Journal of Electronics and Electrical Engineering
https://ojs.wiserpub.com/index.php/JEEE/

Research Article

An Enhanced Particle Swarm Optimization Algorithm via Adaptive Dynamic Inertia Weight and Acceleration Coefficients

Yaw O. M. Sekyere*, Francis B. Effah, Philip Y. Okyere

Department of Electrical and Electronic Engineering, Kwame Nkrumah University of Science and Technology, Kumasi, Ghana
E-mail: yawsekyere@gmail.com

Received: 6 November 2023; **Revised:** 17 December 2023; **Accepted:** 27 December 2023

Abstract: The particle swarm optimization (PSO) algorithm counts among the most popular metaheuristic algorithms based on swarm intelligence. Since the publication of the first article on this optimization technique, researchers have developed many PSO variants with some improvement in its performance. The PSO optimization performance hinges on its ability to achieve a good exploration-exploitation balance. The most common method that helps to improve exploration-exploitation balance is modifying the PSO three controlling parameters, namely the inertia weight and two acceleration coefficients. In this paper a PSO variant that combines adaptive dynamic inertia weight and adaptive dynamic acceleration coefficients to enhance the exploration-exploitation balance of the PSO is proposed. The enhanced PSO algorithm called Adaptive Dynamic Inertia Weight and Acceleration Coefficient Optimization (ADIWACO) algorithm is tested on seven well-known standard test functions comprising four unimodal and three multimodal ones. The performance of the PSO is then compared with that of the standard PSO (SPSO) and four existing PSO variants. The experimental results comprising optimum value, runtime, mean value, standard deviation and convergence rate, and confirmed by the results of ranking statistics and Wilcoxon signed rank test conducted on the experimental results, indicate significantly better performance by the proposed PSO algorithm.

***Keywords*:** particle swarm optimization (PSO), metaheuristics, inertia weight, acceleration coefficient

1. Introduction

Particle swarm optimization (PSO) is a well-established swarm optimization algorithm that has proven to provide high performance in many application areas which require that optimization problems be solved [1, 2, 3]. In the engineering fields, it has been applied in robotics [4], power systems [5, 6] solar energy systems [7] image processing [8, 9, 10] control [11, 12], wireless communication networks [13, 14], training of artificial neural networks [15, 16], and many others. The original or standard PSO (SPSO) algorithm has few parameters to tune, is easy to program and has shown excellent optimization performance. However, it suffers the disadvantage of converging prematurely in complex problems of high dimension [1, 16, 17]. The PSO performs better when there is better balance between exploration and exploitation. Since its introduction in 1995, researchers have been developing PSO variants to improve its performance through the creation of a proper exploration-exploitation balance. The common research areas on PSO algorithm development can be broadly classified into three groups: hybridizing PSO with other well-known metaheuristic algorithms, neighborhood topological structure and modification of controlling parameters [18].

DOI: https://doi.org/10.37256/jeee.3120243868

There are many hybrid variants which combine the particle swarm with evolutionary algorithms to surmount its drawbacks including premature convergence and poor solution quality. The PSO hybridized with genetic algorithm (GA), for instance, is known to provide better performance with regard to the quality of the solution and speed of convergence as compared with the individual PSO and GA [12] and hybridization with differential evolution (DE) has shown to be effective in providing high solution quality and efficient computation [19]. Other algorithms which have been hybridized with PSO include ant colony optimization (ACO) [20], gravitational search algorithm (GSA) [21], grey wolf optimizer (GWO) [3, 22], and simulated annealing (SA) [23].The harnessing of the strengths of the PSO algorithm and the evolutionary algorithms that it is combined with has proved to be a good strategy to maintain a good exploration-exploitation balance, and thus preventing the population from becoming stagnant and the algorithm from converging prematurely [18, 24].

Particles in a swarm are bonded in some form of structure usually referred to as a neighborhood topology within which there is communication and sharing of information between particles [1]. Studies have shown that the topology if properly selected can effectively enhance the PSO performance [25]. Therefore to enhance the PSO performance, various topologies have been proposed: the star topology, ring topology [26], the Von Neumann topology [27], the dynamic topology [24], complex network and others. In the original PSO algorithm, every particle sees all other particles as its neighbors [1]. The PSO is said to be using a star topology. It has the fastest convergence but converges to local optima [1]. The ring topology prevents the algorithm to converge to local optima but its convergence speed decreases. Many test cases have shown that the performance of the Von Neumann topology is superior to that of others [27]. In the dynamic topology, the neighborhood of each particle is updated by dynamically selecting particles that are closest to the current particle. A PSO variant based on this topology was able to find multiple Pareto optimal solutions [28]. The complex neighborhood topology uses a complex neighborhood network to prevent the algorithm from converging prematurely and increase its exploration capabilities. A proposed complex neighborhood particle swarm optimizer for solving global optimization problems performed better than the original PSO in every test problem [29].

The PSO has, in general, three controlling parameters: inertia weight, and two acceleration coefficients. It has been shown in the literature that the PSO performance is very sensitive to all these three controlling parameters [30, 31]. Therefore, a proper setting of these parameters is necessary to achieve good performance. Researchers have developed PSO variants that improve the performance of the SPO by using various techniques to modify the controlling parameters. Velocity clamping is one of the early important ones used to modify the PSO controlling parameters [32]. Proper choice of the velocity clamping parameter prevents the velocity of the particles from diverging and hence promoting convergence towards the optimal solution. However, improper choice leads to poor performance [1]. The concept of constriction factors [15, 33, 34, 35, 36], consisting of a diverse range of mathematical formulas in the velocity update equation, has also been applied to ensure convergence without using velocity clamping [15].

The inertia weight has been shown by experimental studies to be the most important controlling parameter of the PSO [18, 23]. For this reason, many recent variants are based on modification of this controlling parameter. There are three techniques in the literature for changing this parameter: the adaptive, the time-varying or dynamic, and the random techniques. The adaptive inertia weight strategies that use the distances of the particles to their personal best and global best positions have been widely proposed in the literature [37, 38, 39, 40]. Random and chaotic theory based inertia weights have also been proposed with some degree of success [32, 41]. Most PSO variants based on modification of inertia weight use the popular time-varying techniques [1]. Time-varying logarithm, exponential, trigonometry and linear functions have been used with some degree of success [1, 16]. The sigmoid function has also been combined with a linearly increasing inertia weight [42]. This showed improvement in convergence speed and violent movement that narrows towards the solution region [42]. Other weight modulation techniques include a logarithm decreasing inertia weight combined with a chaos mutation operator to improve the speed of convergence and the ability to come out of the local optima [7, 43], an exponent decreasing inertia weight combined with stochastic piecewise mutation resulting in an improved PSO that prevents the algorithm from converging prematurely and having later period oscillatory occurrences [7], and a random inertia weight, where a random inertia weight is used at each iteration. The random inertia weight is found to be best suited for applications in a dynamic environment where whether large or small inertia weight is required cannot be easily predicted [1]. A recent PSO variant in [17] combines a time-varying hyperbolic tangent function with an adaptive factor which makes use of the difference between the global and personal best positions. This variant showed good performance when tested on seven well-known benchmark functions.

