Inertia Weight Strategies in Particle Swarm Optimization

[1]J. C. Bansal, [2]P. K. Singh [3]Mukesh Saraswat, [4]Abhishek Verma, [5]Shimpi Singh Jadon, [6,7]Ajith Abraham

[1,2,3,4,5]ABV-Indian Institute of Information Technology & Management, Gwalior, India

[6]Machine Intelligence Research Labs (MIR Labs), USA

[7]VSB Technical University of Ostrava, Czech Republic

[1]jcbansal@gmail.com, [2]pksingh7@gmail.com, [3]saraswatmukesh@gmail.com, [4]abhishekverma.cs@gmail.com, [5]shimpisingh2k6@gmail.com, , [6]ajith.abraham@ieee.org

Abstract: **Particle Swarm Optimization is a popular heuristic search algorithm which is inspired by the social learning of birds or fishes. It is a swarm intelligence technique for optimization developed by Eberhart and Kennedy [1] in 1995. Inertia weight is an important parameter in PSO, which significantly affects the convergence and exploration-exploitation trade-off in PSO process. Since inception of Inertia Weight in PSO, a large number of variations of Inertia Weight strategy have been proposed. In order to propose one or more than one Inertia Weight strategies which are efficient than others, this paper studies 15 relatively recent and popular Inertia Weight strategies and compares their performance on 05 optimization test problems.**

Keywords-Particle Swarm Optimization; Inertia Weight; Convergence.

I. INTRODUCTION

Particle Swarm Optimization (PSO) is an optimization technique inspired by social behavior of bird flocking and fish schooling in search of food. The technique was originally designed and developed by Eberhart and Kennedy [1]. The prominent features of PSO are its easy implementation, robustness to control parameters and computation efficiency compared with other existing heuristic algorithms such as genetic algorithm in a continuous problem. PSO can be applied to non-differentiable, non-linear, huge search space problems and gives the better results with a good efficiency. In PSO, instead of using more traditional genetic operators, each particle modifies its movement according to its own experience and its neighboring particle experience. The two equations which are used in PSO are position update equations and velocity update equation. These are to be modified in each iteration of PSO algorithm to converge the optimum solution. For an n- dimensional search space, the ith particle of the swarm is represented by a n- dimensional vector, $X_i = (x_{i1}, x_{i2}, \ldots, x_{in})^T$. The velocity of this particle is represented by another n-dimensional vector $V_i = (v_{i1}, v_{i2}, \ldots, v_{in})^T$. The previously best visited position of the ith particle is denoted as $P_i = (p_{i1}, p_{i2}, \ldots, p_{in})^T$. 'g' is the index of the best particle in the swarm. The velocity of the ith particle is updated using the velocity update equation given by (1) and the position is updated using (2).

$$v_{id} = v_{id} + c_1 r_1 (p_{id} - x_{id}) + c_2 r_2 (p_{gd} - x_{id}) \qquad (1)$$

$$x_{id} = x_{id} + v_{id} \qquad (2)$$

where $d = 1, 2 \ldots n$ represents the dimension and $i = 1, 2, \ldots, S$ represents the particle index. S is the size of the swarm and c_1 and c_2 are constants, called cognitive and social scaling parameters respectively (usually, $c_1 = c_2$; r_1, r_2 are random numbers drawn from a uniform distribution). Equations (1) and (2) define the classical version of PSO algorithm. A constant, V_{max}, was introduced to arbitrarily limit the velocities of the particles and improve the resolution of the search. The maximum velocity V_{max}, serves as a constraint to control the global exploration ability of particle swarm. Further, the concept of an Inertia Weight was developed by Shi and Eberhart [2] in 1998 to better control exploration and exploitation. The motivation was to be able to eliminate the need for V_{max}. The resulting velocity update equation becomes:

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

As there is a large effect of initial velocity in the balancing of exploration and exploitation process of swarm, Inertia Weight (w) is used to control the velocity. In this paper, Inertia Weight for PSO is reviewed and experiments are carried out over five basic benchmark optimization functions to compare different strategies of setting Inertia Weight.

