Journal of Information & Optimization Sciences
ISSN 0252-2667 (Print), ISSN 2169-0103 (Online)
Vol. 47 (2026), No. 5-B, pp. 2119–2127
DOI : 10.47974/JIOS-2302

Dynamic inertia weight control using fitness metrics in particle swarm optimization algorithm

Suchita Arora *
Department of Computer Science and Engineering
Amity University Rajasthan
Jaipur 303002
Rajasthan
India

and

Department of Computer Science and Engineering
Poornima University
Jaipur 303905
Rajasthan
India

Sunil Kumar †
Department of Computer Science and Engineering
Amity University Rajasthan
Jaipur 303002
Rajasthan
India

Sandeep Kumar §
Department of AI and Data Science Engineering
CHRIST (Deemed to be University)
Bangalore 560074
Karnataka
India

* *E-mail:* suchi25arora@gmail.com (Corresponding Author)
† *E-mail:* skumar@jpr.amity.edu
§ *E-mail:* sandeepkumar@christuniversity.in

Abstract

Nature solves complex problems in a simple manner. This inspired researchers to develop a mathematical model to solve real-world problems that are not solvable or require high-end resources. The particle swarm optimization (PSO) is a nature-inspired techniques that have solved numerous engineering optimization problems. Sometimes PSO suffers from slow and premature convergence due to a high rate of exploration and exploitation, respectively. To overcome this problem, this paper introduced a modified PSO algorithm with dynamic inertia weight control with the help of fitness to improve convergence. The proposed approach uses dynamic inertia weight control in PSO to improve balancing in exploration and exploitation. Due to a fitness-driven approach, the proposed variant is adaptable for the population and avoids premature convergence, which is usually a problem basic PSO faces. The proposed approach is named as dynamic inertia weight control using fitness metrics in the PSO Algorithm (DIW-PSO). The experimental results demonstrated that DIW-PSO is best suited for multimodal high-dimensional problems and in a dynamic environment.

Subject Classification: *Primary 68T01, Secondary 68Q32.*

Keywords: *Particle swarm optimization, Engineering optimization, Nature inspired algorithm, Adaptive inertia weight.*

1. Introduction

Algorithms inspired by natural phenomenon are termed as Nature-inspired algorithms (NIA) and mainly divided into two classes: swarm intelligence and evolutionary algorithms [1-3]. Well-known swarm-based algorithms [4] are: ant colony optimization (ACO) by Dorigo and Caro [5] in 1992 with foraging and pheromone communication in ants, PSO developed by Kennedy and Eberhart [6] in 1995 inspired by collective behavior of fish schooling/bird flocking, spider monkey optimization (SMO) [7]. Other swarm-based algorithms are artificial bee colony (ABC) algorithm [8], cuckoo search (CS) [9], firefly algorithm (FA) [10], bat algorithm (BA) [11], bacterial foraging optimization (BFO) [12], and many more.

This study explored PSO and its applications with recent variants. Chauhan, Shivani, and Suganthan [13] conducted in-depth research of PSO and identified some challenges. Some of them are: effect of individual or hybrid strategies on performance, impact of learning mechanisms on convergence speed, robustness, exploration-exploitation, and development of self-adaptive variants for solving different problems across domains. Inertia weight (ω) is important parameters in PSO to control its performance. It gives better exploration with larger values while improving exploitation with small values of (ω)—a balanced value of ω in PSO is required in each iteration for better performance.

2. Literature Review

Zhang and Li [14] introduced a nonlinear decreasing inertia weight with mutation for high-dimensional problems. Zhang and Li [14] used the following approach (refer to Eq. (1)) to compute the inertia weight.

$$\omega = \omega_{min} + (\omega_{max} - \omega_{min}) \times (1 - (iter/MAX\_ITER)^{\alpha}) \quad (1)$$

Where ω_{max} and ω_{min} are upper and lower bounds on weight, $iter$ and MAX_ITER are current and maximum number of iterations. The proposed modification helped to avoid premature convergence.

