Available *online* at : http://jnte.ft.unand.ac.id/

Jurnal Nasional Teknik Elektro

| ISSN (Print) 2302-2949 | ISSN (Online) 2407-7267 |

Hyperbolic Tangent - Based Adaptive Inertia Weight Particle Swarm Optimization

Yaw Opoku Mensah Sekyere, Francis Boafo Effah, Philip Yaw Okyere

Department of Electrical/Electronic Engineering, KNUST, Kumasi, Ghana.

ARTICLE INFORMATION

Received: May 09, 2023
Revised: June 27, 2023
Accepted: July 13, 2023
Available online: July 31, 2023

KEYWORDS

Particle Swarm Optimization, Adaptive Inertia Weight, Hyperbolic Tangent Function, Benchmark Functions, Convergence Rate

CORRESPONDENCE

Phone: +233 553577139
E-mail: yawsekyere@gmail.com

A B S T R A C T

This paper presents a study on using adaptive inertia weight (AIW) in particle swarm optimization (PSO) for solving optimization problems. An AIW function based on the hyperbolic tangent function was proposed, with the function parameters adaptively tuned based on the particle best and global best values. The performance of the proposed AIW-PSO was compared with standard PSO and other PSO variations using seven benchmark functions. The results showed that the proposed AIW-PSO outperformed the other variations in terms of minimum cost and mean cost while reducing the standard deviation of cost. The performance of the different PSO variations was also analysed by plotting the best cost against iteration, with the proposed AIW-PSO showing a faster convergence rate. Overall, the study demonstrates the effectiveness of using an adaptive inertia weight function in PSO for optimizing problems.

INTRODUCTION

The particle swarm optimization (PSO) algorithm has gained popularity recently due to its easy implementation, fast convergence, and ability to converge to satisfactory solutions [1]. The performance of the PSO algorithm heavily depends on the balance between its global and local searches, which is regulated by its control parameters, namely the inertia weight and acceleration coefficients [2]. The inertia weight is a key factor in controlling the balance between global and local searches during the search process [3]. For this reason, researchers have proposed various PSO variants that seek to enhance the original PSO algorithms' performance using the inertia weight modulation technique. One commonly used adaptive method is the time-varying parameter strategy, where the control parameters are updated over time [2]. The linearly decreasing strategy is a popular approach that enhances the efficiency and performance of PSO [4], [5], [6]. The global-local best inertia weight takes a function of the local best and global best of the particles in each iteration to prevent premature convergence to the local minimum. This method often results in slow convergence rates and reduced population diversity [7]. To address this issue, some researchers have explored adaptive strategies based on the distances of the particles to their personal best (pbest) and global best (gbest) positions [3],[8]. In one such method, the inertia weight is updated based on the distances between the particle's current position and its pbest and gbest positions [9]. This approach has improved PSO algorithms' convergence rate and population diversity. Also, a PSO algorithm that uses an exponential-based sigmoid function to update the inertia weight has been proposed [10]. The sigmoid function allows for gradual adjustments to the inertia weight, which helps to maintain population diversity and avoid premature convergence. The sigmoid function has also been combined with a linearly increasing inertia weight [10],[7]. This showed improvement in quick convergence ability and aggressive movement narrowing towards the solution region. Other weight modulation techniques include a logarithm decreasing inertia weight combined with a chaos mutation operator to improve the convergence speed and the ability to jump out of the local optima [7], an exponent decreasing inertia weight combined with stochastic piecewise mutation to produce an improved PSO that overcomes premature convergence and later period oscillatory occurrences [7], and a random Inertia Weight, where the Inertia Weight is randomly generated at each iteration [11], [12].

Various inertia weight modulation strategies have been proposed to enhance the capabilities of PSO. Each strategy has its strengths and weaknesses, and researchers continue to explore and propose new strategies to overcome the limitations

https://doi.org/10.25077/jnte.v12n2.1095.2023

of existing ones. This paper proposes a new approach for adaptive inertia weight in PSO based on the hyperbolic tangent (tanh) function. The proposed approach uses an adaptive equation to fine-tune the tanh function's parameters, which depend on the particle's best solution and the global best solution. We evaluate our approach using several benchmark functions, and the results show that our proposed approach outperforms the standard PSO, Linearly Decreasing Inertia Weight, Random Inertia Weight (RIW), where random inertia weights are generated after each iteration, and Exponential based Sigmoid inertia weight (ESIW) in terms of minimum cost and mean cost.

PARTICLE SWARM OPTIMIZATION

The original Particle Swarm Optimization (PSO) is a metaheuristic optimization algorithm inspired by the social behaviour of bird flocking or fish schooling, where each bird or fish is considered a particle [13].

Initialization

The PSO algorithm starts with a randomly generated particle population, each representing a potential solution to the optimization problem. These particles move through the search space by updating their velocity and position according to the best solution found by the particle itself and its neighbouring particles.