PSO variants based on acceleration coefficients have not fully been explored. Using the time-varying technique to change the acceleration coefficients is among the few in the literature on PSO using acceleration coefficients that vary with time [44, 45]. The study in [45] indicated significantly better performance if acceleration coefficients that vary with time are used rather than fixed ones.

Researchers continue to look for new concepts to improve the PSO performance. One recent concept attracting the attention of researchers is the opposition-based learning concept [28, 29, 30, 31]. The opposition-based learning PSO (OLPSO) applies opposition-based learning method to enhance the swarm's diversity by considering both original and opposite positions during optimization [19]. OLPSO has demonstrated superior performance in providing high convergence speed and solution quality. Its main drawbacks are increased computational complexity and sensitivity to parameter tuning which may impact convergence times and robustness [46].

Although the existing PSO variants have shown promising results when used to solve optimization problems, they continue to converge prematurely in some complex problems that have high dimensions [47, 48]. Therefore, the PSO algorithm is still being developed further so that it can perform well on complex real-world optimization problems [16, 48, 49]. Modification of the controlling parameters has become a more popular method in the literature to enhance the PSO performance [50]. The method has the advantage of providing enhancement of the standard PSO without much compromising its main merit of simplicity in terms of implementation and coding. The work of the authors in [17] uses adaptive dynamic inertia weight. This paper seeks to enhance further its performance by adding adaptive dynamic acceleration coefficients. The combination of the two techniques aims to synergistically harness the strengths of both dynamic adaptive inertia weight and dynamic adaptive acceleration coefficients to improve the PSO capability of exploiting and exploring the search space.

The remaining part of this paper is structured as follows: Section 2 introduces the enhanced PSO (ADIWACO) algorithm. Section 3 details the testing procedures while the results and discussion are presented in Section 4. Section 5 presents the conclusion of the study.

2. Enhanced PSO (ADIWACO)

The standard PSO is a bioinspired metaheuristic optimization technique formulated on the collective behavior of bird flocks or fish schools. In the PSO, a population of potential solutions, represented as particles, explores the solution space by adjusting their velocities and positions based on their own best-known solutions and the globally best solution found by the entire population [51]. Mathematically, the velocity, v and position, x of each particle are updated iteratively by the following equations:

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

$$v_i(t+1) = wv_i(t) + c_1(p_i(t) - x_i(t)) + c_2(g(t) - x_i(t)) \tag{2}$$

where w = the inertia weight. c_1, c_2 = acceleration coefficients, $x_i(t)$ = the current position of a particle, $x_i(t+1)$ = the updated position of a particle, $p_i(t)$ = the personal best of a particle, $g(t)$ = the global best of a particle, $v_i(t)$ = the velocity of a particle and $v_i(t+1)$ = updated velocity of the particle with the updated position $x_i(t+1)$ [51].

The coefficient c_1, also called the cognitive constant, enables each particle to return to its previous best position for effective local search, while the coefficient c_2, also called the social constant, causes the particle to move towards the overall best position of the swarm, based on its proximity. The inertia weight w controls the exploration-exploitation balance during the search process [17, 51].

The standard PSO uses constant controlling parameters. The proposed algorithm enhances its performance by employing adaptive dynamic inertia weight and also adaptive dynamic acceleration coefficients. This is an improvement of work done in [17] where only an adaptive dynamic inertia weight was used.

The adaptive dynamic inertia weight is calculated using the formula:

$$w = \mu \tanh \delta \tag{3}$$

where μ and δ are defined as follows:

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

$$\delta = W_{\max} - \frac{(W_{\max} - W_{\min}) \times the\,number\,of\,the\,current\,iteration}{Maximum\,number\,of\,iteration} \tag{5}$$

The adaptive coefficient μ is given by the scaled difference between the particle best and the global best positions. If the best cost of a particle is significantly better than the global best cost, the inertia coefficient increases, allowing for greater exploration to discover potentially better solutions. Conversely, if the best cost of a particle is close to the global best cost, the inertia weight decreases, facilitating exploitation to refine the current best solution. The time-varying or dynamic component of the inertia weight δ decreases linearly from W_{max} at the start of the iteration to W_{min} at the maximum number of iterations. The hyperbolic tangent function is applied to δ to scale it to a range between 0 and 1. The function gives a smoother transition from the maximum inertia weight value to the minimum inertia weight value as the iteration increases compared with the linearly-decreasing inertia weight component alone.

Figure 1 shows a typical variation of the inertia weight with time for each of these three cases: $w = \delta$, $w = tanh\ \delta$ and $w = \mu tanh\ \delta$. From the figure, the tanh-based inertia weight ($w = tanh\ \delta$) exhibits a smoother transition compared with linearly decreasing inertia weight ($w = \delta$), thus enhancing the particles capability of exploiting and exploring the search space. It is also observed that if the dynamic factor $tanh\ \delta$ is multiplied by the adaptive factor μ, a performance-dependent inertia weight is obtained. This further enhances the capabilities of the particles to explore and exploit the search space.

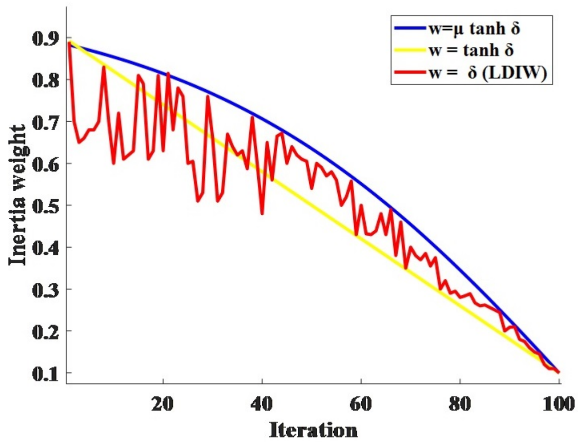

Figure 1. Behaviour of different Inertia weight functions during the optimization process.

These dynamic adaptive acceleration coefficients c_1 and c_2 have the same value as recommended in the literature [51]. The acceleration coefficients are obtained at every iteration by the equation:

$$c_1 = c_2 = \mu \cosh \psi \tag{6}$$

where:

$$\psi = C_{\max} - \frac{(C_{\max} - C_{\min}) \times the\,number\,of\,the\,current\,iteration}{maximum\,number\,of\,iteration} \tag{7}$$

The adaptive coefficient, μ is given by (4)

Applying the *cosh* function to ψ produces a smooth transition from the maximum acceleration coefficient C_{max} to the minimum acceleration coefficient C_{min} as the number of iterations increases. This gradual change in values enables a controlled and stable optimization process, contributing to more reliable results [52]. Figure 2 shows a typical variation of the acceleration factors with time. The curve indicates that it is also performance-dependent.

The flow chart for the proposed PSO is presented in Figure 3.

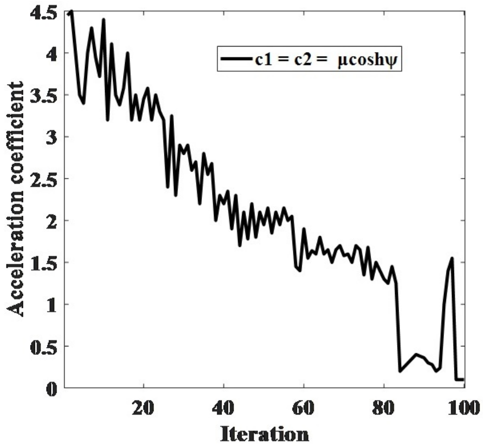

Figure 2. Behaviour of Acceleration Coefficient function.