II. DIFFERENT INERTIA WEIGHT STRATEGIES FOR PARTICLE SWARM OPTIMIZATION

Inertia Weight plays a key role in the process of providing balance between exploration and exploitation process. The Inertia Weight determines the contribution rate of a particle's previous velocity to its velocity at the current time step. The basic PSO, presented by Eberhart and Kennedy in 1995 [1], has no Inertia Weight. In 1998, first time Shi and Eberhart [2] presented the concept of Inertia Weight by introducing Constant Inertia Weight. They stated that a large Inertia Weight facilitates a global search while a small Inertia Weight facilitates a local search. Further,

dynamical adjusting of Inertia Weight was introduced by many researchers which can increase the capabilities of PSO. A review of Inertia Weight strategies in PSO is given chronologically in subsequent paragraphs.

Eberhart and Shi [3] proposed a Random Inertia Weight strategy and experimentally found that this strategy increases the convergence of PSO in early iterations of the algorithm. The Linearly Decreasing strategy [6] enhances the efficiency and performance of PSO. It is found experimentally that Inertia Weight from 0.9 to 0.4 provides the excellent results. In spite of its ability to converge optimum, it gets into the local optimum solving the question of more apices function.

In Global-Local Best Inertia Weight [9], the Inertia Weight is based on the function of local best and global best of the particles in each generation. It neither takes a constant value nor a linearly decreasing time-varying value. To overcome the weakness of premature convergence to local minimum, Adaptive Inertia Weight strategy [4] is proposed to improve its searching capability. It controls the population diversity by adaptive adjustment of Inertia Weight.

Fayek et al. [11] introduces an optimized Particle Swarm technique (PSOSA) that uses Simulated Annealing for optimizing the Inertia Weight and tested the approach on urban planning problem. The proposed technique gives much better as regards convergence speed as well as sustainability to increased load of growing number of blocks to be fitted in the urban planning problem. Chen et al. [13] present two Natural Exponent Inertia Weight strategies which are based on the basic idea of Decreasing Inertia Weight. Experimentally, these two new strategies converge faster than linear one during the early stage of the search process and provide better results for most continuous optimization problems. Using the merits of chaotic optimization, Chaotic Inertia Weight has been proposed by Feng et al. [7]. Comparison between CRIW PSO and RIW PSO has been done and found that CRIW PSO performs excellently. It has rough search stage and minute search stage alternately in all its evolutionary process.

Malik et al. [5] presented a Sigmoid Increasing Inertia Weight. They found that sigmoid function has contributed in getting minimum fitness function while Linearly Increasing Inertia Weight gives contribution to quick convergence ability. So they combine sigmoid function and Linear Increasing Inertia Weight and provides a SIIW which has produced a great improvement in quick convergence ability and aggressive movement narrowing towards the solution region. Oscillating Inertia Weight [8] provides periodically alternates between global and local search waves and conclusion was drawn that this strategy appears to be generally competitive and, in some cases, outperform particularly in terms of convergence speed.

Gao et al. [14] proposed a new PSO algorithm which combined the Logarithm Decreasing Inertia Weight with Chaos mutation operator. The Logarithm Decreasing Inertia Weight can improve the convergence speed, while the Chaos mutation can enhance the ability to jump out of the local optima. In order to overcome the premature convergence and later period oscillatory occurrences of the standard PSO, an Exponent Decreasing Inertia Weight and a stochastic mutation to produce an improved PSO has been proposed by Gao et al. [12] which uses the Exponent Decreasing Inertia Weight along with stochastic piecewise mutation for current global optimal particle during the running time, thus strengthened jumped out the partial optimal solution ability.

The summary of different Inertia Weight strategies is tabulated in Table 1 along with the required constraints.

III. EXPERIMENTAL RESULTS

To suggest a better strategy for a user of PSO with Inertia Weight, experiments have been carried out for 15 different Inertia Weight strategies over five optimization test problems.

A. Parameter Settings

Swarm size is taken to be 50. Number of decision variables is fixed to be 10 for each experiment. The termination criterion is set to the *"no improvement observed for 200 iterations (similar fitness value achieved for 200 consecutive iterations)"*. For those which require maximum number of iterations, 1000 iterations are used. To avoid the effect of choice of initial population, 30 simulations are taken. The value of acceleration parameters c_1 & c_2 are taken equal to 2. As previously discussed, five different test optimization functions are used for experiments. These functions are shown in Table 2 along with their range of search space. For implementing these 15 strategies in PSO, a C++ code has been developed and compiled in Dev C++ compiler.