Position and Velocity Update process

The algorithm consists of two main processes: the position update process and the velocity update process. The position update process determines the new position of each particle in the search space, while the velocity update process determines the new velocity of each particle, as depicted in Fig. 1. The two processes are updated iteratively until a stopping criterion is met [3], [13]. The position and velocity update processes can be represented by (1) and (2), respectively:

$$x_i(t+1) = x_i\,(t) + \; v_i(t+1) \tag{1}$$

$$v_i(t+1) = w\, v_i\,(t) + \; c_1\big(p_i(t) - \; x_i\,(t)\big) + c_2\big(g(t) - \; x_i\,(t)\big) \tag{2}$$

Where $w \triangleq$ inertia weight (a constant for standard PSO). $c_1, c_2 \triangleq$ acceleration coefficients (which is a chosen constant for standard PSO), $x_i\,(t)$ represents the current position of a particle and $x_i(t+1)$ represents the updated position of a particle. $p_i(t)$ represents the personal best of a particle, g(t) represents the personal best of a particle. $v_i(t)$ represents the velocity of a particle and $v_i(t+1)$ the updated velocity of the updated particle with the position $x_i(t+1)$.

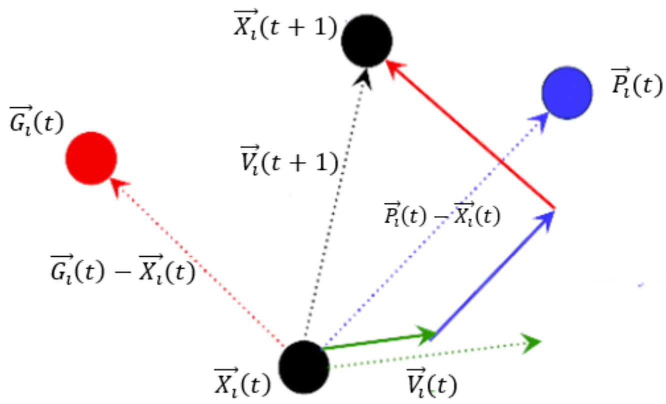

Figure 1. Particle swarm optimization vector diagram

The PSO algorithm iteratively updates the position and velocity of each particle in the swarm until a stopping criterion is met, such as reaching a maximum number of iterations or a desired level of fitness [14]. The algorithm searches for the optimal solution by exploring the search space and exploiting the best solutions. To facilitate the search process, the algorithm employs two constants, namely the cognitive and social constants, which allow each particle to consider its individual information and the impact of the group of particles, respectively [15]. The cognitive component (c_1) enables each particle to return to its previous best position for effective local search, while the social component (c_2) encourages the particle to move towards the overall best position of the swarm based on its proximity. These coefficients are also referred to as acceleration coefficients. In addition to the cognitive and social constants, PSO employs an inertia weight to control the balance between exploration and exploitation during the search process. The value of the inertia weight, *w*, affects the magnitude of the particle's velocity from one iteration to another. In the standard PSO, c1, c2 and *w* are all chosen as constants. The velocity of each particle is updated using (2), while the position is updated using (1) iteratively until an optimal solution is found. PSO has been applied successfully in various optimization problems, including engineering design, function optimization, and data clustering [16], [17].

(1)

PROPOSED ADAPTIVE INERTIA WEIGHT PSO

(2)

Like any optimization technique, the standard PSO has limitations, such as the premature convergence of the swarm to a suboptimal solution and the difficulty of handling high-dimensional problems [11]. Therefore, researchers have proposed various modifications to the standard PSO algorithm, including adaptive inertia weight and acceleration coefficients, to improve its convergence rate and optimization performance [18], [19], [20].

Proposed Adaptive Inertia Weight Function

The proposed PSO is achieved by modifying the constant Inertia Weight, *w* in the standard PSO into an adaptive function given in (3).

$$w = a \times \tanh(b) \tag{3}$$

https://doi.org/10.25077/jnte.v12n2.1095.2023

where 'a' is the tuning factor chosen as the function

$$a = \frac{Personal_{best} - Global_{best}}{Personal_{best}} \tag{4}$$

and 'b' is given by:

$$b = \frac{(w_{max} - w_{min})}{number\ of\ iterations\ performed} \tag{5}$$

These mathematical equations better balance exploration and exploitation processes in Particle Swarm Optimization by dynamically adjusting the Inertia Weight. The value of 'a' ensures the control of population diversity by adaptive adjustment of Inertia Weight, while the value of 'b' allows for a smooth transition from a large Inertia Weight that facilitates global search to a small Inertia Weight that facilitates local search. The variable is chosen to be inversely proportional to the number of iterations as convergence gets closer as more iterations are performed, and hence there will be the need to reduce the inertia weight. This is the idea based on which Linearly Decreasing Inertia Weight Operates. The advantage of the proposed model in this paper is that using the calculated ‘b’ as input to the hyperbolic tanh sets boundaries. The constant ‘a’ also constantly adjusts the boundaries of the tanh output as the difference between the global best and the personal best changes. Figure 2 shows that adjusting the constant ’a’ value adjusts the output boundary limits.