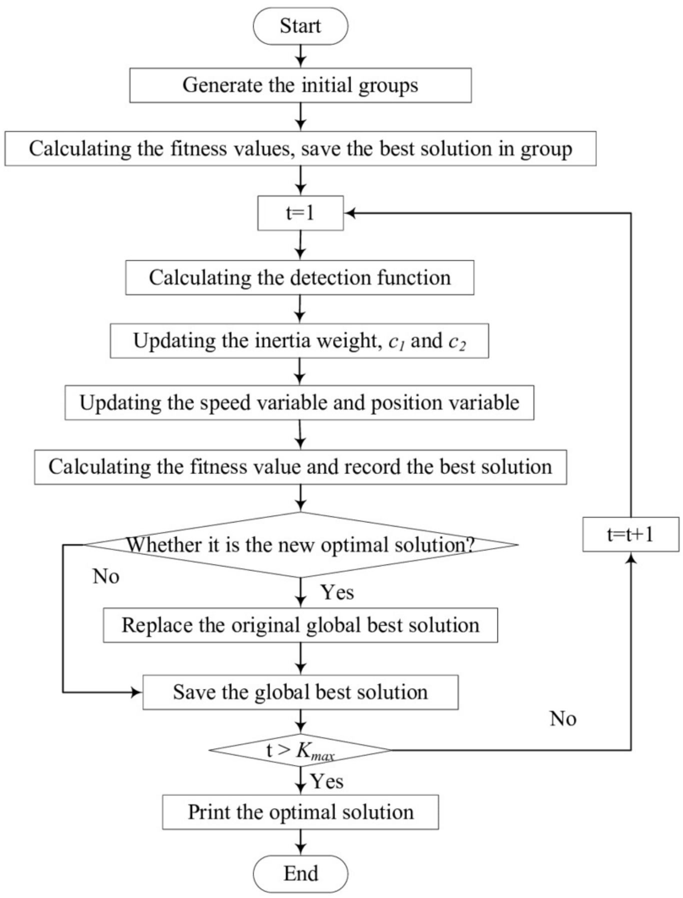

Figure 3. Flow chart for the enhanced PSO.

3. Testing

The performance evaluation of the enhanced PSO algorithm (ADIWACO) is performed on seven (7) common benchmark optimization functions using MATLAB R2023a. The computer setup used for the testing comprises the following specifications: Windows 11 (64-bit) for the software environment, and an Intel(R) Core (TM) i5-8250U CPU @ 1.60GHz 1.80 GHz with 24.0 GB installed RAM for the hardware environment. The results are compared with those obtained using the standard PSO and four variants of the PSO in the literature, namely Tanh-based Adaptive Inertia Weight PSO (TIW) [17], Randomised Adaptive Inertia Weight PSO (RIW) [53], Linearly Decreasing Inertia Weight PSO (LDIW) [54] and Exponential Sigmoid function based PSO (ESIW) [39]. These variants are all based on modification of the PSO controlling parameters.

3.1 *Benchmark optimization functions*

The test functions are defined in (8)–(14) and their details presented in Table 1.

$$g_1(x_1, x_2, \dots, x_n) = \sum_{i=1}^{N} i \cdot x_i^2 \tag{8}$$

$$g_2(x_1, x_2, \dots, x_n) = \sum_{i=1}^{N} x_i^2 \tag{9}$$

$$\begin{aligned} g_3(x_1, x_2, \dots, x_n) = {} & 100\left(x_1^2 - x_2\right)^2 + (x_1 - 1)^2 + (x_3 - 1)^2 + 90\left(x_3^2 - x_4\right)^2 \\ & + 10.1\left[(x_2 - 1)^2 + (x_4 - 1)^2\right] + 19.8\,(x_2 - 1)(x_4 - 1) \end{aligned} \tag{10}$$

$$g_4(x_1, x_2, \dots, x_n) = 0.26\left(x_1^2 + x_2^2\right) - 0.48\,x_2 x_3 \tag{11}$$

$$g_5(x_1, x_2, \dots, x_n) = \sum_{i=1}^{N-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + (1 - x_i)^2\right] \tag{12}$$

$$g_6(x_1, x_2, \dots, x_n) = \frac{1}{4000} \sum_{i=1}^{N} x_1^2 - \prod_{i=1}^{N} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \tag{13}$$

$$g_7(x_1, x_2, \dots, x_n) = \sum_{i=1}^{N} \left(\sum_{j=1}^{N} x_j\right)^2 \tag{14}$$

Table 1. Details of benchmark functions.

Function	Function ID	Function Name	Search Range	Dimension	Optimum Value	Description
$g_1(x_1, x_2, \dots, x_n)$	$g1$	Sum Square	[−10,10]	5	0	Unimodal
$g_2(x_1, x_2, \dots, x_n)$	$g2$	Sphere	[−100, 100]	20	0	Unimodal
$g_3(x_1, x_2, \dots, x_n)$	$g3$	Colville	[−10,10]	4	0	Multimodal
$g_4(x_1, x_2, \dots, x_n)$	$g4$	Matyas	[−10, 10]	2	0	Unimodal
$g_5(x_1, x_2, \dots, x_n)$	$g5$	Rosenbrock	[−5, 10]	20	0	Multimodal
$g_6(x_1, x_2, \dots, x_n)$	$g6$	Greiwank	[−600, 600]	2	0	Multimodal
$g_7(x_1, x_2, \dots, x_n)$	$g7$	Rotated Hyper-Ellipsoid	[−65.536,65.536]	2	0	Unimodal

The following parameters were used for all algorithms:
Population of search particles = 500
Maximum number of iterations = 50
Maximum number of runs = 10
The following parameters were also required by the proposed algorithm.
$W_{max} = 1$, $W_{min} = 0.1$, $C_{max} = 5$ and $C_{min} = 2$.

Initial global best = infinity.
Initial personal best = random (randomly generated within the search range).

3.2 *Experimental procedure*

For each benchmark function, the following metrics were obtained for each of the algorithms: optimum solution value which measures the algorithm's efficiency in discovering solutions of high quality, runtime which measures the algorithm's speed, standard deviation which measures the algorithm's stability and consistency, mean value which provides an average performance assessment and convergence curves. Ranking statistics were performed on the experimental results for intuitive comparison of the performance of the algorithms. Also Wilcoxon signed rank test was done to verify if the differences in performance between ADIWACO and the comparison algorithms were significant.

4. Results and discussion

The experimental results on the benchmark functions are compared in Table 2. The best values are shown in bold. The convergence curves are also presented.

Table 2. Optimal Values, Runtime, Mean Value and Standard Deviation.