TABLE 1. DIFFERENT INERTIA WEIGHT

Sr. No.	Name Of Inertia Weight	Formula of Inertia Weight	Reference
1.	Constant Inertia Weight	w=c c=0.7(considered for experiments)	[2]
2.	Random Inertia Weight	$w = 0.5 + \dfrac{Rand()}{2}$	[3]
3.	Adaptive Inertia Weight	$w_i(t+1) = w(0) + (w(n_t) - w(0))\dfrac{e^{m_i(t)} - 1}{e^{m_i(t)} + 1}$ $m_i(t) = \dfrac{gbest - current}{gbest + current}$	[4]
4.	Sigmoid Increasing Inertia Weight	$w_k = \dfrac{(w_{start} - w_{end})}{(1 + e^{u*(k-n*gen)})} + w_{end}$ $u = 10^{(log(gen)-2)}$	[5]
5.	Sigmoid Decreasing Inertia Weight	$w_k = \dfrac{(w_{start} - w_{end})}{(1 + e^{-u*(k-n*gen)})} + w_{end}$ $u = 10^{(log(gen)-2)}$	[5]
6.	Linear Decreasing Inertia Weight	$w_k = w_{max} - \dfrac{w_{max} - w_{min}}{iter_{max}} \times k$	[6]
7.	The Chaotic Inertia Weight	$z = 4 \times z \times (1 - z)$ $w = (w_1 - w_{2)} \times \dfrac{MAXiter - iter}{MAXiter} + w_2 \times z$	[7]
8.	Chaotic Random Inertia Weight	$z = 4 \times z \times (1 - z)$ $w = 0.5 \times rand() + 0.5 \times z$	[7]
9.	Oscillating Inertia Weight	$w(t) = \dfrac{w_{min} + w_{max}}{2} + \dfrac{w_{max} - w_{min}}{2}\cos\left(\dfrac{2\pi t}{T}\right)$ $T = \dfrac{2S_1}{3 + 2k}$	[8]
10.	Global-Local Best Inertia Weight	$Inertia\ weight\ w_i = (1.1 - \dfrac{gbest_i}{(pbest_i)})$	[9]
11.	Simulated Annealing Inertia Weight	$w_k = w_{min} + (w_{max} - w_{min}) \times \lambda^{(k-1)}$ $\lambda = 0.95$	[11]
12.	Natural Exponent Inertia Weight Strategy (e1-PSO)	$w(t) = w_{min} + (w_{max} - w_{min}).e^{-\left[\frac{t}{\left(\frac{MAXITER}{10}\right)}\right]}$	[13]
13	Natural Exponent Inertia Weight Strategy (e2 -PSO)	$w(t) = w_{min} + (w_{max} - w_{min}).e^{-\left[\frac{t}{\left(\frac{MAXITER}{4}\right)}\right]^2}$	[13]
14.	Logarithm Decreasing Inertia Weight	$w = w_{max} + (w_{min} - w_{max}) \times \log_{10}(a + \dfrac{10t}{T_{max}})$	[14]
15.	Exponent Decreasing Inertia Weight	$w = (w_{max} - w_{min} - d_1)\exp\left(\dfrac{1}{1 + \dfrac{d_2 t}{t_{max}}}\right)$	[12]

TABLE 2. DIFFERENT FUNCTIONS FOR SIMULATIONS

Function Name	Objective Functions	Search Space	Optimal Function Value
Sphere	$Min\ f(x) = \sum_{i=1}^{n} x_i^2$	$-5.12 \le x_i \le 5.12$	0
Griewank	$Min\ f(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$-600 \le x_i \le 600$	0
Rosenbrock	$Min\ f(x) = \sum_{i=1}^{n-1}[100(x_{i+1}-x_i^2)^2 + (x_i-1)^2]$	$-5 \le x_i \le 10$	0
Rastrigin	$Min\ f(x) = 10n + \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i)]$	$-5.12 \le x_i \le 5.12$	0
Ackley	$Min\ f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\cdot\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	$-30 \le x_i \le 30$	0

