

Review on Inertia Weight Strategies for Particle Swarm Optimization

Ankush Rathore^(✉) and Harish Sharma

Rajasthan Technical University, Kota, India
ankushrathore777@gmail.com

Abstract. In the category of swarm intelligence based algorithms, Particle Swarm Optimization (PSO) is an effective population-based meta-heuristic used to solve complex optimization problems. In PSO, global optima is searched with the help of individuals. For the efficient search process, individuals have to explore whole search space as well as have to exploit the identified search area. Researchers are continuously working to balance these two contradictory properties i.e. exploration and exploitation and have been modified the PSO in many different ways to improve its solution search capability in the search space. In this regard, incorporation of inertia weight strategy in PSO is a significant modification and after that many researchers have been developed different inertia weight strategies to improve the solution search capability of PSO. This paper presents an analysis of the developed inertia weight strategies in respect to problem-solving capability and their effect in the solution search process of PSO. The effect of 30 recent inertia weight strategies on PSO is measured while comparing over ten well known test functions of having different degree of complexity and modularity.

Keywords: Soft computing · Optimisation · Inertia weight · Swarm intelligence · Nature inspired algorithms

1 Introduction

Particle Swarm Optimization (PSO) algorithm was developed by Eberhart and Kennedy in 1995 [1]. It is inspired by the intelligent behaviour of bird in search of food. The PSO algorithm is used to solve the different complex optimization problems including economics, engineering, complex real-world problems, biology and industry [2]. PSO can be applied to non-linear, non-differentiable, huge search space problems and gives better results with good accuracy [3].

For n- dimensional search space, the velocity and position of the i^{th} particle represents as: $V_i = (v_{i1}, v_{i2}, \dots, v_{id})^T$ and $X_i = (x_{i1}, x_{i2}, \dots, x_{id})^T$ respectively. Where, v_{id} and x_{id} is the velocity and position of i^{th} particle in d-dimension respectively. The velocity of the swarm (particle) is defined as follows:

$$v_{id}(new) = v_{id}(old) + c_1r_1(p_{id} - x_{id}) + c_2r_2(p_{gd} - x_{id}) \quad (1)$$

$$x_{id}(new) = x_{id}(old) + v_{id}(new) \quad (2)$$

where, $d = 1, 2, \dots, n$ presents the dimension and $i = 1, 2, \dots, N$ represents the particle index, N is the size of the swarm, c_1 and c_2 are called social scaling and cognitive parameters respectively that determines the magnitude of the random force in the direction of particle's previously best visited position (p_{id}) and best particle (p_{gd}) and r_1, r_2 are the uniform random variable between $[0, 1]$. The maximum velocity (V_{max}) assists as a constraint to control the position of the swarms within the solution search space.

Further, Shi and Eberhart [4] was developed the concept of an inertia weight (IW) in 1998 to ensure an optimal tradeoff between exploration and exploitation mechanisms of the swarm population. This inertia weight strategy was to be able to eliminate the need of maximum velocity (V_{max}). Inertia weight controls the particles movement by maintaining its previous memory. The velocity update equation is considered as follows:

$$v_{id}(new) = w * v_{id}(old) + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}) \quad (3)$$

This paper discusses the 30 different inertia weight strategies on 10 benchmark functions for PSO algorithm. A comprehensive review on 30 inertia weight strategies have been presented in next section.

2 A Review on Different Inertia Weight Strategies for PSO

Inertia weight plays an important role in the process of providing a trade-off between diversification and intensification skills of PSO algorithm. When the inertia weight strategy is implemented to PSO algorithm, the particles move around while adjusting their velocities and positions according to Eqs. (1) and (2) in the search space.

In 1998, first time Shi and Eberhart [4] proposed the concept of constant inertia weight. A small inertia weight helps in explore the search space while a large inertia weight facilitates in exploit the search space. Eberhart and Shi [5] proposed a random inertia weight strategy and enhances the performance and efficiency of PSO algorithm.

The linearly decreasing strategy [6] increases the convergence speed of PSO algorithm in early iterations of the search space. The inertia weight starts with some large value and then linearly decreases to some smaller value. The inertia weight provides the excellent results from 0.9 to 0.4. In global-local best inertia weight [7], the inertia weight is based on the global best and local best of the swarms in each generation. It increases the capabilities of PSO algorithm and neither takes a linearly decreasing time-varying value nor a constant value.

Fayek et al. [8] introduced a particle swarm simulated annealing technique (PSOSA). This inertia weight strategy is optimized by using simulated annealing and improves its searching capability.

Chen et al. [9] present two natural exponent inertia weight strategies as e1-PSO and e2-PSO, which are based on the exponentially decreasing the inertia weight. Experimentally, these strategies become a victim of premature convergence, despite its quick convergence speed towards the optimal positions at the early stage of the search process.

Using the merits of chaotic optimization, chaotic inertia weight has been proposed by Feng et al. [10] and PSO algorithm becomes better global search ability, convergence precision and quickly convergence velocity.

Malik et al. [11] presented a sigmoid increasing inertia weight (SIIW) and sigmoid decreasing inertia weight (SDIW). These strategies provide better performance with quick convergence ability and aggressive movement narrowing towards the solution region.

Oscillating Inertia Weight [12] provides a balance between diversification and intensification waves and concludes that this strategy looks to be competitive and, in some cases, better performs in terms of consistency.

Gao et al. [13] proposed a logarithmic decreasing inertia weight with chaos mutation operator. The chaos mutation operator can enhance the ability to jump out the premature convergence and improve its convergence speed and accuracy.