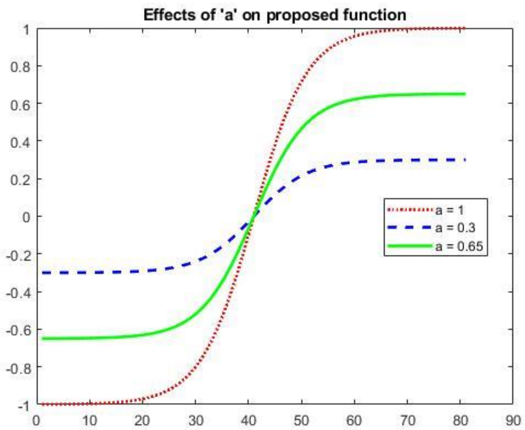

Figure 2. Effects of adjusting the variable ‘a’ on the proposed function

Figure 2 again shows that if this proposed model is applied, inertia weight will change smoothly as more iterations occur to encourage exploitation.

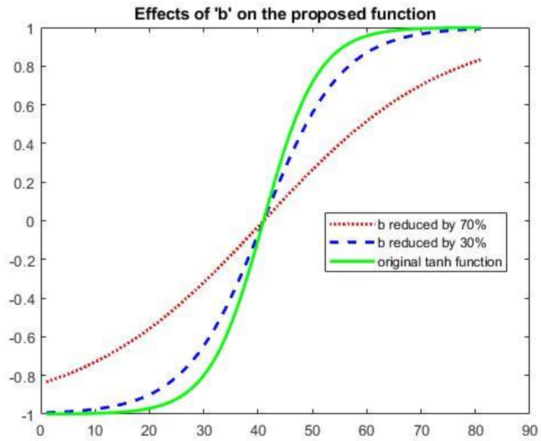

Figure 3. Effects of ‘b’ on our proposed function

To explain the advantages of the tanh-based adaptive inertia PSO over the other types of inertia weights, we need to analyse the plotted graphs and their characteristics.

First, let's consider the exponential-based sigmoid inertia weight function, which is given by

$$y = \frac{1}{1 + e^{-x}} \tag{6}$$

This function starts at 0 and gradually increases, reaching a maximum value of 1 when x is infinity. This means the inertia weight decreases rapidly at the beginning and slowly reaches a maximum value. However, this function does not have a clear plateau, which can lead to premature convergence and lack of exploitation. This is depicted in Figure 4.

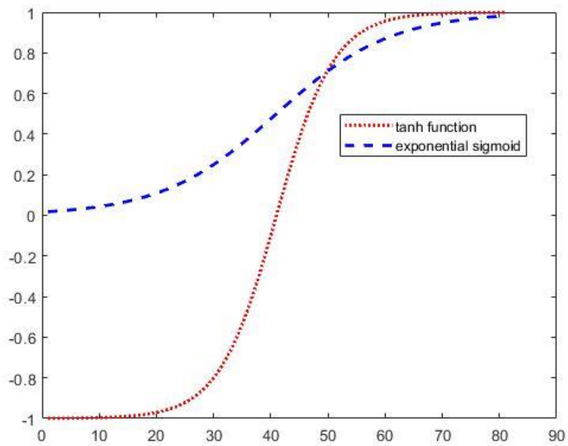

Figure 4. Exponential sigmoid function plotted alongside a tanh-based function

On the other hand, considering a linearly decreasing inertia weight function,

$$y = mx \tag{7}$$

This has a constant slope, and the function linearly decreases with time, which can lead to a lack of exploration at the beginning of the search process. This function does not have a plateau; even though it can converge quickly, it may not reach the optimal solution.

Strengths of the Proposed Adaptive Inertia Weight Function

The proposed tanh-based inertia weight function has several advantages over the other two. First, this function has a clear plateau at the beginning of the search process, which allows for exploration and prevents premature convergence. Second, this function increases gradually and reaches a maximum value, allowing exploitation and convergence to the optimal solution. Third, this function has an adjustable slope, which ensures a gradual transition between exploration and exploitation. These characteristics balance exploration and exploitation, leading to better search performance and convergence to the optimal solution in adaptive inertia particle swarm optimization.