B.   Experimental Results and Analysis

The result analysis is done over three different criteria. These are average error, average number of iterations and minimum error obtained through all the simulations. Table 3 shows the result of average error obtained and corresponding box plots are given in Figure 1. It is observed that in case of Rosenbrock function; most of the Inertia Weight strategies produce poor results in comparison to all the other test functions taken. From Table 3 and Figure 1, it is obvious that Chaotic Inertia Weight strategy is best from the point of view of accuracy, while Chaotic Random Inertia Weight strategy is worst among all considered strategies. Average number of iterations required to produce the results for no improvement up to 200 iterations are also tabulated in Table 4 and corresponding box plots are given in Figure 2. From Figure 2, it is found that minimum average number of iterations is taken by Random Inertia Weight and maximum is taken by Constant Inertia Weight. Table 5 represents the minimum

error obtained after all simulations in each case considered above and corresponding box plots are given in Figure 3. It is clear that from the data represented in the Figure 3 that constant and linear decreasing Inertia Weight produce near optimum results in comparison to other methods. The summary of observations is given in Table 6.

IV.   CONCLUSIONS

This paper presents a comparative study on 15 strategies to set Inertia Weight in Particle Swarm Optimization Algorithm. A set of 5 most common optimization test problems and three criteria for comparison have been considered. As an overall outcome of the experiments carried out in this paper, Chaotic Inertia Weight is the best strategy for better accuracy. Random Inertia Weight strategy is best for better efficiency.

TABLE 3. AVERAGE ERROR VALUE OF DIFFERENT INERTIA WEIGHT STRATEGIES FOR DIFFERENT TEST PROBLEMS

Problem / Inertia Weight Strategy	Sphere	Griewank	Rosenbrock	Rastrigin	Ackley
Constant	0	0.0660	87.1177	0.9959	3.76E-15
Random	16.2943	84.7315	49419.74	99.8390	18.4242
Adaptive	5.1986	16.5427	4525.81	77.6964	13.3161
Sigmoid increasing	6.6605	29.4044	3138.98	61.6105	13.6855
Sigmoid decreasing	28.7272	83.9782	11677.4	85.4881	18.1589
Linear decreasing	7.01E-81	0.0691	6.1676	39.7121	2.94E-15
Chaotic	5.48E-81	0.0913	3.6398	3.2203	3.41E-15
Chaotic random	15.6258	91.7888	68234.573	85.3247	17.7203
Oscillating	0	0.0562	232.9622	2.8883	3.76E-15
Global-local best	19.8213	54.2368	67725.4	78.9104	17.9878
Simulated annealing	0	0.0669	44.077	4.183	2.94E-15
Natural exponent (e1-PSO)	0	0.0752	4.7199	2.7223	3.05E-15
Natural exponent (e2-PSO)	5.0885	30.6310	631.2029	31.8677	12.1880
Logarithm decreasing	2.34E-36	0.0859	3.7097	4.9466	4.36E-15
Exponent decreasing	0	0.0717	3.5584	3.6519	2.94E-15

TABLE 4. AVERAGE NUMBER OF ITERATIONS OF DIFFERENT INERTIA WEIGHT STRATEGIES FOR DIFFERENT TEST PROBLEMS

Problem / Inertia Weight Strategy	Sphere	Griewank	Rosenbrock	Rastrigin	Ackley
Constant	27611.9	3236.77	11512	3097.13	2853.97
Random	202.13	202.43	202	201.93	202
Adaptive	306	281.667	320.067	297.033	292.1
Sigmoid increasing	469.37	265.07	419	319.6	352.5
Sigmoid decreasing	205.57	206.73	205	210.4	203.37
Linear decreasing	2278.3	1460.767	1573.4	526.533	1254.9
Chaotic	2404.9	1345.733	2469.767	1121	1012.633
Chaotic random	202.2	202	202.57	201.9	202.43
Oscillating	13383.67	2207.9	26406.73	1535.133	1472.9
Global-local best	224.67	220.93	223	239.67	248.5
Simulated annealing	4245.8	821.43	2017.6	712.7	642.87
Natural exponent (e1-PSO)	4390.5	1043.9	1231.2	928.27	790.5
Natural exponent (e2-PSO)	1256.133	380.2	2248.667	531.3667	332.6
Logarithm decreasing	1573.1	822.87	1033.7	671	681.57
Exponent decreasing	3627.9	870.9	1958.4	754.17	764.83