To overcome the stagnation and premature convergence of the PSO algorithm, Gao et al. [14] proposed an exponent decreasing inertia weight (EDIW) with stochastic mutation (SM). The stochastic mutations (SM) is used to enhance the diversity of the swarm while EDIW is used to improve the convergence speed of the individuals (Table 1).

Linearly decreasing inertia weight have been proposed by Shi and Eberhart [4] and greatly improved the accuracy and convergence speed. A large inertia weight facilitates at the inceptive phase of search space while later linearly decreases to a small inertia weight.

Adewumi et al. [25] proposed the swarm success rate random inertia weight (SSRRIW) and swarm success rate descending inertia weight (SSRDIW). These strategies use swarm success rates as a feedback parameter. Further, it enhances the effectiveness of the algorithm regarding convergence speed and global search ability.

Shen et al. [18] proposed the dynamic adaptive inertia weight and used to solve the complex and multi-dimensional function optimization problems. This strategy can timely adjust the particle speed, jump out of a locally optimal solution and improve the convergence speed.

Ting et al. [24] proposed the exponent inertia weight. There exist two important parameters as a local attractor (a) and global attractor (b). This method controls the population diversity by adaptive adjustment of local attractor (a) and global attractor (b).

Chatterjee and Siarry [22] proposed nonlinear decreasing inertia weight strategy with nonlinear modulation index. This strategy is quite effective as well as avoid premature issues. Lei et al. [17] proposed adaptive inertia weight. It furnishes with automatically harmonize global and local search ability and obtained the global optima.

Table 1. Inertia weight strategies

S.No.	Name of inertia weight	Formula of inertia weight
1	Logarithm Decreasing Inertia Weight [13]	$w = w_{max} + (w_{min} - w_{max}).\log_{10}(a + \frac{10t}{T})$
2	Exponent Decreasing Inertia Weight [14]	$w = (w_{max} - w_{min} - d_1).exp(\frac{1}{1 + \frac{t}{T}})$
3	Natural Exponent Inertia Weight Strategy(e2-PSO) [9]	$w = w_{min} + (w_{max} - w_{min}).e^{-[\frac{t}{4}]^2}$
4	Natural Exponent Inertia Weight Strategy(e1-PSO) [9]	$w = w_{end} + (w_{start} - w_{end}).e^{[\frac{-t}{10}]}$
5	Global-Local Best Inertia Weight [7]	$w = [1.1 - \frac{gbest_i}{pbest_i}]$
6	Simulated Annealing Inertia Weight [8]	$w = w_{min} + (w_{max} - w_{min}).\lambda^{k-1}$
7	Oscillating Inertia Weight [12]	$w = (\frac{w_{min} + w_{max}}{2} + \frac{w_{max} - w_{min}}{2} \cos(\frac{2\pi t}{T})),$ where $T = \frac{2S_1}{3+2k}$
8	Chaotic Random Inertia Weight [10]	$z = 4 * z * (1 - z), w = 0.5 * rand + 0.5 * z$
9	The Chaotic Inertia Weight [10]	$w = (w_{max} - w_{min}) * (\frac{T-t}{T}) + w_{min} * z,$ where, $z = 4 * z * (1 - z)$
10	Linear Decreasing Inertia Weight [6]	$w = w_{max} - (w_{max} - w_{min})(\frac{t}{T})$
11	Sigmoid Decreasing Inertia Weight [11]	$w = \frac{(w_{max} - w_{min})}{(1 + e^{-u(k-n*gen)})} + w_{min},$ $u = 10 \log((gen) - 2)$
12	Sigmoid Increasing Inertia Weight [11]	$w = \frac{(w_{max} - w_{min})}{(1 + e^{u(k-n*gen)})} + w_{min},$ $u = 10 \log((gen) - 2)$
13	Random Inertia Weight [5]	$w = 0.5 + 0.5 * rand$
14	Constant Inertia Weight [4]	$w = c,$ where $c = 0.2$ (considered for experiments)
15	Chaotic Adaptive Inertia Weights (CAIWS-D) [15]	$w = [(w_{max} - w_{min})(\frac{T-t}{T}) + w_{min}] * z,$ where, $z = 4 * SR * (1 - SR)$
16	Chaotic Adaptive Inertia Weights (CAIWS-R) [15]	$w = (0.5 * SR + 0.5) * z,$ where $z = 4 * SR * (1 - SR)$
17	Decreasing Exponential Function Inertia Weight (DEFIW) [15]	$w = t^{-\sqrt[3]{t}}$
18	Fixed inertia weight (FIW) [16]	$w = \frac{1}{2 \ln(2)}$
19	Adaptive Inertia Weight Strategy [17]	$w = [\frac{1 - (\frac{t}{T})}{(1 + S \frac{t}{T})}]$
20	Dynamic Adaptive Inertia Weight [18]	$w = w_{min} + (w_{max} - w_{min})F(t)\Psi(t),$ $\Psi(t) = exp(-\frac{t}{2\sigma^2})$ and $\sigma = T/3$
21	Decreasing Inertia Weight (DIW) [19]	$w = w_{init} * u^{-t}$
22	Inertia Weight Strategy [20]	$w = \frac{(w_{init} - 0.4)(g_{size} - t)}{(g_{size} + 0.4)}$
23	Double Exponential Dynamic Inertia Weight [2]	$w = exp(-exp(-R)),$ where $R = (\frac{T-t}{T})$
24	Tangent Decreasing Inertia Weight (TDIW) [21]	$w = (w_{max} - w_{min}) * \tan(\frac{\pi}{8}(1 - \frac{t}{T})^k)$
25	Nonlinear Decreasing Inertia Weight (NDIW) [22]	$w = (w_{max} - w_{min})(\frac{T-t}{T})^n + w_{min}$
26	Linear or Non-Linear Decreasing Inertia Weight [23]	$w = (\frac{2}{t})^{0.3}$
27	Exponent Inertia Weight [24]	$w = w_0 e^{-a(\frac{t}{T})}$
28	Swarm Success Rate Random Inertia Weight (SSRRIW) [25]	$w = 0.5 * rand + 0.5 * srr_{t-1}$
29	Swarm Success Rate Descending Inertia Weight (SSRDIW) [25]	$w = (w_{max} - w_{min})(\frac{T-t}{T}) + w_{min} * srr_{t-1}$
30	Descending Inertia Weight [25]	$w = w_{min} + (w_{max} - w_{min})(\frac{T-t}{T})$