Gandhi et al. [15] modified ω to improve convergence and reduce the influence of particles' previous velocity, and improved exploration and exploitation. Jiang et al. [16] introduced an adaptive approach for deciding the inertia weight dynamically using Eq. (2).

$$\omega_i^t = \begin{cases} m_1 & ftn_1(t) \geq ftn_{avg}(t) \\ m1 + \{m1 - m2\} \times \frac{ftn(t) - ftn_{gbt}(t)}{ftn_{avg}(t) - ftn_{gbt}(t)}, & ftn_i(t) < ftn_{avg}(t) \end{cases} \quad (2)$$

Jiang et al. [16] suggested assigning weights according to the population's average fitness. It improves local search if fitness is lower than average and enhances global search with higher fitness.

Chen et al. [17] introduced a variant of PSO with increasing inertia weight using a sigmoidal function. Earlier, a similar approach (refer Eq (3)) was used in SMO [18].

$$\omega = \omega_{min} + \frac{\omega_{max} - \omega_{min}}{1 + e^{a - bx \times \frac{iter}{iter_{max}}}} \quad (3)$$

The sigmoidal approach prevents PSO from trapping in local optima, giving a greater inertia weight at later iterations and a smaller value during the initial stage.

Wang et al. [19] changed the inertia weight dynamically using eq. (4), it updates the weight according to the situation and uses a classifier with optimized parameters.

$$\omega(t) = \omega_{min} + \left(\frac{\omega_{max} - \omega_{min}}{2}\right) * \frac{1}{\ln(e + (kz)^2)} + \left(\frac{\omega_{max} - \omega_{min}}{2}\right) * rand() \quad (4)$$

Lower and upper bounds for ω are 0.4 and 0.9, correspondingly [19]. The second component reduces the value of ω gradually, and this value is adjusted by the third component over time.

Bala, Chauhan, and Mitchell [20] worked on PSO with mutation strategy based on self-adaptive inertia weight and orthogonal initialization for improved convergence. Kumari and Kumar [21] predict student performance using different parameters to track their progress. The author proposed a new variant of PSO to select the optimal features that increase the model's efficiency.

Hossain et al. [22] formulated the performance index and applied the PSO algorithm to reduce electricity costs. The cost function is validated using different case studies, and it found that operational cost is reduced by 12% of the original cost function. Table 1 shows more modifications in PSO for different applications.

Table 1
Recent modifications in PSO

Author [Ref]	Modification in PSO	Application
Cui et al. [23]	Quaternary-valued PSO, define the particles' trajectories and velocities	Planar graph colouring problem
Abdi, Motamedi and Sharifian [24]	Merge the shortest job to the fastest processor algorithm into PSO	Task Scheduling in the cloud computing environment
Khan et al. [25]	Update parameters	Unimodal and multimodal test function, engineering inverse problem
Yan et al. [26]	Exponential decay weight for inertia weight	Exponential weight is tested on different benchmark functions
Khalilzadeh et al. [27]	Modified the rules for displacement of particles	Minimize the total weighted resource tardiness penalty cost
Syahputra, Soesanti, and Ashari [28]	Inertia weight is decreased linearly in proposed in PSO to improve exploration	Enhance the performance of distribution.
Zhou et al. [29]	Dynamically adjust the inertia weight to improve the speed of convergence	Cloud computing energy consumption model
Hung, Mao, and Huang [30]	Modify the calculation formulas of inertia weights	Train recurrent fuzzy neural network (RFNN)
Kumar et al. [31]	Sigmoidal approach for update inertia weight	Sentiment analysis using tweets related to Covid-19 vaccination public opinion

3. Dynamic Inertia Weight Control Using Fitness Metrics in PSO (DIW-PSO)

The PSO sometimes it suffers with the problem of premature or slow convergence caused due to excessive exploitation or excessive exploration respectively. The inertia weight (ω) is one of the important factors that decides balancing in these two activities. In basic PSO value of ω is linearly decreased from a high to a low with advancement in iteration, which is a static and creating problem.