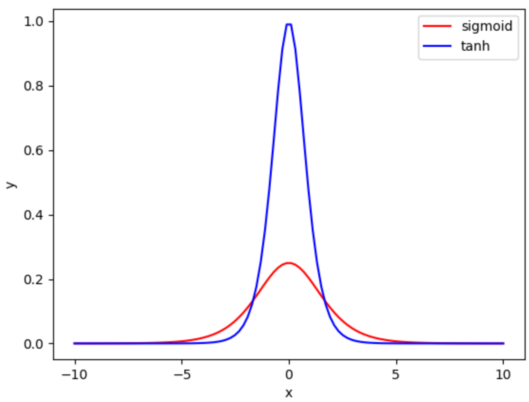

Figure 5. Derivative curves of tanh and sigmoid functions

The derivative curves of the tanh and sigmoid functions illustrate that the tanh function has a steeper slope when input values approach zero compared to the sigmoid function. This means that the value of the tanh-based function changes more rapidly as its input approaches zero, which is an important property for an adaptive inertia weight equation.

In a PSO algorithm, the adaptive inertia weight is used to balance exploration and exploitation during the search process, and its value is updated at each iteration based on the fitness improvement achieved by the particles. The proposed equation for updating the inertia weight includes a tanh function that inputs the difference between the global and particle best. As the input to the tanh function approaches zero, the steep slope of the tanh derivative curve ensures that the value of the inertia weight converges more rapidly towards its upper or lower boundaries, which are also adjusted adaptively based on the difference between global best and personal best, depending on whether the fitness improvement is positive or negative, respectively. This enables the particles to quickly adapt their behaviour towards the direction of improvement and explore more efficiently without getting stuck in local optima.

In contrast, if the exponential sigmoid function were used in the inertia weight equation, its softer slope near zero would result in slower convergence of the inertia weight, leading to slower adaptation of particle behaviour and potentially longer search times. Therefore, the proposed tanh-based function will be a better choice for the inertia weight equation in PSO, as it enables faster convergence and more efficient exploration of the search space. Therefore, based on their derivatives and their ability to encourage exploration, the tanh function is a better choice for adaptive inertia weight application in particle swarm optimization than the sigmoid function.

Also, if the input values are small, the tanh function can be approximated using a Taylor series expansion, which reduces the computational resources required to evaluate the function.
Lastly, the computation resource requirement is significantly reduced when using the Taylor series expansion to approximate the tanh function for small input values. The Taylor series expansion of the tanh function is shown in (5).

$$\tanh(x) = x - \frac{1}{3}x^3 + \frac{2}{15}x^5 - \frac{17}{315}x^7 + \cdots \quad (5)$$

From (5), as x gets smaller, higher terms can be ignored. When this is used, the increased computational resources involved in computing exponentials, as in the sigmoid function, can be reduced.

TESTING

This section describes the series of computational experiments conducted to assess the effectiveness of the proposed tanh-based adaptive inertia particle swarm optimization (TIW-PSO). To evaluate the performance of the TIW-PSO, we applied it to 7 standard benchmark optimization functions using MATLAB programming software. The performance of the TIW-PSO was then compared to that of the standard PSO and other proposed adaptive inertia weight particle swarm optimization algorithms found in the literature to determine its efficacy.

Benchmark Optimization Functions

Details of the benchmark functions are listed in Table 1.

Table 1: Benchmark functions

Function Name	Search Range	Dimension	Optimum Value
Sum Square	[-10,10]	50	0
Sphere	[-100, 100]	10	0
Colville	[-10,10]	4	0
Matyas	[-10, 10]	2	0
Rosenbrock	[-5, 10]	50	0
Greiwank	[-600, 600]	2	0
Rotated Hyper-Ellipsoid	[-65.536,65.536]	2	0

https://doi.org/10.25077/jnte.v12n2.1095.2023

The proposed Tanh-based Inertia weight (TIW) is tested on seven (7) benchmark optimization functions. These functions are commonly used optimization test functions for testing the effectiveness of optimization algorithms. The test function used in the experiment are given in Table 1. The proposed method is benchmarked against some of the most efficient weight definitions in the current literature (random inertia weight (RIW), Linearly Decreasing Inertia Weight (LDIW) and Exponential Based Sigmoid Function (ESIW)).

RESULTS AND DISCUSSION

Results for Benchmark Functions

In the conducted experiments, a comparative analysis was performed on the performance of different Particle Swarm Optimization (PSO) variations across multiple benchmark functions detailed in Table 1. The metrics considered for evaluation were the minimum cost, mean cost, mean time, and standard deviation of cost. The comparison Test Results of the Benchmark Functions are presented in Table 2. The convergence curves are also presented in Figures 6 – 11. Among the tested PSO variations, the proposed TIW (Tanh-based Adaptive Inertia Weight) consistently outperformed the others regarding minimum cost. This indicates that the proposed TIW algorithm was successful in finding better solutions compared to the standard PSO, LDIW (linearly decreasing inertia weight), RIW (random inertia weight), and ESIW (exponentially decreasing inertia weight) variations. The lower minimum cost obtained by the proposed TIW demonstrates its effectiveness in optimizing the objective function. However, it is worth noting that the standard PSO exhibited the lowest mean time, as expected. This is because the standard PSO does not involve any additional computations associated with an adaptive inertia weight, as it employs constant inertia throughout the optimization process. Therefore, it had a time advantage over the other variations.