J. asoc. [23] proposed the linear or non-linear decreasing inertia weight. This strategy has global search ability and also helpful to find a better optimal solution. It overcomes the weakness of premature convergence and converges faster

at the early stage of the search process. Jiao et al. [19] proposed the decreasing inertia weight (DIW). This strategy provides the algorithm with dynamic adaptability and controls the population diversity by adaptive adjustment of inertia weight.

Li, L. et al. [21] proposed the tangent decreasing inertia weight (TDIW) based on tangent function (TF). This strategy is to increase the diversity of swarm for more exploration of the search space at initial iterations while later exploit the search area. So that this approach provides better results with accuracy.

Chauhan et al. [2] proposed the double exponential dynamic inertia weight (DEDIW). The inertia weight is calculated for whole swarm iteratively by using gompertz function, and it is capable of providing a stagnation free environment with better accuracy. Peram et al. [20] proposed a new inertia weight that provides the less susceptible to premature convergence and less likely to be stuck in local optima. Sheng-Ta Hsieh et al. [16] introduced fixed inertia weight (FIW). It provides better convergence speed and less computational efforts.

The decreasing exponential function inertia weight (DEFIW) [15] decreases the value of inertia weight iteratively as the algorithm approaches equilibrium state and furnishes the superiority to the competitors in fitness quality.

Arasomwan et al. [15] Proposed chaotic adaptive inertia weights as CAIWS-D and CAIWS-R. These strategies simply combine chaotic mapping with the swarm success rate as a feedback parameter to harness together chaotic and adaptivity characteristics. These approaches provide more refine accuracy, faster convergence speed as well as global search ability.

3 Experimental Results

To evaluate the performance of the inertia weight strategy, it is tested over 10 different benchmark functions (F_1 to F_{10}) as given in Table 2.

3.1 Parameter Settings

Following experimental settings are adopted:

- $G_0 = 100$ and $\alpha = 20$ [26],
- Number of runs = 30,
- Number of populations = 50,
- Maximum number of iterations (T) = 1000,
- Value of c_1 and c_2 are 2.0 [25].

3.2 Results and Discussion

In this section, 30 different inertia weight strategies are analyzed on 10 benchmark problems in terms of average number of function evaluations (AFE's), mean error (ME) and standard deviation (SD). The AFE's, ME and SD are presented in Tables 3, 4 and 5 respectively. Boxplot of AFE's, ME and SD are shown in Figs. 1, 2 and 3 respectively.

Table 2. Test problems, D: Dimensions, AE: Acceptable Error

Test problem	Objective function	Search range	Optimum value	D	AE
Sphere	$f_1(x) = \sum_{i=1}^D x_i^2$	[-5.12 5.12]	$f(\mathbf{0}) = 0$	30	$1.0E - 05$
De Jong f4	$f_2(x) = \sum_{i=1}^D i \cdot (x_i)^4$	[-5.12 5.12]	$f(\mathbf{0}) = 0$	30	$1.0E - 05$
Ackley	$f_3(x) = -20 + e + exp(-\frac{0.2}{D} \sqrt{\sum_{i=1}^D x_i^3})$	[-30, 30]	$f(0) = 0$	30	$1.0E - 05$
Alpine	$f_4(x) = \sum_{i=1}^D x_i \sin x_i + 0.1x_i $	[-10, 10]	$f(0) = 0$	30	$1.0E - 05$
Michalewicz	$f_5(x) = -\sum_{i=1}^D \sin x_i (\sin(\frac{i \cdot x_i}{\pi}))^{20}$	[0, π]	$f_{min} = -9.66015$	10	$1.0E - 05$
Cosine Mixture	$f_6(x) = \sum_{i=1}^D x_i^2 - 0.1(\sum_{i=1}^D \cos 5\pi x_i) + 0.1D$	[-1, 1]	$f(0) = -D \times 0.1$	30	$1.0E - 05$
Exponential	$f_7(x) = -(exp(-0.5 \sum_{i=1}^D x_i^2)) + 1$	[-1, 1]	$f(0) = -1$	30	$1.0E - 05$
brown3	$f_8(x) = \sum_{i=1}^{D-1} (x_i^{2(x_{i+1})^2+1} + x_{i+1}^{2x_i^2+1})$	[-1 4]	$f(0) = 0$	30	$1.0E - 05$
Beale	$f_9(x) = [1.5 - x_1(1 - x_2)]^2 + [2.25 - x_1(1 - x_2^2)]^2 + [2.625 - x_1(1 - x_2^3)]^2$	[-4.5,4.5]	$f(3, 0.5) = 0$	2	$1.0E - 05$
Colville	$f_{10}(x) = 100[x_2 - x_1^2]^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1[(x_2 - 1)^2 + (x_4 - 1)^2] + 19.8(x_2 - 1)(x_4 - 1)$	[-10,10]	$f(\mathbf{1}) = 0$	4	$1.0E - 05$

It is clear from the reported results that most of the Inertia weight strategies produce poor results in case of michalewicz function (F_5). It clear from Fig. 1 that constant inertia weight and linearly decreasing inertia weight (LDIW) is best and worst strategy respectively in terms of AFE's. It is observed from Fig. 2 that the mean error taken by chaotic random inertia weight strategy and global local best inertia weight strategy are minimum and maximum in terms of mean error respectively compared to the other inertia weight strategies.