50 Iterations, Search Population = 500							
Function ID	**Function Name**	**PSO Variant**	**Optimum Value**	**Runtime (s)**	**Mean Value**	**Standard Deviation**	**Rank**
$g1$	Sum Squares	Standard PSO [51]	0.011974	**0.1897**	0.011974	**0**	6
		LDIW-PSO [54]	3.6375e-44	0.4669	3.8038e-44	5.3537e-44	5
		TIW-PSO [17]	1.2828e-85	0.55056	4.7153e-79	1.8057e-78	3
		EIW-PSO [39]	5.6509e-69	0.67845	4.9053e-67	6.1438e-66	4
		RIW-PSO [53]	**6.2609e-100**	0.62898	3.0053e-103	7.9238e-90	1
		Proposed ADIWACO	1.3756e-95	0.69173	**6.0703e-90**	**0**	2
$g2$	Sphere	Standard PSO [51]	1.4335	**0.04287**	1.4335	1.7853e-15	6
		LDIW-PSO [54]	7.5184e-48	0.44477	6.4696e-44	6.4696e-44	4
		TIW-PSO [17]	1.8038e-81	0.71201	1.5331e-74	5.9791e-74	**2**
		EIW-PSO [39]	7.5184e-39	0.45277	6.5972e-41	5.2345e-40	5
		RIW-PSO [53]	1.5341e-72	0.45093	1.5385e-72	1.8946e-74	**3**
		Proposed ADIWACO	**3.5882e-202**	0.72157	**1.8744e-182**	**0**	1
$g3$	Colville	Standard PSO [51]	48.7237	**0.38764**	48.7237	1.4355e-14	6
		LDIW-PSO [54]	0.045487	0.43577	0.050952	0.0031712	4
		TIW-PSO [17]	0.0056674	0.46682	0.0065689	0.00048376	2
		EIW-PSO [39]	0.017414	0.47653	0.018262	0.00047017	3
		RIW-PSO [53]	0.077708	0.47088	0.077708	2.7689e-07	5
		Proposed ADIWACO	**0.002699**	0.46119	**0.002978**	**5.7239e-19**	1
$g4$	Matyas	Standard PSO [51]	0.0014141	**0.20969**	0.0014141	**0**	6
		LDIW-PSO [54]	9.9112e-101	0.47751	4.2021e-94	4.8056e-94	2
		TIW-PSO [17]	1.4105e-79	0.36962	2.2164e-71	4.1205e-71	4
		EIW-PSO [39]	2.2774e-39	0.47291	2.0715e-37	2.565e-37	5
		RIW-PSO [53]	2.5468e-69	0.4088	2.9856e-75	7.2203e-65	3
		Proposed ADIWACO	**7.9052e-145**	0.45486	**1.6897e-133**	**3.529e-132**	1
$g5$	Rosenbrock	Standard PSO [51]	1.9332	**0.092476**	1.9332	4.4633e-16	6
		LDIW-PSO [54]	1.1327e-11	0.49021	1.887e-11	9.3791e-12	2
		TIW-PSO [17]	1.1514e-10	0.29827	1.3823e-09	2.5782e-09	4
		EIW-PSO [39]	8.3127e-14	0.49021	5.9887e-11	8.1231e-13	3
		RIW-PSO [53]	2.5784e-09	0.27102	2.5802e-09	1.9955e-12	5
		Proposed ADIWACO	**2.9105e-16**	0.42517	**3.6707e-16**	**1.4889e-16**	**1**
$g6$	Grienwank	Standard PSO [51]	0.31656	**0.26602**	0.049257	2.4487e-17	5
		LDIW-PSO [54]	**0.012316**	0.38015	0.028097	6.0025e-17	2
		TIW-PSO [17]	0.012321	0.31251	**0.012321**	2.5069e-14	1
		EIW-PSO [39]	0.087564	0.345256	0.098765	0.0999876	6
		RIW-PSO [53]	0.012987	0.38455	0.028949	1.5225e-17	3
		Proposed ADIWACO	0.13164	0.63721	0.034633	**2.661e-17**	4
$g7$	Rotated hyper ellipsoid	Standard PSO [51]	21.5454	0.10784	21.5454	3.5706e-15	6
		LDIW-PSO [54]	2.6935e-116	0.038439	4.8917e-108	2.3852e-108	2
		TIW-PSO [17]	5.6116e-67	0.038801	1.2383e-61	3.0207e-61	3
		EIW-PSO [39]	6.4766e-10	0.046644	9.2983e-9	1.7149e-9	5
		RIW-PSO [53]	7.1195e-29	**0.022611**	5.5181e-27	9.6907e-29	4
		Proposed ADIWACO	**7.3597e-203**	0.044595	**4.7303e-183**	**0**	1

4.1 *Comparison of optimal value, standard deviation, mean value and runtime*

From Table 2, ADIWACO produces the best optimum values for the functions $g2$, $g3$, $g4$, $g5$ and $g7$. Apart from the multimodal, low dimensional function $g6$, the optimum values obtained by ADIWACO for all functions including unimodal high-dimensional function $g2$ and multimodal, high-dimensional function $g5$ are close to the theoretical optimum values. In terms of the standard deviation, ADIWACO produces the best values for all functions apart from $g4$. SPSO obtains the same standard deviation value of zero as ADIWACO for the function $g1$ and the best standard deviation value for the function $g4$. The results clearly show that ADIWACO is more efficient and stable or robust than all the other comparison PSO variants. ADIWACO also outperformed all the algorithms regarding the mean value for all the functions except for $g6$ indicating a superior optimization performance.

The average value of the runtime of the 6 algorithms in seconds are compared in Figure 4. The average value of the runtime of the standard PSO is the least among the 6 algorithms (0.185 s), which means that SPSO is the fastest. ADIWACO has the highest average runtime of 0.491 s. This is not surprising since the standard PSO algorithm uses constant controlling parameters and the comparison PSO variants modify only the inertia weight whereas ADIWACO modifies all the three controlling parameters. The rise in ADIWACO's computational time by about 126% over that of the standard PSO can be justified by its significant improvement in the SPSO performance.

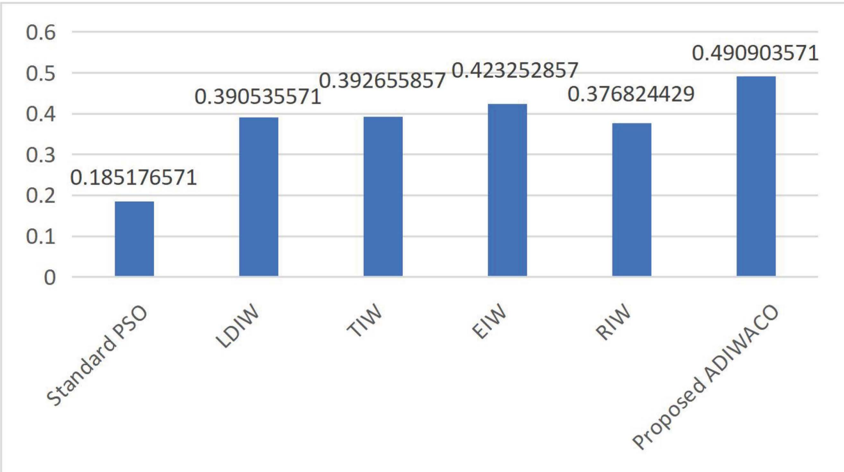

Figure 4. Average runtime of algorithms in seconds.