For the specific case of the sum square benchmark function, the proposed TIW was the second-best performing variation, with the LDIW achieving the lowest minimum cost. This suggests that the linearly decreasing inertia weight was particularly effective for this function, surpassing the proposed TIW in finding optimal solutions. Overall, the results indicate that the proposed TIW has the potential to improve optimization performance in most cases, yielding lower minimum costs. However, other variations, such as LDIW, might perform better depending on the specific benchmark function. The selection of the most appropriate PSO variation may vary depending on the optimisation problem's characteristics.

Convergence Rates

From the convergence curves in Figures 6 -10, the proposed TIW generally outperformed the other variations in terms of convergence, except for the sum square function, which was second to the LDIW. Across most benchmark functions, the convergence curves of the proposed TIW showed significant improvements over iterations, indicating a rapid convergence towards the optimal solution. The proposed TIW exhibited faster convergence compared to the Standard PSO, RIW (random inertia weight), and ESIW (exponentially decreasing inertia weight) variations. It consistently achieved lower costs in shorter iterations, demonstrating its effectiveness in finding optimal solutions.

It is important to note that the performance of each PSO variation can be influenced by the characteristics of the optimised benchmark function. While the proposed TIW generally showed superior convergence behaviour, choosing the most suitable PSO variation may depend on the specific optimization problem. Analysing the convergence curves alongside other performance metrics provides a comprehensive understanding of the effectiveness and efficiency of each PSO variation.

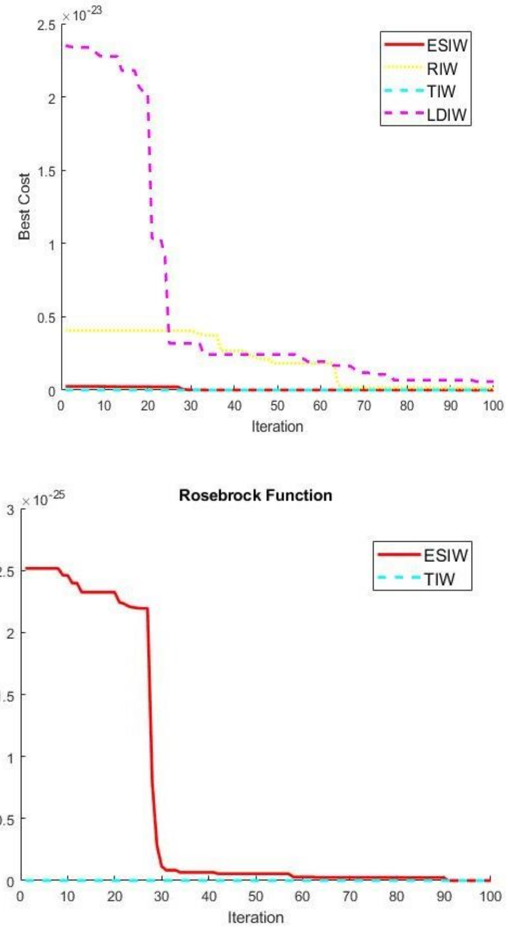

Figure 6. Convergence rate of the Rosenbrock function

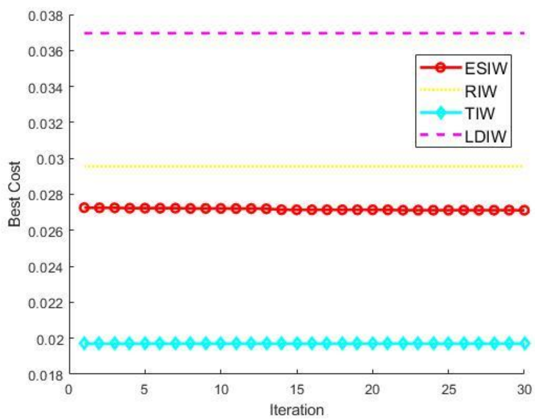

Figure 7. Convergence rate of the Grienwank function.