This paper proposed a fitness-driven adaptive variant of PSO where the inertia weight is calculated dynamically using the fitness-based method, improving the proposed model's performance.

$$\omega = \omega_{max} - iter * f \tag{6}$$

where

$$f = \frac{(\omega_{max}-\omega_{min})(Fit_{max}-Fit_{min})}{(Fit_{max}+Fit_{min})(iter_{Max})} \tag{7}$$

Here in Eq. (6) and Eq. (7), ω_{max} and ω_{min} represent maximum (0.9) and minimum (0.2) value of inertia. This value is fixed after exhaustive experiments. Fit_{max} and Fit_{min} denotes maximum and minimum fitness in solutions.

The proposed dynamic weight adjustment makes PSO more responsive and improves convergence. The proposed dynamic inertia weight control using fitness metrics in PSO improve exploitation in early stages with diverse population as it decays faster due to large gap and maintained exploration in later stages with converged population as it decays slower due to narrow gap in $Fit_{max} - Fit_{min}$. These modifications make population adaptive with change in iteration and avoid premature convergence. Steps of proposed algorithm shown in Algorithm 1.

Algorithm 1: Dynamic Inertia Weight Control Using Fitness Metrics in PSO

1. Initialize position x_i and velocity v_i for each particle i = 1 to N.
2. Set personal best $pBest_i = x_i$
3. Evaluate fitness $f_i = f(x_i)$
4. Determine global best: $gBest = arg\ min_i\ f(pBest_i)$
5. For iter = 1 to $iter_{Max}$ do:
6. a. Compute $Fit_{max} = max_i f(pBest_i)$ and $Fit_{min} = min_i f(pBest_i)$
7. b. Calculate fitness-based decay factor:
8. $f = \left((\omega_{max} - \omega_{min}) * (Fit_{max} - Fit_{min})\right) / \left((Fit_{max} + Fit_{min}) * iter_{Max}\right)$
9. c. Update inertia weight:
10. $\omega = \omega_{max} - iter * f$
11. d. For each particle i = 1 to N do:
12. i. Generate $r_1, r_2 \sim U(0,1)$
13. ii. Update velocity:
14. $v_i' = \omega * v_i + c_1 * r_1 * (pBest_i - x_i) + c_2 * r_2 * (gBest - x_i)$
15. iii. Update position: $x_i = x_i + v_i'$
16. iv. Evaluate fitness $f_i = f(x_i)$
17. v. If $f_i < f(pBest_i)$, then set $pBest_i = x_i$
18. vi. If $f_i < f(gBest)$, then set $gBest = x_i$
19. e. End For
20. End For
21. Return $gBest,\ f(gBest)$

4. Experimental Results

The modified PSO has been compared with existing PSO for success rate, number of average function evaluation and standard deviation as given in Table 2. These identification algorithms are performed on the same parameters, an equal number of iterations, population size, and observations.

Table 2

Comparison for PSO and DIW-PSO

Function\Algorithm	PSO			DIW-PSO		
Evaluation Metric	SD	F Eval	SR	SD	F Eval	SR
Ackley	3.92E-07	76295	100	4.43E-07	69955	100
Alpine	5.00E-05	96774.5	99	5.02E-07	73832	100
Zakharov's	1.62E-02	196207.5	31	5.38E-04	101343	100
Cigar	4.97E-07	68869	100	8.13E-07	59724	100
Schewel Prob 3	3.96E-07	71026	100	4.93E-07	66396	100
Rotated hyper-ellipsoid	6.79E-07	56702	100	7.28E-07	54901	100
2D Tripod function	3.36E-01	38554	87	2.55E-01	36734	93
Shifted Rosenbrock	6.74E+00	186800.5	48	1.22E+01	123303.5	73
Hosaki Problem	3.29E-06	15323	93	3.38E-06	14995.5	94
Meyer and Roth Problem	2.68E-06	3354.5	100	2.91E-06	6180	100
Pressure Vessel	3.23E-05	96886	62	3.19E-05	94929	66

5. Conclusion

The modified inertia weight function based PSO proposed in this paper. This paper compared the performance of PSO with new variant using benchmark problems and finds that modified PSO gives better results than other existing algorithms. The proposed algorithm balances the exploration and exploitation phase. The results of modified PSO were compared with PSO for three metric and results were found better than existing algorithm.