4.2 *Convergence curves*

From the convergence curves in Figures 5–11, ADIWACO shows the fastest convergence for all the functions except $g6$ where its convergence speed is the second highest. For the function $g6$, it is the convergence speed of LDIW that is highest. ADIWACO, vis-a-vis the other algorithms, shows consistency in convergence speed and on the whole exhibits the best convergence performance.

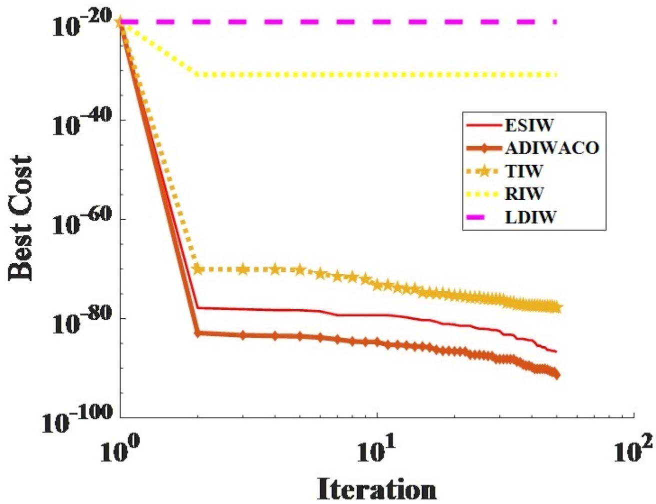

Figure 5. Optimization performance for the Sum of squares function.

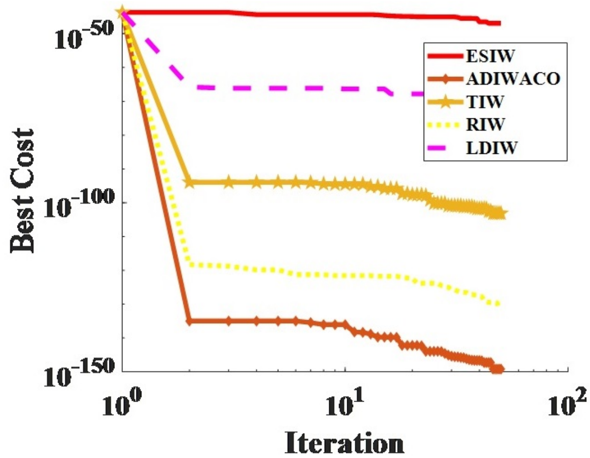

Figure 6. Optimization performance for the Sphere function.

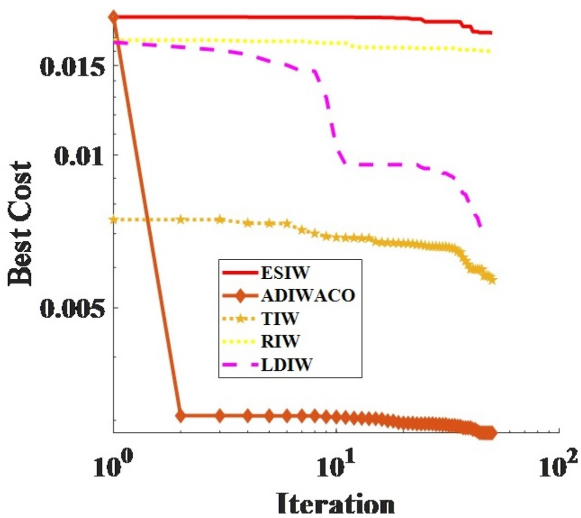

Figure 7. Optimization performance for the Coville function.

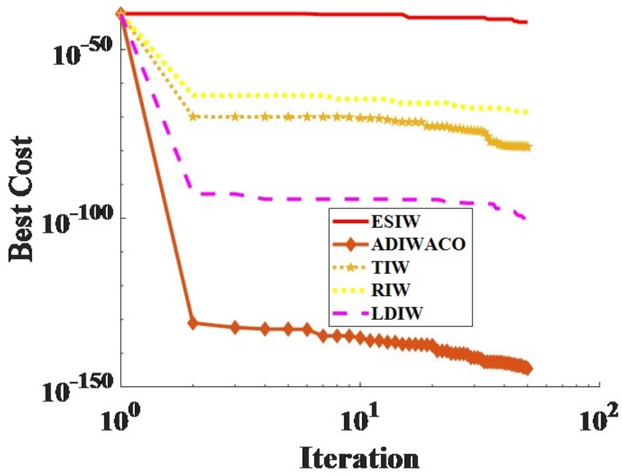

Figure 8. Optimization performance for the Matyas function.

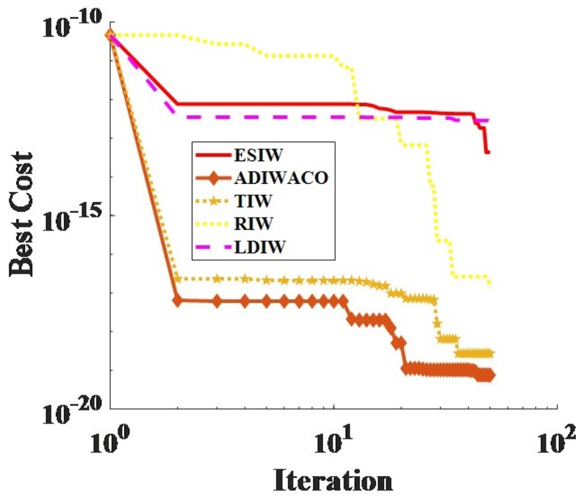

Figure 9. Optimization performance for the Rosenbrock function.

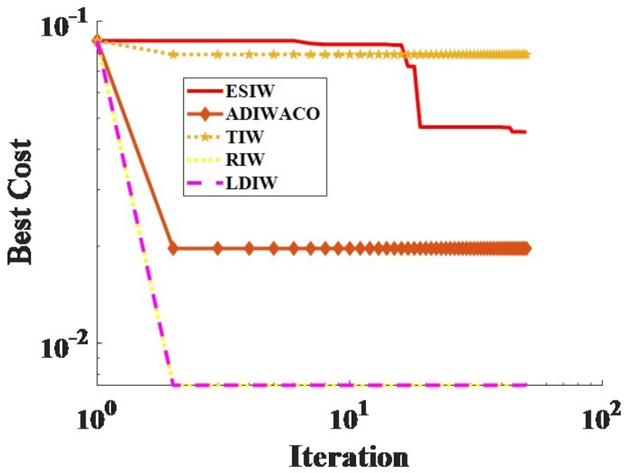

Figure 10. Optimization performance for the Griewank function.

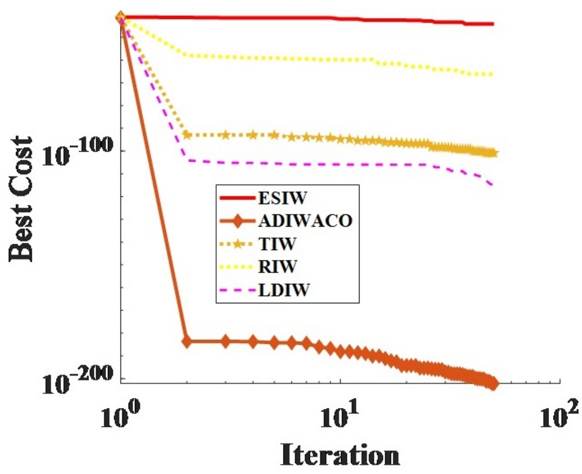

Figure 11. Optimization performance for the Rotated hyper-ellipsoid function.