TABLE 5. MINIMUM ERROR IN DIFFERENT INERTIA WEIGHT STRATEGIES FOR DIFFERENT TEST PROBLEMS

Problem / Inertia Weight Strategy	Sphere	Griewank	Rosenbrock	Rastrigin	Ackley
Constant	0	0.02954	0.000562	0	3.41E-15
Random	7.8711	53.5997	25029.5	83.968	16.3486
Adaptive	4.2681	14.363	658.723	62.4569	12.4772
Sigmoid increasing	5.9954	29.2358	3138.98	58.4766	12.9809
Sigmoid decreasing	26.7911	52.1624	11677.4	78.4488	17.4313
Linear decreasing	1.17E-121	0.0196	0.8224	1.9919	1.44E-16
Chaotic	1.64E-118	0.0221	3.0817	1.9919	3.41E-15
Chaotic random	12.1858	53.9584	46424.3	74.6869	16.9954
Oscillating	0	0.0147	1.29E-13	0.9960	3.41E-15
Global-local best	17.1738	48.6398	67725.4	71.5302	17.3844
Simulated annealing	0	0.0074	0.0099	2.9879	1.44E-16
Natural exponent (e1-PSO)	0	0.0467	1.2135	0.9959	1.44E-16
Natural exponent (e2-PSO)	0	0.1501	4.1470	1.9919	3.41E-15
Logarithm decreasing	5.75E-52	0.0859	2.6435	0.9959	3.41E-15
Exponent decreasing	0	0.0295	0.0013	1.9919	1.44E-16

TABLE 6. SUMMARY OF RESULTS

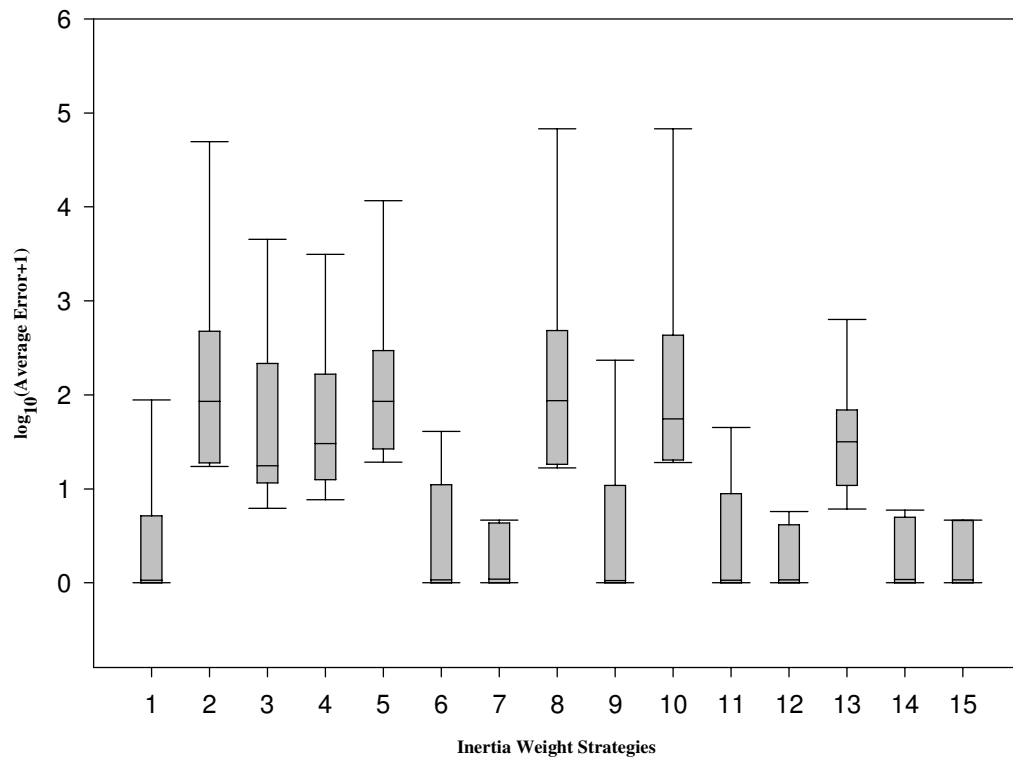| Criterion	Best Inertia Weight Strategy	Worst Inertia Weight Strategy
Average Error	Chaotic Inertia Weight	Chaotic Random Inertia Weight
Average Number of Iterations	Random Inertia Weight	Constant Inertia Weight
Minimum Error	Constant Inertia Weight / Linear decreasing Inertia Weight	Chaotic Random Inertia Weight / Global-Local Best Inertia Weight