In the future, this algorithm will address different image classification-based problems and real-world optimization problems. In addition to this, the algorithm can add or modify other parameters to solve the optimization problems very efficiently.

References

[1] R. Jain, L. Raja, S. K. Sharma, and D. P. Bhatt, “Particle swarm optimization model for Hindi text summarization,” *Journal of Information and Optimization Sciences*, vol. 45, no. 4, pp. 839–850 (2024), doi: 10.47974/JIOS-1609.

[2] H. A. H. Hasan and A. S. Ahmed, “Development of a new approach algorithm for ant colony optimization using Python program,” *Journal of Interdisciplinary Mathematics*, vol. 28, no. 1, pp. 95–105 (2025), doi: 10.47974/JIM-1800.

[3] I. Fister Jr, X. S. Yang, I. Fister, J. Brest, and D. Fister, "A brief review of nature-inspired algorithms for optimization," arXiv preprint, arXiv:1307.4186 (2013).

[4] E. Bonabeau, M. Dorigo, and G. Theraulaz, Swarm intelligence: from natural to artificial systems. Oxford University Press (1999).

[5] M. Dorigo and G. Di Caro, "Ant colony optimization: a new metaheuristic," in *Proc. 1999 Congress on Evolutionary Computation (CEC'99),* vol. 2, *IEEE* (1999).

[6] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Networks,* vol. 4, pp. 1942–1948 (1995).

[7] J. C. Bansal, H. Sharma, S. S. Jadon, and M. Clerc, "Spider monkey optimization algorithm for numerical optimization," *Memetic Computing*, vol. 6, no. 1, pp. 31–47 (2014).

[8] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Technical Report TR06, Erciyes University Press (2005).

[9] X. S. Yang, "Cuckoo search and firefly algorithm: overview and analysis," in Cuckoo search and firefly algorithm: theory and applications, pp. 1–26 (2013).

[10] X. S. Yang, O. Watanabe, and T. Zeugman, "Firefly algorithms for multimodal optimisation," in *Proc. 5th Symposium on Stochastic Algorithms, Foundations and Applications*, LNCS 5792, pp. 169–178 (2009).

[11] X. S. Yang, "A new metaheuristic bat-inspired algorithm," in *Nature inspired cooperative strategies for optimization (NICSO 2010)*, Springer, pp. 65–74 (2010).

[12] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Systems Magazine,* vol. 22, no. 3, pp. 52–67 (2002).

[13] D. Chauhan, Shivani, and P. N. Suganthan, "Learning Strategies in Particle Swarm Optimizer: A Critical Review and Performance Analysis," arXiv preprint, arXiv:2504.11812 (2025).

[14] J. Zhang and K. Li, "Fusion of Nonlinear Inertia Weight and Probability Mutation for Binary Particle Swarm Optimization Algorithm," in *Intelligence Computation and Applications (ISICA 2023)*, CCIS, vol. 2146, Springer, Singapore. https://doi.org/10.1007/978-981-97-4393-3_4

[15] R. Gandhi, D. Bhattacharya, A. Anand, S. Gope, A. Banik, and R. Roy, "Speed Control of PMSM Using Modified Particle Swarm Optimization Technique Based on Inertia Weight Updating Mechanism,"

SN Computer Science, vol. 4, no. 774 (2023), https://doi.org/10.1007/s42979-023-02095-3

[16] Z. Jiang, D. Zhu, X. Y. Li, and L. B. Han, "A Hybrid Adaptive Particle Swarm Optimization Algorithm for Enhanced Performance," *Applied Sciences,* vol. 15, no. 11, p. 6030 (2025), https://doi.org/10.3390/app15116030

