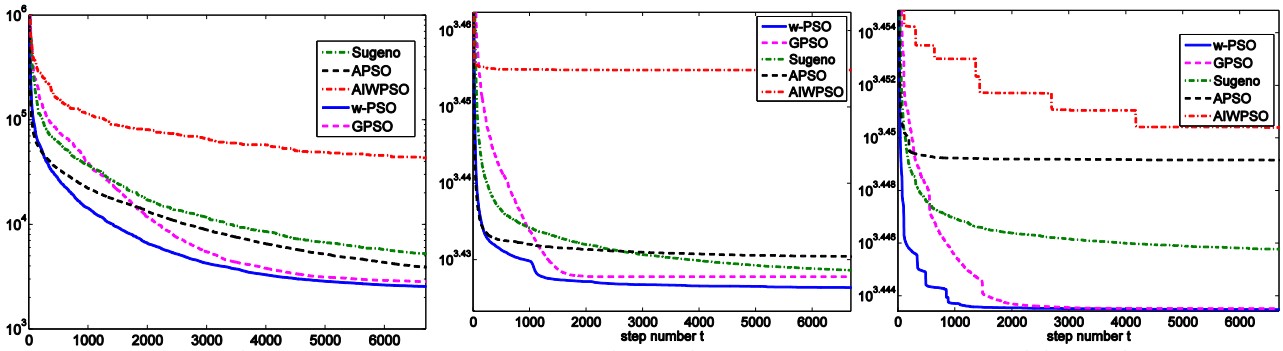


(d) function f_{25}. (e) function f_{27}.

Fig. 2. The mean of the best fitness for 30 independent runs as a function of step number in dimension D = 10 for functions f_8, f_{15}, f_{18}, f_{25} and f_{27}. [1]



(a) function f_1. (b) function f_2. (c) function f_{15}



(d) function f_{20}. (e) function f_{24}. (f) function f_{26}.

Fig. 3. The mean of the best fitness for 30 independent runs as a function of step number in dimension D = 50 for functions f_1, f_2, f_{15}, f_{20}, f_{24} and f_{26}.

V. Conclusions

In this paper, a new PSO algorithm (w-PSO) is introduced for global optimization. The objective of the study is to alternate exploration and exploitation during the algorithm progress.

We introduced a simple algorithm with constant accelerations coefficients and an adaptive inertia weight. The exploitation and exploration are alternated via the inertia weight, which is varying in the range [0.4, 0.9] using a feedback on particles' best positions. When particles' best positions get closer to each other, the inertia weight is increased to enable more exploration and prevent a premature convergence. The exploitation is ensured by decreasing ω every K steps. With this setting, the inertia weight keeps oscillating through the algorithm process instead of being automatically decreased as in many previous studies.

The new algorithm is tested on a set of thirty test functions (CEC 2014 benchmark functions) and compared with four other settings of inertia weight. Results show that the new setting is competitive with linear (GPSO) and Sugeno settings in low dimension. In dimension 10, with the new setting, the solutions are found to be the best in 19 out of 30 cases, giving to w-PSO the second place after Sugeno. Most importantly, w-PSO outperforms the other algorithms in solving problems in high dimension (D=50).

For its simplicity and efficiency, we expect the w-PSO to be successfully applied to solve many problems. For instance, in a future work, the w-PSO will be applied to optimize complex plasmonic structures [22].

References

[1] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proc. IEEE International Conference on Neural Networks*, Perth, Austarlia, 1995, pp. 1942-1948.

[2] S. Kessentini, D. Barchiesi, T. Grosges, and M. L. de la Chapelle, "Particle swarm optimization and evolutionary methods for plasmonic biomedical applications," in *Proc. IEEE Congress on Evolutionary Computation (CEC'11)*, New Orleans, LA, 2011, pp. 2315-2320.

[3] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization," in *Proc. IEEE Congress on Evolutionary Computation (CEC'99)*, Washington, DC, 1999, pp. 1945-1950.

[4] K. Lei, Y. Qiu, and Y. He, "A new adaptive well-chosen inertia weight strategy to automatically harmonize global and local search ability in

[1] Fig. 2 and Fig. 3 should be in printed color.