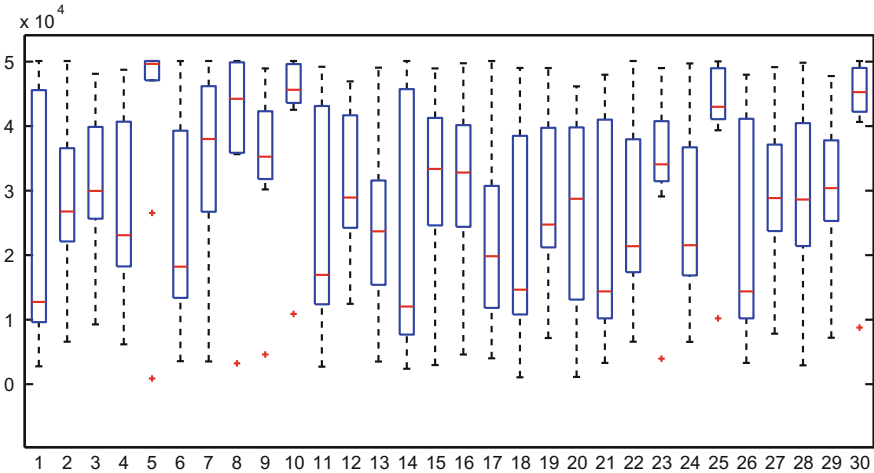


Fig. 1. Boxplots for average number of function evaluations of 30 different Inertia Weight strategies on 10 benchmark functions as per Table 3

Table 3. Average number of function evaluations of different inertia weight strategies for different benchmark functions

Inertia Weight	Sphere	De Jong f4	Ackley	Alpine	Michalewicz	Cosine Mixture	Exponential	brown3	Beale	Colville
1	09851.67	08540.00	15218.33	45575.00	49731.67	50100.00	9611.67	10270.00	2771.667	26260.00
2	22313.33	19670.00	32858.33	30181.67	50100.00	42166.67	22125.00	23351.67	6585.00	36578.33
3	25801.66	24028.33	35703.33	33166.67	48105.00	41483.33	25651.67	26728.33	09263.33	39870.00
4	18846.67	16835.00	29221.67	26516.67	48733.33	40691.67	18248.33	19665.00	06175.00	40780.00
5	50100.00	49196.67	50100.00	26536.67	48840.00	50100.00	50100.00	50100.00	00885.00	47108.33
6	13663.33	11920.00	23591.67	21495.00	50100.00	44928.33	13378.33	14915.00	3565.00	39290.00
7	29278.33	23780.00	50100.00	46190.00	47988.33	45698.33	26725.00	32083.333	3526.67	43920.00
8	39623.33	35648.33	50100.00	50100.00	47638.33	49888.33	35870.00	41895.00	3230.00	46575.00
9	32738.33	30196.67	42308.33	39525.00	48946.67	42490.00	31791.67	33640.00	04591.67	36876.67
10	44853.33	42535.00	50100.00	50096.67	48795.00	49615.00	43596.67	45206.67	10890.00	46073.33
11	13218.33	11451.67	23440.00	19735.00	49196.67	46841.67	12383.33	14135.00	02711.67	43118.33
12	25103.33	23176.67	34655.00	32036.67	46930.00	46618.33	24230.00	25821.67	12451.67	41675.00
13	15840.00	13016.67	29701.67	29538.33	49053.33	48191.67	15398.33	17851.67	03503.33	31575.00
14	8433.33	7330.00	18216.67	15083.33	49228.33	50100.00	07688.33	08975.00	02385.00	45735.00
15	26308.33	23300.00	47571.67	38541.67	48951.67	39000.00	24606.67	28170.00	2968.33	41273.33
16	26348.33	22995	47791.67	37485.00	49750.00	39408.33	24390.00	28135.00	4603.33	40158.33
17	11821.67	14613.33	28691.67	25045.00	50100.00	30740.00	11070.00	12343.33	4011.67	49971.67
18	11116.67	10255.00	20148.33	17286.67	49011.67	47786.67	10798.33	12016.67	01056.67	38490.00
19	21593.33	20201.67	28485.00	27230.00	49003.33	42706.67	21200.00	22236.67	07161.67	39731.67
20	13108.33	11238.33	44071.67	20541.67	46168.33	39801.67	36930.00	14003.33	1121.67	38113.33
21	10585.00	09818.33	17485.00	17256.67	47970.00	47661.67	10198.33	11496.67	03293.33	41001.67
22	18141.67	15730.00	26410.00	23940.00	50100.00	37963.33	17363.33	18841.67	06580.00	45256.67
23	32145	29100	44275.00	40775.00	49000.00	37770.00	31460.00	33501.67	03953.33	34638.33
24	17571.67	15713.33	26353.33	24056.67	49713.33	36716.67	16858.33	19021.67	06556.67	41691.67
25	42213.33	39358.33	50041.67	48980.00	49168.33	47785.00	41083.33	42838.33	10196.67	43158.33
26	10585.00	9818.33	17485.00	17256.67	47970.00	47661.67	10198.33	11496.67	03293.33	41131.67
27	24753.33	22180.00	35250.00	31966.67	49148.33	38595.00	23735.00	25720.00	07821.67	37138.33
28	23265.00	19488.33	40460.00	33003.33	49821.67	35823.33	21410.00	24258.33	02921.67	45111.67
29	26200.00	24011.67	36238.33	33400.00	47765.00	37798.33	25305.00	27360.00	07193.33	38271.67
30	45070.00	42221.67	50100.00	50085.00	49010.00	49023.33	43833.33	45480.00	08773.33	40638.33