4.3 *Ranking statistics*

The mean and standard deviation values are used to obtain the ranking of each algorithm as in [24]. The better the mean value of an algorithm for a function, the higher is its ranking. Where the mean values of two algorithms are the same, the one with a better standard deviation value is given a higher ranking. If their standard deviation values are also the same, the two algorithms are given the same ranking. The rankings of the algorithms on each function are included in Table 2. Figure 12 shows the ranking statistics and Table 3 gives its summary. ADIWACO ranks first in 5 functions ($g2$, $g3$, $g4$, $g5$ and $g7$). It obtains a total rank of 11, representing an average rank of 1.571 which is the best among all the algorithms. With this result, ADIWACO shows superior optimization performance over the comparison algorithms.

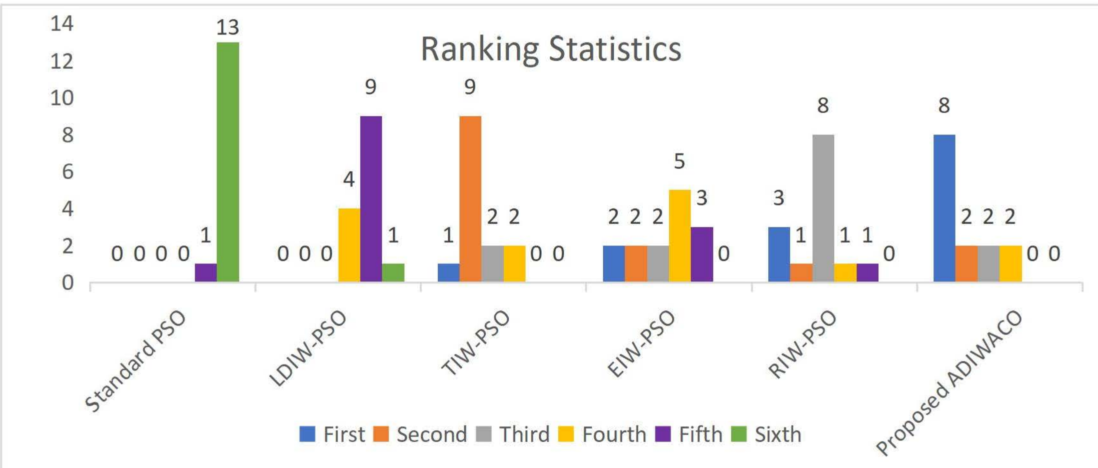

Figure 12. Ranking statistics of experimental results.

Table 3. Summary of ranking statistics.

Rank	Standard PSO	LDIW-PSO	TIW-PSO	EIW-PSO	RIW-PSO	ADIWACO
First	0	0	1	0	1	5
Second	0	4	2	0	0	1
Third	0	0	2	1	2	0
Fourth	0	2	2	2	2	1
Fifth	1	1	0	3	2	0
Sixth	6	0	0	1	0	0
Total	41	21	19	32	25	11
Average	5.857	3.000	2.714	4.571	3.571	1.571

4.4 *Wilcoxon signed rank test*

To confirm if the difference in optimization performance between ADIWACO and the comparison PSO variants is significant, the Wilcoxon signed rank test was performed. The significance level of the test was set at 0.05. The test data were taken from Table 2 and the results obtained presented in Table 3. R^+ indicates positive rank, R^- negative rank. In the symbol $n/w/t/l$, n is the number of the test functions, w is the number of functions in which ADIWACO showed better performance, t is the number of functions where the ADIWACO and the comparison variant showed equal performance, and l is the number of functions in which ADIWACO showed poorer performance. From Table 4, the p-Values of SPSO and a recently proposed ESIW [39] are all less than 0.05 indicating that they are inferior to ADIWACO. The remaining three PSO variants (LDIW, TIW and RIW) have p-Values greater than 0.05. However, in the 7 benchmark functions, ADIWACO is superior to LDIW in 6 functions and inferior in 1 function. The corresponding figures are 6 and 1 when compared to TIW, and 5 and 2 when compared to RIW. These results sufficiently prove that ADIWACO outperforms these three variants.

Table 4. Wilcoxon signed rank statistical analysis.

	h	p-Value	R^+	R^-	*n/win/tie/loss*
ADIWACO vs standard PSO	+	0.0003	30	0	7/7/0/0
ADIWACO vs LDIW PSO	-	0.1305	12	2	7/6/0/1
ADIWACO vs TIW PSO	-	0.1924	11	3	7/6/0/1
ADIWACO vs ESIW PSO	+	0.0228	20	0	7/7/0/0
ADIWACO vs RIW PSO	-	0.2586	15	2	7/5/0/2

5. Conclusions

This paper has proposed a new PSO variant (ADIWACO) that synergistically harnesses the strengths of both adaptive dynamic inertia weight and adaptive dynamic acceleration coefficients to improve the PSO capability of exploiting and exploring the search space. The mathematical formulas for the three controlling parameters are based on hyperbolic trigonometric functions which yield smooth changes in their values during the iteration to improve the PSO performance. The proposed PSO variant is tested on seven benchmark functions including three multimodal ones and its performance compared to that of the standard PSO and four PSO variants using optimum value, mean value, standard deviation and runtime as the performance metrics. ADIWACO obtains the best optimum value for 5 out of 7 functions including the high-dimensional unimodal function $g2$ and the high-dimensional multimodal function $g5$ indicating high efficiency in high-dimensional and multimodal optimization problems. Its mean and standard deviation values are also the best for 6 out of 7 functions demonstrating higher optimization performance and better stability vis-a-vis the other PSO variants. In the Ranking statistics and Wilcoxon signed rank test conducted on the experimental results, ADIWACO proves to be superior over the comparison algorithms in optimization performance. ADIWACO, outperforming the comparison PSO variants in terms of optimal values, mean values, stability and convergence rate, is seen as a more promising PSO variant for tackling complex optimization problems.

In the near future, the proposed algorithm will be used to optimally tune PID controllers for load frequency control in interconnected power systems as an example of a real-world engineering application.

Conflict of interest

There is no conflict of interest for this study.

References

[1] 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, https://doi.org/10.1109/ACCESS.2022.3142859.

[2] J. Safarik, and V. Snasel, "Acceleration of Particle Swarm Optimization with AVX Instructions," *Appl. Sci.*, vol. 13, p. 734, 2023, https://doi.org/10.3390/app13020734.

[3] U. Hassan, and A. Ahmad, "Optimal Dispatch of Distributed Generators in Multiple Microgrids Using Hybrid PSO-GWO," in *Proc. ICECE 2023*, Lahore, Pakistan, Mar. 15–16, 2023, https://doi.org/10.1109/ICECE58062.2023.10092500.

[4] G. M. Nayeem, M. Fan, and Y. Akhter, "A Time-Varying Adaptive Inertia Weight based Modified PSO Algorithm for UAV Path Planning," in *Proc. ICREST 2021*, Dhaka, Bangladesh, Jan. 5–7, 2021, https://doi.org/10.1109/ICREST51555.2021.9331101.