particle swarm optimization," in *Proc. First International Symposium on Systems and Control in Aerospace and Astronautics*, Harbin, 2006, pp. 977-980.
[5] Z.-H. Zhan, J. Zhang, Y. Li, and H. S.-H. Chung, "Adaptive particle swarm optimization," *IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics*, vol. 39, pp. 1362-1381, 2009.
[6] S. Kessentini, D. Barchiesi, T. Grosges, L. G. Moreau, and M. Lamy de la Chapelle, "Adaptive non-uniform particle swarm optimizetion: application to plasmonic design," *International Journal of Applied Metaheuristic Computing*, vol. 2, pp. 18-28, 2011.
[7] Y. Shi and R. C. Eberhart, "A modified particle swarm optimizer," in *Proc. IEEE Congress on Evolutionary Computation (CEC'98)*, Anchorage, AK, 1998, pp. 69-73.
[8] A. Nickabadi, M. M. Ebadzadeh, and R. Safabakhsh, "A novel particle swarm optimization algorithm with adaptive inertia weight," *Applied Soft Computing*, vol. 11, pp. 3658-3670, 2011.
[9] J. J. Liang, B. Y. Qu, and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization," Technical Report, December 2013.
[10] R. C. Eberhart and Y. Shi, "Tracking and optimizing dynamic systems with particle swarms," in *Proc. IEEE Congress on Evolutionary Computation (CEC'01)*, Seoul, South Korea, 2001, pp. 94-100.
[11] R. F. Malik, T. A. Rahman, S. Z. M. Hashim, and R. Ngah, "New particle swarm optimizer with sigmoid increasing inertia weight," *International Journal of Computer Science and Security*, vol. 1, pp. 35-44, 2007.
[12] W. A. Hassan, M. B. Fayek, and S. I. Shaheen, "PSOSA: An optimized particle swarm technique for solving the urban planning problem," in *Proc. International Conference on Computer Engineering and Systems*, 2006, pp. 401-405.
[13] G. Chen, X. Huang, J. Jia, and Z. Min, "Natural exponential inertia weight strategy in particle swarm optimization," in *Proc. Sixth World Congress on Intelligent Control and Automation (WCICA)*, 2006, vol. 1, pp. 3672-3675.
[14] H. R. Li and Y. L. Gao, "Particle swarm optimization algorithm with exponent decreasing inertia weight and stochastic mutation," in *Proc. Second International Conference on Information and Computing Science*, 2009, pp. 66-69.
[15] Y. Gao, X. An, and J. Liu, "A particle swarm optimization algorithm with logarithm decreasing inertia weight and chaos mutation," in *Proc. International Conference on Computational Intelligence and Security*, 2008, vol. 1, pp. 61-65.
[16] A. Nikabadi and M. Ebadzadeh, "Particle swarm optimization algorithms with adaptive inertia weight: a survey of the state of the art and a novel method," *IEEE Journal of Evolutionary Computation*, 2008.
[17] J. C. Bansal, P. K. Singh, M. Saraswat, A. Verma, S. S. Jadon, and A. Abraham, "Inertia weight strategies in particle swarm optimization," in *Proc. Third World Congress on Nature and Biologically Inspired Computing*, 2011, pp. 640-647.
[18] M. A. Arasomwan and A. O. Adewumi, "On the performance of linear decreasing inertia weight particle swarm optimization for global optimization," *The Scientific World Journal*, pp. 1-12, 2013.
[19] M. R. Rapaic and Z. Kanovic, "Time varying PSO — convergence analysis, convergence-related parameterization and new parameter adjustment schemes," *Information Processing Letters*, vol. 109, pp. 548-552, 2009.
[20] M. Jiang, Y. P. Luo, and S. Y. Yang, "Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm," *Information Processing Letters*, vol. 102, pp. 8-16, 2007.
[21] J. L. F. Martínez and E. G. Gonzalo, "The PSO family: Deduction, stochastic analysis and comparison," *Swarm Intelligence*, vol. 3, pp. 245-273, 2009.
[22] S. Kessentini and D. Barchiesi, "Quantitative comparison of optimized nanorods, nanoshells and hollow nanospheres for photothermal therapy," *Biomedical Optics Express*, vol. 3, pp. 590-604, 2012.

Sameh Kessentini was born in Tunisia. She gets her polyvalent engineering diploma in 2007 and master degree in mathematical engineering in 2008 from the Tunisia Polytechnic School. She receives her Ph.D. degree in optimization and systems security from University of Technology of Troyes, France in 2012. She is now working as a lecturer in Faculty of Science of Sfax-Tunisia, in the Department of Mathematics. Her major fields of interest are mathematical modeling, numerical methods, and optimization and advanced methods; with engineering applications. Her research working published in many journals and conferences' proceedings. She is also a reviewer for two indexed journals and many conferences since 2012.

Dominique Barchiesi was born on March 12, 1966 in France. He receives his B.S. degree in physics 1988 and B.S. degree in mathematics 1993, M.S. degree in physics 1989, Ph.D. degree in engineering 1993, tenure in physics and signal processing 1999 from the University of Franche-Comté. His major fields of research interest are numerical modelling, optimization and advanced methods with application to engineering of nanotechnologies and plasmonics, teaching of mathematics with strong links to physics and signal processing, and SPOC design.

He was an assistant professor at the University of Franche-Comté, France from 1993 to 1999 and is nowadays a full professor of theoretical physics, applied mathematics and statistics at the University of Technology of Troyes, France. The result of his research has been published in over one hundred fifty articles, conferences, and book chapters since 1993, in the fields of cryptography, signal processing, optimization, finite element method, plasmonics, near-field optical microscopies, optics, electromagnetism, and didactic of physics.