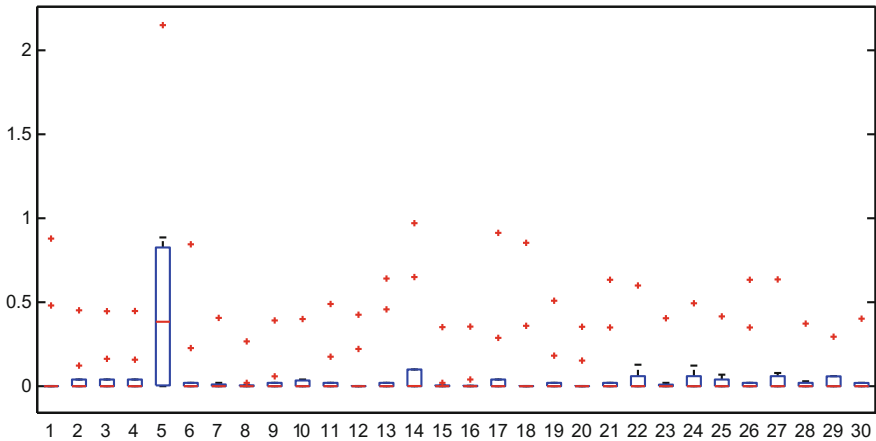


Fig. 2. Mean error value of 30 different Inertia Weight strategies on 10 benchmark functions as per Table 4

Table 4. Mean error value of different inertia weight strategies for different benchmark functions

Inertia Weight	Sphere	De Jong f4	Ackley	Alpine	Michalewicz	Cosine Mixture	Exponential	brown3	Beale	Colville
1	8.90E-06	8.29E-06	5.03E-04	9.14E-04	8.79E-01	4.81E-01	8.40E-06	8.66E-06	4.09E-06	1.50E-03
2	9.15E-06	8.91E-06	9.59E-06	9.57E-06	4.52E-01	1.23E-01	9.23E-06	9.16E-06	3.99E-02	1.57E-03
3	9.32E-06	8.97E-06	9.55E-06	9.64E-06	4.47E-01	1.63E-01	9.17E-06	9.13E-06	3.99E-02	1.34E-03
4	9.28E-06	8.74E-06	9.50E-06	9.54E-06	4.48E-01	1.58E-01	9.28E-06	9.18E-06	3.99E-02	1.22E-03
5	4.54E-01	3.14E-01	5.21E-01	1.15E-04	8.86E-01	2.15E+00	2.19E-01	8.26E-01	5.55E-06	4.84E-03
6	9.08E-06	9.13E-06	9.52E-06	9.25E-06	8.45E-01	2.27E-01	9.15E-06	9.24E-06	2.00E-02	1.18E-03
7	9.04E-06	8.78E-06	3.96E-05	1.37E-05	4.07E-01	1.08E-02	8.77E-06	9.36E-06	2.00E-02	6.33E-03
8	9.44E-06	8.87E-06	4.51E-04	5.51E-05	2.67E-01	5.41E-03	9.33E-06	9.51E-06	2.00E-02	5.13E-03
9	9.25E-06	8.97E-06	9.61E-06	9.13E-06	3.92E-01	5.91E-02	9.28E-06	9.19E-06	2.00E-02	1.84E-03
10	9.26E-06	8.96E-06	1.78E-04	6.26E-05	4.00E-01	3.45E-02	9.30E-06	9.43E-06	3.99E-02	4.95E-03
11	9.15E-06	8.91E-06	9.67E-06	9.56E-06	4.90E-01	1.76E-01	9.20E-06	9.15E-06	2.00E-02	3.64E-03
12	9.22E-06	8.61E-06	9.58E-06	9.40E-06	4.26E-01	2.22E-01	9.11E-06	9.30E-06	4.57E-06	1.54E-03
13	9.17E-06	9.26E-06	9.58E-06	8.69E-06	6.41E-01	4.58E-01	9.13E-06	9.05E-06	2.00E-02	9.58E-04
14	8.96E-06	8.77E-06	1.00E-01	8.07E-06	9.71E-01	6.50E-01	9.23E-06	8.98E-06	2.00E-02	2.55E-03
15	9.13E-06	8.74E-06	1.27E-05	9.31E-06	3.52E-01	4.94E-03	9.14E-06	9.54E-06	2.00E-02	3.33E-03
16	9.43E-06	9.05E-06	1.39E-05	9.62E-06	3.55E-01	9.43E-06	9.26E-06	9.44E-06	3.99E-02	2.86E-03
17	8.52E-06	8.88E-06	8.69E-06	5.52E-04	9.13E-01	4.93E-03	8.08E-06	8.46E-06	3.99E-02	2.88E-01
18	9.06E-06	8.69E-06	9.59E-06	9.31E-06	8.54E-01	3.60E-01	8.94E-06	9.28E-06	5.66E-06	1.32E-03
19	9.12E-06	8.56E-06	9.46E-06	9.36E-06	5.09E-01	1.82E-01	8.96E-06	8.90E-06	2.00E-02	2.06E-03
20	9.32E-06	9.22E-06	9.61E-06	9.38E-06	3.54E-01	1.53E-01	9.18E-06	9.23E-06	4.88E-06	1.09E-03
21	8.84E-06	8.84E-06	9.54E-06	7.78E-06	6.34E-01	3.50E-01	8.64E-06	9.01E-06	2.00E-02	4.69E-03
22	8.98E-06	9.09E-06	9.62E-06	9.38E-06	6.00E-01	1.28E-01	9.12E-06	9.13E-06	5.99E-02	2.13E-03
23	9.32E-06	9.19E-06	9.68E-06	9.52E-06	4.05E-01	9.86E-03	9.26E-06	9.35E-06	2.00E-02	2.02E-03
24	9.24E-06	9.12E-06	9.61E-06	9.63E-06	4.94E-01	1.23E-01	9.19E-06	9.01E-06	5.99E-02	1.94E-03
25	9.11E-06	8.62E-06	3.85E-05	1.29E-05	4.16E-01	6.90E-02	9.33E-06	9.35E-06	3.99E-02	2.93E-03
26	8.84E-06	8.84E-06	9.54E-06	7.78E-06	6.34E-01	3.50E-01	8.64E-06	9.01E-06	2.00E-02	2.01E-03
27	9.17E-06	9.04E-06	9.54E-06	9.50E-06	6.36E-01	7.88E-02	9.49E-06	9.05E-06	5.99E-02	1.18E-03
28	9.26E-06	9.23E-06	9.61E-06	9.67E-06	3.73E-01	2.96E-02	9.43E-06	9.21E-06	2.00E-02	3.46E-03
29	9.27E-06	8.97E-06	9.49E-06	9.65E-06	2.95E-01	5.91E-02	9.27E-06	9.29E-06	5.99E-02	1.54E-03
30	9.31E-06	9.07E-06	1.93E-04	4.63E-05	4.02E-01	1.97E-02	9.52E-06	8.96E-06	2.00E-02	3.62E-03