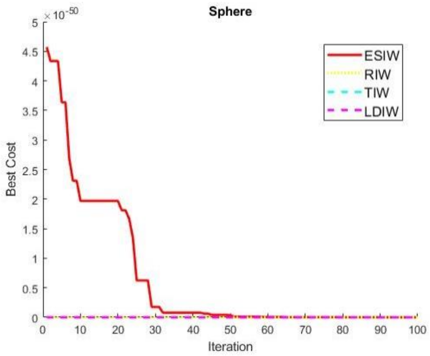

Figure 8a. *Convergence rate of the Sphere function.*

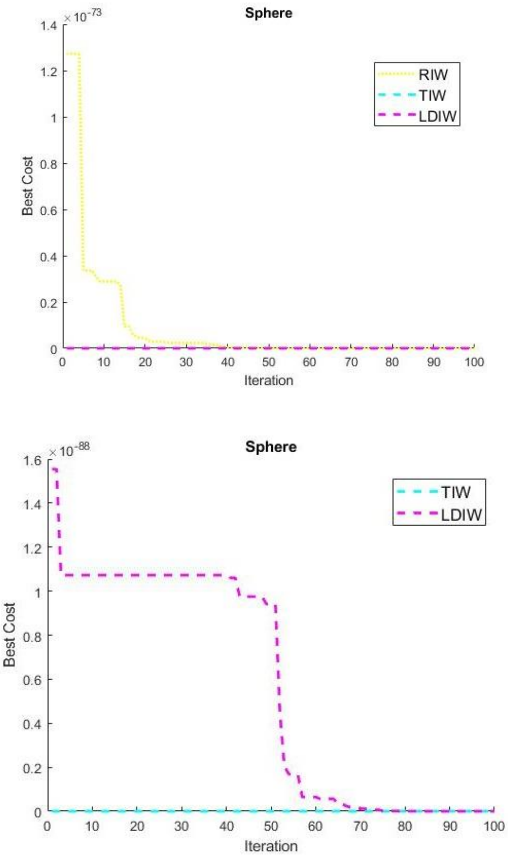

Figure 8b. *Convergence rate of the Sphere function.*

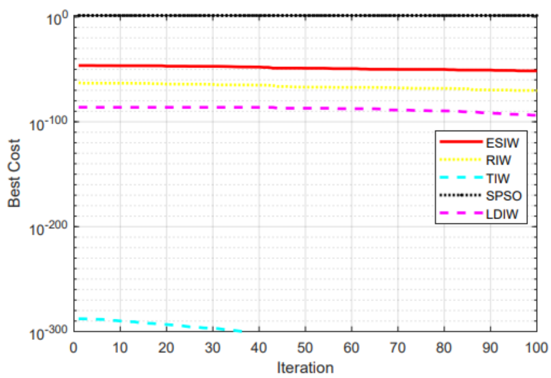

Figure 9. Convergence rate of the Rotated Hyper Ellipsoid

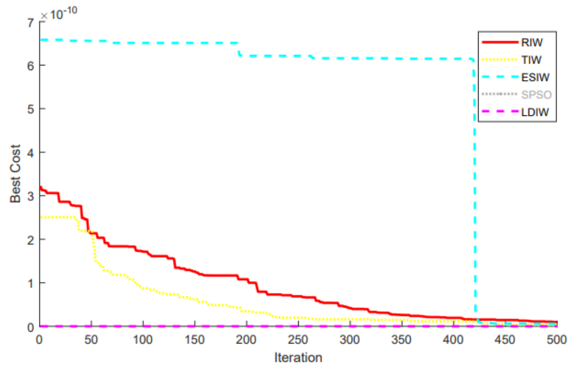

Figure 10. Convergence Rate of the Sum Squares Benchmark Function

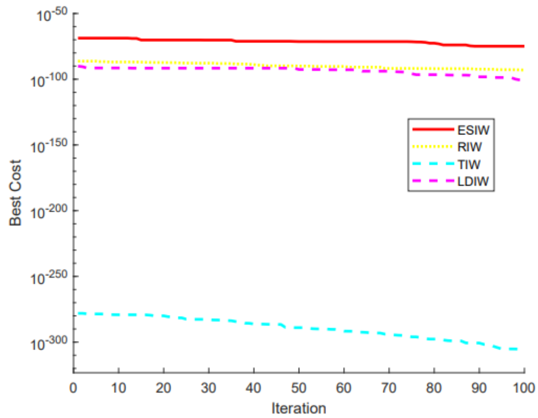

Figure 11. Convergence Rate of the Matyas Benchmark Function

CONCLUSION

In conclusion, the study presented a novel approach for implementing adaptive inertia weight (AIW) in particle swarm optimization (PSO) by proposing an AIW function based on the hyperbolic tangent function. The proposed Tanh-based AIW-PSO outperformed standard PSO and other PSO variations on multiple benchmark functions in terms of minimum cost, mean cost, and standard deviation of cost. Furthermore, the proposed AIW-PSO demonstrated a faster convergence rate, as seen in the plots of best cost against iteration. These results demonstrate the effectiveness of using an adaptive inertia weight function in PSO for optimizing problems and suggest that using a hyperbolic tangent function for the AIW can lead to improved performance over linear or exponential sigmoid-based functions.