Figure 1. Average error value of different Inertia Weight strategies for different test problems as per Table 3
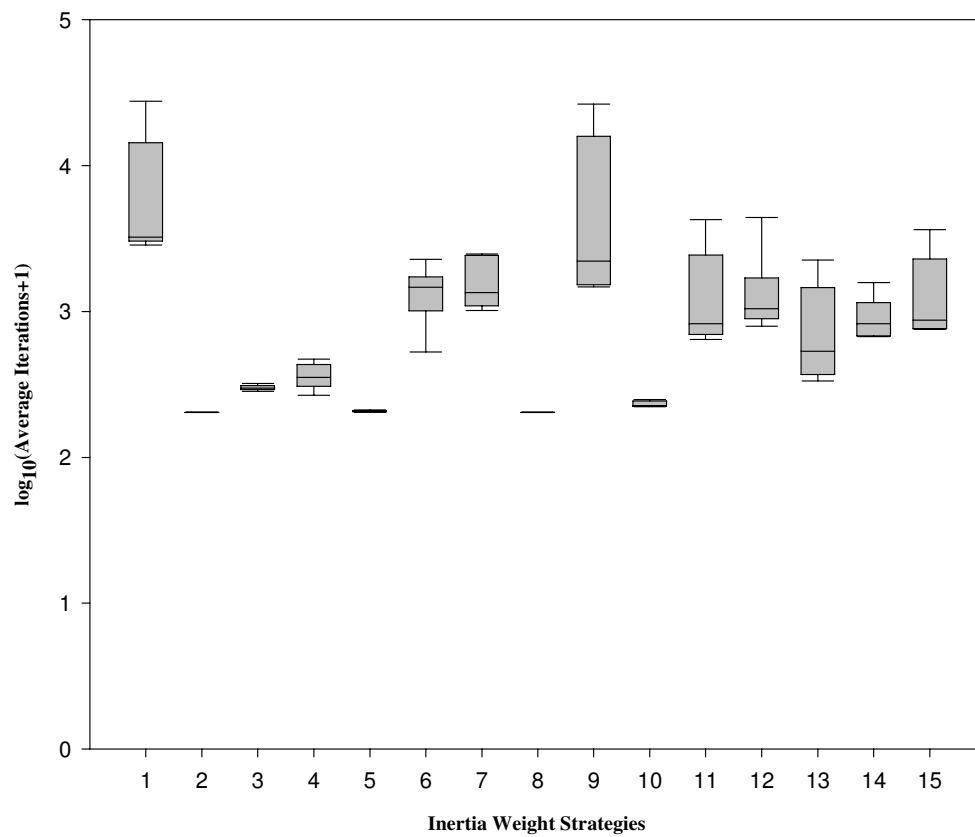