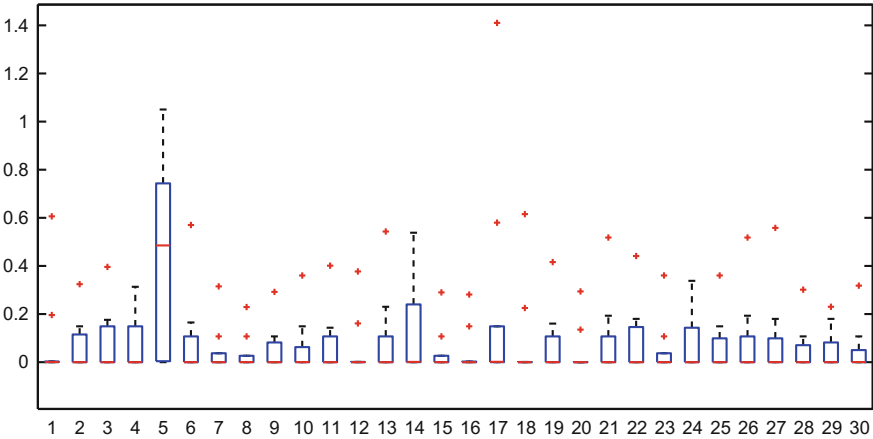


Fig. 3. Standard Deviation value of 30 different Inertia Weight strategies on 10 benchmark functions as per Table 5

Table 5. Standard deviation value of different inertia weight strategies for different benchmark functions