[17] Z. Chen, Y. Wang, T. H. T. Chan, X. Li, and S. Zhao, "A Particle Swarm Optimization Algorithm with Sigmoid Increasing Inertia Weight for Structural Damage Identification," *Applied Sciences*, vol. 12, no. 7, p. 3429 (2022), https://doi.org/10.3390/app12073429

[18] B. Sharma, V. K. Sharma, and S. Kumar, "Sigmoidal Spider Monkey Optimization Algorithm," in *Soft Computing: Theories and Applications*, AIS, vol. 1053, Springer, Singapore (2020), https://doi.org/10.1007/978-981-15-0751-9_10

[19] J. Wang, X. Wang, X. Li, and J. Yi, "A Hybrid Particle Swarm Optimization Algorithm with Dynamic Adjustment of Inertia Weight Based on a New Feature Selection Method to Optimize SVM Parameters," *Entropy*, vol. 25, no. 3, p. 531 (2023).

[20] I. Bala, D. Chauhan, and L. Mitchell, "Orthogonally initiated particle swarm optimization with advanced mutation for real-parameter optimization," in *Proc. Genetic and Evolutionary Computation Conference Companion*, pp. 735–738 (2024).

[21] R. Kumari and S. Kumar, "Student's Performance Prediction Using Modified PSO," in *IEEE World Conference on Applied Intelligence and Computing* (AIC) (2023).

[22] M. A. Hossain, H. R. Pota, S. Squartini, and A. F. Abdou, "Modified PSO algorithm for real-time energy management in grid-connected microgrids," *Renewable Energy*, vol. 136, pp. 746–757 (2019).

[23] G. Cui, L. Qin, S. Liu, Y. Wang, X. Zhang, and X. Cao, "Modified PSO algorithm for solving planar graph coloring problem," *Progress in Natural Science*, vol. 18, no. 3, pp. 353–357 (2008).

[24] S. Abdi, S. A. Motamedi, and S. Sharifian, "Task scheduling using modified PSO algorithm in cloud computing environment," in Int. Conf. Machine Learning, Electrical and Mechanical Engineering, vol. 4, no. 1 (2014).

[25] S. Khan, M. Kamran, O. U. Rehman, L. Liu, and S. Yang, "A modified PSO algorithm with dynamic parameters for solving complex engi-

neering design problem," *International Journal of Computer Mathematics*, vol. 95, no. 11, pp. 2308–2329 (2018).

[26] C. Yan, G. Lu, Y. Liu, and X. Deng, "A modified PSO algorithm with exponential decay weight," in Proc. 13th Int. Conf. Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), *IEEE* (2017).

[27] M. Khalilzadeh, F. Kianfar, A. Shirzadeh Chaleshtari, S. Shadrokh, and M. Ranjbar, "A modified PSO algorithm for minimizing the total costs of resources in MRCPSP," *Mathematical Problems in Engineering*, (2012).

[28] R. Syahputra, I. Soesanti, and M. Ashari, "Performance Enhancement of Distribution Network with DG Integration Using Modified PSO Algorithm," *Journal of Electrical Systems*, vol. 12, no. 1 (2016).

[29] Z. Zhou, J. Chang, Z. Hu, J. Yu, and F. Li, "A modified PSO algorithm for task scheduling optimization in cloud computing," *Concurrency and Computation: Practice and Experience*, vol. 30, no. 24, pp. e4970 (2018).

[30] C. W. Hung, W. L. Mao, and H. Y. Huang, "Modified PSO Algorithm on Recurrent Fuzzy Neural Network for System Identification," *Intelligent Automation & Soft Computing*, vol. 25, no. 2 (2019).

[31] S. Kumar, M. B. Khan, M. H. A. Hasanat, A. K. J. Saudagar, A. AlTameem, and M. AlKhathami, "Sigmoidal Particle Swarm Optimization for Twitter Sentiment Analysis," *Computers, Materials & Continua*, vol. 75, no. 1, pp. 897–914 (2023).

Received April, 2025