Figure 2. Average number of iterations of different Inertia Weight strategies for different test problems as per Table 4
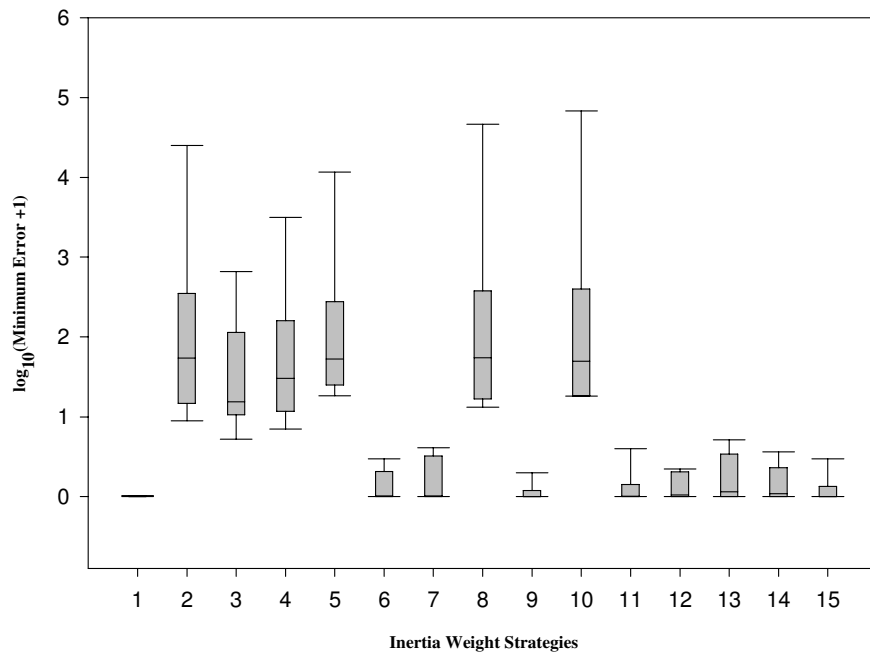
Figure 3. Minimum error in different Inertia Weight strategies for different test problems as per Table 5

REFERENCES

[1] J. Kennedy, R.C. Eberhart, et al., "Particle swarm optimization", In Proceedings of IEEE international conference on neural networks, volume 4, pages 1942–1948. Perth, Australia, 1995.

[2] Y. Shi and R. Eberhart., "A modified particle swarm optimizer", In Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., The 1998 IEEE International Conference on, pages 69–73. IEEE, 2002.

[3] R.C. Eberhart and Y. Shi., "Tracking and optimizing dynamic systems with particle swarms", In Evolutionary Computation, 2001. Proceedings of the 2001 Congress on, volume 1, pages 94–100. IEEE, 2002.

[4] A.Nikabadi, 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

[5] 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 (IJCSS), 1(2):35, 2007.

[6] J. Xin, G. Chen, and Y. Hai., "A Particle Swarm Optimizer with Multistage Linearly-Decreasing Inertia Weight", In Computational Sciences and Optimization, 2009. CSO 2009. International Joint Conference on, volume 1, pages 505–508. IEEE, 2009.

[7] Y. Feng, G.F. Teng, A.X. Wang, and Y.M. Yao., "Chaotic Inertia Weight in Particle Swarm Optimization", In Innovative Computing, Information and Control, 2007. ICICIC'07. Second International Conference on, page 475. IEEE, 2008.

[8] K. Kentzoglanakis and M. Poole., "Particle swarm optimization with an oscillating Inertia Weight", In Proceedings of the 11th Annual conference on Genetic and evolutionary computation, pages 1749–1750. ACM, 2009.

[9] M.S. Arumugam and MVC Rao., "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 Dynamics in Nature and Society, 2006, 2006.

[10] Y. Shi and R.C. Eberhart., "Empirical study of particle swarm optimization", In Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on, volume 3. IEEE, 2002.

[11] W. Al-Hassan, MB Fayek, and SI Shaheen, "Psosa: An optimized particle swarm technique for solving the urban planning problem", In Computer Engineering and Systems, The 2006 International Conference on, pages 401–405. IEEE, 2007.

[12] H.R. Li and Y.L. Gao., "Particle Swarm Optimization Algorithm with Exponent Decreasing Inertia Weight and Stochastic Mutation", In —2009 Second International Conference on Information and Computing Science, pages 66–69. IEEE, 2009.

[13] G. Chen, X. Huang, J. Jia, and Z. Min., "Natural exponential Inertia Weight strategy in particle swarm optimization", In Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on, volume 1, pages 3672–3675. IEEE, 2006.

[14] Y. Gao, X. An, and J. Liu., "A Particle Swarm Optimization Algorithm with Logarithm Decreasing Inertia Weight and Chaos Mutation", In Computational Intelligence and Security, 2008. CIS'08. International Conference on, volume 1, pages 61–65. IEEE, 2008.