[5] Z. Xin-Gang, L. Ji, M. Jin, and Z. Ying, "An improved quantum particle swarm optimization algorithm for environmental economic dispatch," *Expert Syst. Appl.*, vol. 152, p. 113370, 2020, https://doi.org/10.1016/j.eswa.2020.113370.

[6] W. Elsayed, Y. G. Hegazy, M. S. El-Bages, and F. M. Bendary, "Improved Random Drift Particle Swarm Optimization With Self-Adaptive Mechanism for Solving the Power Economic Dispatch Problem," *IEEE Trans. Ind. Inform.*, vol. 13, pp. 1017–1026, 2017, https://doi.org/10.1109/TII.2017.2695122.

[7] R. Krishnasamy, R. Aathi, and P. Jeyabalan, "Application of Comprehensive Learning Particle Swarm Optimization to Least Cost Generation Expansion Planning Problem with Solar Plant," in *Proc. INCCES 2019*, Krishnankoil, India, Dec. 18–20 2019, https://doi.org/10.1109/INCCES47820.2019.9167689.

[8] M. Hajihassani, D. J. Armaghani, and R. Kalatehjari, "Applications of Particle Swarm Optimization in Geotechnical Engineering: A Comprehensive Review," *Geotech. Geol. Eng.*, vol. 36, pp. 705–722, 2017, https://doi.org/10.1007/s10706-017-0356-z.

[9] J. T. Machado, S. M. A. Pahnehkolaei, and A. Alfi, "Complex-order particle swarm optimization," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 92, p. 105448, 2020, https://doi.org/10.1016/j.cnsns.2020.105448.

[10] R. Malik, R. Dhir, and S. K. Mittal, "Remote sensing and landsat image enhancement using multiobjective PSO based local detail enhancement," *J. Ambient. Intell. Humaniz. Comput.*, vol. 10, pp. 3563–3571, 2018, https://doi.org/10.1007/s12652-018-1082-y.

[11] T. Mohamed, H. Abubakr, M. M. Hussein, and G. Shabib, "Load Frequency Controller Based on Particle Swarm Optimization for Isolated Microgird System," *Int. J. Appl. Energy Syst.*, vol. 1, pp. 69–75, 2019, https://doi.org/10.21608/ijaes.2019.169953.

[12] H. Farshi, and K. Valipour, "Hybrid PSO-GA algorithm for automatic generation control of multi-area power system," *IOSR J. Electr. Electron. Eng.*, vol. 11, pp. 18–29, 2016, https://doi.org/10.9790/1676-11111829.

[13] R.-Y. Wang, Y.-T. Hsiao, and W.-P. Lee, "A new cooperative particle swarm optimizer with dimension partition and adaptive velocity control," in *Proc. IEEE SMC 2012*, Seoul, Korea, Oct. 14–17, 2012, https://doi.org/10.1109/ICSMC.2012.6377684.

[14] M. Suriya, "Machine learning and quantum computing for 5G/6G communication networks—A survey," *Int. J. Intell. Netw.*, vol. 3, pp. 197–203, 2022, https://doi.org/10.1016/j.ijin.2022.11.004.

[15] M. Swathy, and C. A. Babu, "Opposition Based Constriction Factor Particle Swarm Optimization for Economic Load Dispatch," in *Proc. ICAECT 2022*, Bhilai, India, Apr. 21–22, 2022, https://doi.org/10.1109/ICAECT54875.2022.9807910.

[16] A. G. Gad, "Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review," *Arch. Comput. Methods Eng.*, vol. 29, pp. 2531–2561, 2022, https://doi.org/10.1007/s11831-021-09694-4.

[17] Y. O. M. Sekyere, F. B. Effah, and P. Y. Okyere, "Hyperbolic Tangent—Based Adaptive Inertia Weight Particle Swarm Optimization," *JNTE*, vol. 12, no. 2, 2023, https://doi.org/10.25077/jnte.v12n2.1095.2023.

[18] J. Gou, Y. -X. Lei, W. -P. Guo, C. Wang, Y. -Q. Cai, and W. Luo, "A novel improved particle swarm optimization algorithm based on individual difference evolution," *Appl. Soft Comput.*, vol. 57, pp. 468–481, 2017, https://doi.org/10.1016/j.asoc.2017.04.025.

[19] H. -H. Xu, and R. -L. Tang, "Particle swarm optimization with adaptive elite opposition-based learning for large-scale problems," in *Proc. ICCIA 2020*, Beijing, China, Jun. 19–21, 2020, https://doi.org/10.1109/ICCIA49625.2020.00016.

[20] G. Chen, Z. Li, Z. Zhang, and S. Li, "An Improved ACO Algorithm Optimized Fuzzy PID Controller for Load Frequency Control in Multi Area Interconnected Power Systems," *IEEE Access*, vol. 8, pp. 6429–6447, 2019, https://doi.org/10.1109/ACCESS.2019.2960380.

[21] D. K. Gupta, A. V. Jha, B. Appasani, A. Srinivasulu, N. Bizon, and P. Thounthong, "Load Frequency Control Using Hybrid Intelligent Optimization Technique for Multi-Source Power Systems," *Energies*, vol. 14, p. 1581, 2021, https://doi.org/10.3390/en14061581.

[22] S. Sharma, R. Kapoor, and S. Dhiman, "A Novel Hybrid Metaheuristic Based on Augmented Grey Wolf Optimizer and Cuckoo Search for Global Optimization," in *Proc. ICSCCC 2021*, Jalandhar, India, May 21–23, 2021, https://doi.org/10.1109/ICSCCC51823.2021.9478142.

[23] M. Abdel-Baset, and I. Hezam, "A Hybrid Flower Pollination Algorithm for Engineering Optimization Problems," *Int. J. Comput. Appl.*, vol. 140, pp. 10–23, 2016, https://doi.org/10.5120/ijca2016909119.

[24] X. Zhang, X. Wang, Q. Kang, and J. Cheng, "Differential mutation and novel social learning particle swarm optimization algorithm," *Inform. Sci.*, vol. 480, pp. 109–129, 2018, https://doi.org/10.1016/j.ins.2018.12.030.

[25] B. Mohanty, and P. K. Hota, "Particle swarm optimization based interconnected Hydro-Thermal AGC system considering GRC and TCPS," *Int. J. Electr. Comput. Eng.*, vol. 8, pp. 1195–1201, 2014.

[26] 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, https://doi.org/10.1016/j.procs.2015.07.363.

[27] J. Kennedy, and R. Mendes, "Population structure and particle swarm performance," in *Proc. CEC 2002*, Honolulu, HI, USA, May 12–17, 2002, https://doi.org/10.1109/cec.2002.1004493.

[28] A. Mohammadi, H. Asadi, S. Mohamed, K. Nelson, and S. Nahavandi, "Multiobjective and Interactive Genetic Algorithms for Weight Tuning of a Model Predictive Control-Based Motion Cueing Algorithm," *IEEE Trans. Cybern.*, vol. 49, pp. 3471–3481, 2018, https://doi.org/10.1109/tcyb.2018.2845661.

[29] A. Godoy, and F. J. Von Zuben, "A Complex Neighborhood based Particle Swarm Optimization," in *Proc. IEEE CEC 2009*, Trondheim, Norway, May 18–21, 2009, https://doi.org/10.1109/CEC.2009.4983016.