Inertia Weight	Sphere	De Jong f4	Ackley	Alpine	Michalewicz	Cosine Mixture	Exponential	brown3	Beale	Colville
1	1.54E-06	9.56E-07	2.66E-03	2.05E-03	6.06E-01	1.96E-01	1.86E-06	1.38E-06	2.38E-06	1.53E-03
2	8.26E-07	1.08E-06	3.58E-07	3.59E-07	3.24E-01	1.15E-01	6.84E-07	8.14E-07	1.49E-01	1.10E-03
3	6.19E-07	9.40E-07	3.49E-07	2.86E-07	3.96E-01	1.76E-01	7.40E-07	6.89E-07	1.49E-01	8.07E-04
4	6.45E-07	9.15E-07	4.10E-07	3.84E-07	3.13E-01	1.62E-01	5.22E-07	7.74E-07	1.49E-01	3.59E-04
5	4.34E-01	1.05E+00	5.36E-01	4.54E-04	6.88E-01	1.00E+00	1.64E-01	7.43E-01	3.02E-06	4.00E-03
6	8.16E-07	1.19E-06	3.28E-07	7.43E-07	5.70E-01	1.65E-01	5.50E-07	7.21E-07	1.07E-01	7.74E-04
7	7.26E-07	1.17E-06	2.10E-05	9.87E-06	3.15E-01	3.70E-02	1.09E-06	4.36E-07	1.07E-01	6.96E-03
8	5.53E-07	1.19E-06	2.74E-04	3.25E-05	2.29E-01	2.65E-02	6.58E-07	6.79E-07	1.07E-01	4.58E-03
9	6.61E-07	7.83E-07	3.54E-07	1.10E-06	2.92E-01	8.18E-02	4.59E-07	6.97E-07	1.07E-01	1.23E-03
10	6.61E-07	7.48E-07	1.23E-04	8.33E-05	3.60E-01	6.25E-02	6.82E-07	3.49E-07	1.49E-01	3.99E-03
11	7.83E-07	1.07E-06	4.02E-07	3.87E-07	4.01E-01	1.43E-01	5.79E-07	8.31E-07	1.07E-01	4.99E-03
12	5.26E-07	1.20E-06	4.27E-07	5.60E-07	3.77E-01	1.61E-01	6.72E-07	6.50E-07	3.30E-06	8.53E-04
13	8.54E-07	6.95E-07	4.37E-07	1.86E-06	5.43E-01	2.30E-01	8.55E-07	7.03E-07	1.07E-01	7.45E-05
14	1.02E-06	1.37E-06	3.75E-01	2.89E-06	5.38E-01	2.40E-01	4.98E-07	1.17E-06	1.07E-01	2.14E-03
15	8.31E-07	9.18E-07	5.80E-06	5.50E-07	2.90E-01	2.65E-02	5.92E-07	5.28E-07	1.07E-01	3.27E-03
16	6.17E-07	9.55E-07	1.22E-05	3.57E-07	2.81E-01	6.24E-07	6.58E-07	5.21E-07	1.49E-01	2.54E-03
17	1.74E-06	1.50E-06	1.63E-06	2.31E-03	5.80E-01	2.65E-02	2.04E-06	1.77E-06	1.49E-01	1.41E+00
18	9.20E-07	1.10E-06	5.99E-07	6.01E-07	6.15E-01	2.25E-01	8.30E-07	5.78E-07	2.85E-06	6.36E-04
19	7.09E-07	1.23E-06	5.04E-07	6.34E-07	4.16E-01	1.60E-01	9.73E-07	1.15E-06	1.07E-01	1.85E-03
20	6.24E-07	9.46E-07	3.94E-07	5.39E-07	2.94E-01	1.35E-01	5.31E-07	5.89E-07	2.98E-06	3.50E-04
21	8.07E-07	8.84E-07	3.55E-07	3.06E-06	5.18E-01	1.93E-01	1.36E-06	8.22E-07	1.07E-01	4.41E-03
22	7.53E-07	7.83E-07	4.40E-07	5.57E-07	4.41E-01	1.46E-01	8.90E-07	7.62E-07	1.80E-01	1.56E-03
23	6.08E-07	9.61E-07	3.55E-07	4.06E-07	3.60E-01	3.69E-02	6.05E-07	5.29E-07	1.07E-01	2.07E-03
24	6.71E-07	1.13E-06	3.58E-07	3.92E-07	3.38E-01	1.43E-01	6.37E-07	9.25E-07	1.80E-01	1.19E-03
25	9.37E-07	1.17E-06	2.93E-05	7.08E-06	3.60E-01	9.90E-02	4.63E-07	5.80E-07	1.49E-01	2.84E-03
26	8.07E-07	8.84E-07	3.55E-07	3.06E-06	5.18E-01	1.93E-01	1.36E-06	8.22E-07	1.07E-01	1.25E-03
27	6.81E-07	7.45E-07	4.11E-07	4.33E-07	5.58E-01	9.90E-02	4.22E-07	9.90E-02	1.80E-01	4.49E-04
28	6.41E-07	7.43E-07	3.61E-07	3.08E-07	3.01E-01	7.04E-02	5.76E-07	6.86E-07	1.07E-01	2.49E-03
29	6.65E-07	8.10E-07	3.12E-07	3.45E-07	2.30E-01	8.18E-02	5.80E-07	7.32E-07	1.80E-01	1.03E-03
30	5.71E-07	8.76E-07	9.32E-05	3.83E-05	3.18E-01	5.02E-02	3.12E-07	7.68E-07	1.07E-01	4.22E-03

If the comparison is made through standard divisions (SD's)the chaotic random inertia weight produces near optimal solutions in comparison to other inertia weight strategies as shown in Fig. 3. The summary results of inertia weight strategies are shown in Table 6.

Table 6. Summary Results for Inertia Weight

Criterion	Best inertia weight strategy	Worst inertia weight strategy
Average Function Evaluation	Constant Inertia Weight	Linear Decreasing Inertia Weight
Mean Error	Chaotic Random Inertia Weight	Global-Local Best Inertia Weight
Standard Deviation	Chaotic Random Inertia Weight	Global-Local Best Inertia Weight

4 Conclusion

This paper presents the significance of inertia weight strategies in the solution search process of particle swarm optimization (PSO). Here, total 30 inertia weight strategies in PSO are analyzed in terms of efficiency, reliability and robustness while testing over 10 complex test functions. Through boxplots and

success rate, it is found that the chaotic random inertia weight is better in terms of accuracy while constant inertia weight performs better in terms of efficiency of PSO among the considered inertia weight strategies.