https://doi.org/10.25077/jnte.v12n2.1095.2023

Table 2: Minimum Cost, Mean Time, Mean Cost and Standard Deviation of Cost

100 Iterations, Search Population = 100					
Function	PSO variation used	Minimum Cost	Mean Time (s)	Mean Cost	Standard Deviation
Sum Squares	Standard PSO	22457.3914	**0.35302**	22457.3914	**3.6471e-12**
	LDIW	**1.1436e-11**	0.43326	**2.6706e-10**	1.2998e-10
	RIW	0.0016951	0.36337	0.0062843	0.0027468
	ESIW	0.025385	0.38421	0.046912	0.013776
	Proposed TIW	1.1087e-08	0.44208	1.4375e-08	3.1999e-09
Sphere	Standard PSO	3.4008	**0.074412**	3.4008	4.4633e-16
	LDIW	1.0763e-96	0.10878	5.6467e-89	5.2826e-89
	RIW	4.3514e-82	0.083148	8.9787e-75	2.5887e-74
	ESIW	1.4959e-54	0.11286	6.4887e-51	1.168e-50
	Proposed TIW	**0**	0.10405	**1.1762e-302**	**0**
Colville	Standard PSO	37.9895	**0.158801**	37.9895	1.4282e-14
	LDIW	0.00089039	0.24183	0.0010316	0.00083281
	RIW	0.00017275	0.18214	0.00018119	6.1178e-06
	ESIW	0.0011013	0.2037	0.0016746	0.00015295
	Proposed TIW	**3.0931e-05**	0.21368	**3.531e-05**	**3.3358e-06**
Matyas	Standard PSO	0.0014141	0.20969	0.0014141	**0**
	LDIW	8.232e-102	0.27651	4.4225e-97	1.7279e-96
	RIW	8.4558e-96	0.21945	7.4358e-87	2.1348e-86
	ESIW	5.9617e-74	0.2414	6.4579e-68	1.2615e-67
	Proposed TIW	**9.2734e-313**	0.26959	**1.8097e-283**	**0**
Rosenbrock	Standard PSO	1.9332	0.092476	1.9332	4.4633e-16
	LDIW	2.6006e-23	0.11384	1.4511e-20	2.0773e-20
	RIW	0.029562	0.086774	2.0834e-23	6.8316e-04
	ESIW	1.9287e-24	0.10707	7.5005e-24	5.0951e-05
	Proposed TIW	**6.7066e-29**	0.091724	**1.9778e-27**	2.7802e-27
Grienwank	Standard PSO	0.31656	0.026602	0.51753	3.3876e-16
	LDIW	0.036961	0.038015	0.036961	1.0225e-07
	RIW	0.029562	0.031251	0.029562	6.8316e-04
	ESIW	0.027118	0.043049	0.027174	5.0951e-05
	Proposed TIW	**0.019697**	0.032109	**0.019697**	**2.0271e-17**
Rotated hyper ellipsoid	Standard PSO	21.5454	0.10784	21.5454	3.5706e-15
	LDIW	1.939e-94	0.10111	2.6704e-87	2.8291e-87
	RIW	4.7038e-71	**0.090095**	1.1874e-64	2.5673e-64
	ESIW	3.4177e-52	0.10225	7.5185e-48	1.2752e-47
	Proposed TIW	**0**	0.10846	**1.7013e-310**	0

REFERENCES

[1] H. Farshi and K. Valipour, “Hybrid PSO-GA Algorithm for Automatic Generation Control of Multi-Area Power System,” *IOSR J. Electr. Electron. Eng. Ver. I*, vol. 11, no. 1, pp. 2278–1676, 2016, doi: 10.9790/1676-11111829.

[2] Z. H. Zhan, J. Zhang, Y. Li, and H. S. H. Chung, “Adaptive particle swarm optimization,” *IEEE Trans. Syst. Man, Cybern. Part B Cybern.*, vol. 39, no. 6, pp. 1362–1381, 2009, doi: 10.1109/TSMCB.2009.2015956.

[3] D. C. Diana and S. P. Joy Vasantha Rani, “Modified inertia weight approach in PSO algorithm to enhance MMSE Equalization,” *2021 4th Int. Conf. Electr. Comput. Commun. Technol. ICECCT 2021*, 2021, doi: 10.1109/ICECCT52121.2021.9616720.

[4] M. Alhussein and S. I. Haider, “Improved particle swarm optimization based on velocity clamping and particle penalization,” *Proc. - AIMS 2015, 3rd Int. Conf. Artif. Intell. Model. Simul.*, pp. 61–64, 2016, doi: 10.1109/AIMS.2015.20.

[5] S. Sun, “Cloud computing resource scheduling based on improved particle swarm optimization algorithm,” *J. Phys. Conf. Ser.*, vol. 2023, no. 1, pp. 669–674, 2021, doi: 10.1088/1742-6596/2023/1/012025.