References

1. Eberhart, R.C., Kennedy, J. et al.: A new optimizer using particle swarm theory. In: Proceedings of the sixth international symposium on micro machine and human science, vol. 1, pp. 39–43. New York (1995)
2. Chauhan, P., Deep, K., Pant, M.: Novel inertia weight strategies for particle swarm optimization. *Memetic Comput.* **5**(3), 229–251 (2013)
3. Bansal, J.C., Singh, P.K., Saraswat, M., Verma, A., Jadon, S.S., Abraham, A.: Inertia weight strategies in particle swarm optimization. In: 2011 Third World Congress on Nature and Biologically Inspired Computing (NaBIC), pp. 633–640. IEEE (2011)
4. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: The 1998 IEEE International Conference on Evolutionary Computation Proceedings, IEEE World Congress on Computational Intelligence, pp. 69–73. IEEE (1998)
5. Eberhart, R.C., Shi, Y.: Tracking and optimizing dynamic systems with particle swarms. In: Proceedings of the 2001 Congress on Evolutionary Computation, vol. 1, pp. 94–100. IEEE (2001)
6. Xin, J., Chen, G., Hai, Y.: A particle swarm optimizer with multi-stage linearly-decreasing inertia weight. In: International Joint Conference on Computational Sciences and Optimization, CSO 2009, vol. 1, pp. 505–508. IEEE (2009)
7. Arumugam, M.S., Rao, M.V.C.: On the performance of the particle swarm optimization algorithm with various inertia weight variants for computing optimal control of a class of hybrid systems. *Discrete Dyn. Nat. Soc.* **2006**, 15 (2006)
8. Al-Hassan, W., Fayek, M.B., Shaheen, S.I.: PSOSA: an optimized particle swarm technique for solving the urban planning problem. In: The 2006 International Conference on Computer Engineering and Systems, pp. 401–405. IEEE (2006)
9. Chen, G., Huang, X., Jia, J., Min, Z.: Natural exponential inertia weight strategy in particle swarm optimization. In: The Sixth World Congress on Intelligent Control and Automation, WCICA 2006, vol. 1, pp. 3672–3675. IEEE (2006)
10. Feng, Y., Teng, G-F., Wang, A-X., Yao, Y-M.: Chaotic inertia weight in particle swarm optimization. In: Second International Conference on Innovative Computing, Information and Control, ICICIC 2007, p. 475. IEEE (2007)
11. Malik, R.F., Rahman, T.A., Hashim, S.Z.M., Ngah, R.: New particle swarm optimizer with sigmoid increasing inertia weight. *Int. J. Comput. Sci. Secur.* **1**(2), 35–44 (2007)
12. Kentzoglanakis, K., Poole, M.: Particle swarm optimization with an oscillating inertia weight. In: Proceedings of the 11th Annual conference on Genetic and evolutionary computation, pp. 1749–1750. ACM (2009)
13. Gao, Y-l., An, X-h., Liu, J-m.: A particle swarm optimization algorithm with logarithm decreasing inertia weight and chaos mutation. In: International Conference on Computational Intelligence and Security, CIS 2008, vol. 1, pp. 61–65. IEEE (2008)
14. Li, H-R., Gao, Y-L.: Particle swarm optimization algorithm with exponent decreasing inertia weight and stochastic mutation. In: Second International Conference on Information and Computing Science, ICIC 2009, vol. 1, pp. 66–69. IEEE (2009)

15. Arasomwan, M.A., Adewumi, A.O.: On adaptive chaotic inertia weights in particle swarm optimization. In: 2013 IEEE Symposium on Swarm Intelligence (SIS), pp. 72–79. IEEE (2013)
16. Hsieh, S.-T., Sun, T.-Y., Liu, C.-C., Tsai, S.-J.: Efficient population utilization strategy for particle swarm optimizer. *IEEE Trans. Syst. Man Cybern. Part B Cybern.* **39**(2), 444–456 (2009)
17. Lei, K., Qiu, Y., He, Y.: A new adaptive well-chosen inertia weight strategy to automatically harmonize global and local search ability in particle swarm optimization. In: 1st International Symposium on Systems and Control in Aerospace and Astronautics, ISSCAA 2006, p. 4. IEEE (2006)
18. Shen, X., Chi, Z., Yang, J., Chen, C.: Particle swarm optimization with dynamic adaptive inertia weight. In: 2010 International Conference on Challenges in Environmental Science and Computer Engineering (CESCE), vol. 1, pp. 287–290. IEEE (2010)
19. Jiao, B., Lian, Z., Xingsheng, G.: A dynamic inertia weight particle swarm optimization algorithm. *Chaos Solitons Fractals* **37**(3), 698–705 (2008)
20. Peram, T., Veeramachaneni, K., Mohan, C.K.: Fitness-distance-ratio based particle swarm optimization. In: Proceedings of the 2003 IEEE on Swarm Intelligence Symposium, SIS 2003, pp. 174–181. IEEE (2003)
21. Li, L., Xue, B., Niu, B., Tan, L., Wang, J.: A novel particle swarm optimization with non-linear inertia weight based on tangent function. In: Huang, D.-S., Jo, K.-H., Lee, H.-H., Kang, H.-J., Bevilacqua, V. (eds.) ICIC 2009. LNCS (LNAI), vol. 5755, pp. 785–793. Springer, Heidelberg (2009). doi:[10.1007/978-3-642-04020-7_84](https://doi.org/10.1007/978-3-642-04020-7_84)
22. Chatterjee, A., Siarry, P.: Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization. *Comput. Oper. Res.* **33**(3), 859–871 (2006)
23. Fan, S.-K.S., Chiu, Y.-Y.: A decreasing inertia weight particle swarm optimizer. *Eng. Optim.* **39**(2), 203–228 (2007)
24. Ting, T.O., Shi, Yuhui, Cheng, Shi, Lee, Sanghyuk: Exponential inertia weight for particle swarm optimization. In: Tan, Ying, Shi, Yuhui, Ji, Zhen (eds.) ICSI 2012. LNCS, vol. 7331, pp. 83–90. Springer, Heidelberg (2012). doi:[10.1007/978-3-642-30976-2_10](https://doi.org/10.1007/978-3-642-30976-2_10)
25. Adewumi, A.O., Arasomwan, A.M.: An improved particle swarm optimiser based on swarm success rate for global optimisation problems. *J. Exp. Theor. Artif. Intell.* **28**, 441–483 (2016)
26. Rashedi, E., Nezamabadi-Pour, H., Saryazdi, S.: GSA: a gravitational search algorithm. *Inf. Sci.* **179**(13), 2232–2248 (2009)