[6] M. A. Arasomwan and A. O. Adewumi, “On the performance of linear decreasing inertia weight particle swarm optimization for global optimization,” *Sci. World J.*, vol. 2013, pp. 5–16, 2013, doi: 10.1155/2013/860289.

[7] S. Zdiri, J. Chrouta, and A. Zaafouri, “Cooperative multi-swarm particle swarm optimization based on adaptive and time-varying inertia weights,” *2021 IEEE 2nd Int. Conf. Signal, Control Commun. SCC 2021*, pp. 200–207, 2021, doi: 10.1109/SCC53769.2021.9768349.

[8] S. Zdiri, J. Chrouta, and A. Zaafouri, “Inertia weight strategies in Multiswarm Particle swarm Optimization,” *Proc. Int. Conf. Adv. Syst. Emergent Technol. IC_ASET 2020*, pp. 266–271, 2020, doi: 10.1109/IC_ASET49463.2020.9318226.

[9] M. Fan and Y. Akhter, “A Time-Varying Adaptive Inertia Weight based Modified PSO Algorithm for UAV Path Planning,” *Int. Conf. Robot. Electr. Signal Process. Tech.*, no. February, pp. 573–576, 2021, doi: 10.1109/ICREST51555.2021.9331101.

[10] X. Yu, J. Liu, and H. Li, “An adaptive inertia weight particle swarm optimization algorithm for IIR digital filter,” *2009 Int. Conf. Artif. Intell. Comput. Intell. AICI 2009*, vol. 1, no. 1, pp. 114–118, 2009, doi: 10.1109/AICI.2009.28.

[11] W. Liu, Z. Wang, N. Zeng, Y. Yuan, F. E. Alsaadi, and X. Liu, “A novel randomised particle swarm optimizer,” *Int. J. Mach. Learn. Cybern.*, vol. 12, no. 2, pp. 529–540, 2021, doi: 10.1007/s13042-020-01186-4.

[12] X. Wang, Y. Duan, B. Xiao, Z. Zhou, H. Peng, and Y. Wang, “Hybrid Solar Power System Optimization based on Multi-Objective PSO Algorithm,” *Proc. - 2019 Chinese Autom. Congr. CAC 2019*, pp. 4176–4180, 2019, doi: 10.1109/CAC48633.2019.8997401.

[13] E. O. Addo, E. Twumasi, and D. Kwegyir, “Improvement of Particle Swarm Optimization Using Personal Best Adaptive Weight,” vol. 6, no. 9, pp. 852–858, 2021.

[14] T. Mohamed, H. Abubakr, M. Hussein, and G. Shabib, “Load Frequency Controller Based on Particle Swarm Optimization for Isolated Microgird System,” *Int. J. Appl. Energy Syst.*, vol. 1, no. 2, pp. 69–75, 2019, doi: 10.21608/ijaes.2019.169953.

[15] S. S. Dhillon, J. S. Lather, and S. Marwaha, “Multi Area Load Frequency Control Using Particle Swarm Optimization and Fuzzy Rules,” *Procedia Comput. Sci.*, vol. 57, pp. 460–472, 2015, doi: 10.1016/j.procs.2015.07.363.

[16] P. Satapathy, J. Sahu, M. K. Debnath, and P. K. Mohanty, “PDPID plus DDF cascaded controller for LFC investigation in unified system with wind generating unit,” *1st Odisha Int. Conf. Electr. Power Eng. Commun. Comput. Technol. ODICON 2021*, pp. 6–11, 2021, doi: 10.1109/ODICON50556.2021.9428967.

[17] H. Merah, A. Gacem, D. Benattous, Y. Labbi, and O. P. Malik, “Solving economic dispatch problem using a new hybrid PSO-ALO algorithm,” *CCSSP 2020 - 1st Int. Conf. Commun. Control Syst. Signal Process.*, pp. 487–492, 2020, doi: 10.1109/CCSSP49278.2020.9151630.

[18] E. T. Zapanta and R. V. M. Santiago, “Load Frequency Control of Interconnected Power System Using Particle Swarm Optimization Based Disturbance Observer-Enhanced PI Controller,” pp. 696–701, 2023, doi: 10.1109/cicn56167.2022.10008353.

[19] T. M. Shami, A. A. El-Saleh, M. Alswaitti, Q. Al-Tashi, M. A. Summakieh, and S. Mirjalili, “Particle Swarm Optimization: A Comprehensive Survey,” *IEEE Access*, vol. 10, pp. 10031–10061, 2022, doi: 10.1109/ACCESS.2022.3142859.

[20] Y. -l. Gao, X. -h. An and J. -m. Liu, "A Particle Swarm Optimization Algorithm with Logarithm Decreasing Inertia Weight and Chaos Mutation," 2008 International Conference on Computational Intelligence and Security, Suzhou, China, 2008, pp. 61-65, doi: 10.1109/CIS.2008.183.