[30] A. M. Eltamaly, "A Novel Strategy for Optimal PSO Control Parameters Determination for PV Energy Systems," *Sustainability*, vol. 13, p. 1008, 2021, https://doi.org/10.3390/su13021008.

[31] K. R. Harrison, A. P. Engelbrecht, and B. M. Ombuki-Berman, "Optimal parameter regions and the time-dependence of control parameter values for the particle swarm optimization algorithm," *Swarm Evol. Comput.*, vol. 41, pp. 20–35, 2018, https://doi.org/10.1016/j.swevo.2018.01.006.

[32] M. H. Mojarrad, and P. Ayubi, "Particle swarm optimization with chaotic velocity clamping (CVC-PSO)," in *Proc. IKT 2015*, Urmia, Iran, May 26–28, 2015, https://doi.org/10.1109/IKT.2015.7288811.

[33] U. K. Acharya, and S. Kumar, "Particle swarm optimization exponential constriction factor (PSO-ECF) based channel equalization," in *Proc. INDIACom 2019*, New Delhi, India, Mar. 13–15, 2019.

[34] S. H. Shri, A. F. Mijbas, and M. J. Mohammed, "Constriction Factor Based Particle Swarm Optimization for Solving Reactive Power Optimization Problem," in *Proc. ICECCME 2022*, Maldives, Nov. 16–18, 2022, https://doi.org/10.1109/ICECCME55909.2022.9988572.

[35] R. P. Patwardhan, and S. L. Mhetre, "Effect of constriction factor on minimization of transmission power loss using Particle Swarm Optimization," in *Proc. ICESA 2015*, Pune, India, Oct. 30–Nov. 1, 2015, https://doi.org/10.1109/ICESA.2015.7503330.

[36] M. Li, J. Tu, and Z. Zhou, "An improved constriction factor PSO to prevent premature convergence," in *Proc. CCC 2010*, Beijing, China, Jul. 29–31, 2010, pp. 5177–5181.

[37] D. C. Diana, and S. P. J. V. Rani, "Modified inertia weight approach in PSO algorithm to enhance MMSE Equalization," in *Proc. ICECCT 2021*, Erode, India, Sep. 15–17, 2021, https://doi.org/10.1109/ICECCT52121.2021.9616720.

[38] Z.-H. Zhan, J. Zhang, Y. Li, and H. S.-H. Chung, "Adaptive Particle Swarm Optimization," *IEEE Trans. Syst. Man, Cybern. B Cybern.*, vol. 39, no. 6, pp. 1362–1381, 2009, https://doi.org/10.1109/tsmcb.2009.2015956.

[39] W. Liu, Z. Wang, Y. Yuan, N. Zeng, K. Hone, and X. Liu, "A Novel Sigmoid-Function-Based Adaptive Weighted Particle Swarm Optimizer," *IEEE Trans. Cybern.*, vol. 51, no. 3, pp. 1085–1093, 2019, https://doi.org/10.1109/tcyb.2019.2925015.

[40] E. O. Addo, E. Twumasi, and D. Kwegyir, "Improvement of Particle Swarm Optimization Using Personal Best Adaptive Weight," *Int. J. Innov. Sci. Technol.*, vol. 6, pp. 852–858, 2021.

[41] W. Wang, and L. Qiu, "Optimal Reservoir Operation Using PSO with Adaptive Random Inertia Weight," in *Proc. AICI 2010*, Sanya, China, Oct. 23–24, 2010, https://doi.org/10.1109/AICI.2010.316.

[42] R. F. Malik, T. A. Rahman, S. Z. M. Hashim, and R. Ngah, "New particle swarm optimizer with sigmoid increasing inertia weight," *Int. J. Comput. Sci. Inf. Technol. Secur.*, vol. 1, pp. 35–44, 2007.

[43] K. Akter, L. Nath, T. A. Tanni, A. S. Surja, and M. S. Iqbal, "An Improved Load Frequency Control Strategy for Single & Multi-Area Power System," in *Proc. ICAEEE*, Gazipur, Bangladesh, Feb. 24–26, 2022, https://doi.org/10.1109/ICAEEE54957.2022.9836416.

[44] Q. M. Abdo, H. Ewad, and K. A. Mohamed, "Optimized PID Controller for Single Area Thermal Power System Based on Time Varying Acceleration Coefficients Particle Swarm optimization," in *Proc. ICCCEEE*, Khartoum, Sudan, Feb. 26–Mar. 1, 2021, https://doi.org/10.1109/ICCCEEE49695.2021.9429670.

[45] A. Ratnaweera, S. Halgamuge, and H. Watson, "Self-Organizing Hierarchical Particle Swarm Optimizer With Time-Varying Acceleration Coefficients," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 240–255, 2004, https://doi.org/10.1109/TEVC.2004.826071.

[46] M. G. H. Omran, and S. Al-Sharhan, "Using opposition-based learning to improve the performance of particle swarm optimization," in *Proc. IEEE SIS*, St. Louis, MO, USA, Sep. 21–23, 2008, https://doi.org/10.1109/SIS.2008.4668288.

[47] N. Zeng, H. Zhang, W. Liu, J. Liang, and F. E. Alsaadi, "A switching delayed PSO optimized extreme learning machine for short-term load forecasting," *Neurocomputing*, vol. 240, pp. 175–182, 2017, https://doi.org/10.1016/j.neucom.2017.01.090.

[48] F. Wang, J. Li, Z. Li, D. Ke, J. Du, C. Garcia, and J. Rodriguez, "Design of Model Predictive Control Weighting Factors for PMSM Using Gaussian Distribution-Based Particle Swarm Optimization," *IEEE Trans. Ind. Electron.*, vol. 69, no. 11, pp. 10935–10946, 2021, https://doi.org/10.1109/TIE.2021.3120441.

[49] H. Chen, L. Xiang, L. Lin, and S. Huang, "Fractional-order PID Load Frequency Control for Power Systems Incorporating Thermostatically Controlled Loads," in *Proc. IEEE CIEEC*, Wuhan, China, May 28–30, 2021, https://doi.org/10.1109/CIEEC50170.2021.9510621.

[50] S. Zdiri, J. Chrouta, and A. Zaafouri, "Inertia weight strategies in Multiswarm Particle swarm Optimization," in *Proc. IC ASET*, Hammamet, Tunisia, Dec. 15–18, 2020, https://doi.org/10.1109/IC_ASET49463.2020.9318226.

[51] J. Kennedy and R. Eberhart, "Particle Swarm Optimization," in *Proc. ICNN*, Perth, Australia, Nov. 27–Dec. 1, 1995, https://doi.org/10.1109/ICNN.1995.488968.

[52] X. Shen, Z. Chi, J. Yang, and C. Chen, "Particle Swarm Optimization with Dynamic Adaptive Inertia Weight," in *Proc. CESCE 2010*, Wuhan, China, Mar. 6–7, 2010, https://doi.org/10.1109/CESCE.2010.16.

[53] G. Yue-lin, and D. Yu-hong, "A new particle swarm optimization algorithm with random inertia weight and evolution strategy," in *Proc. CISW*, Harbin, China, Dec. 15–19, 2007, https://doi.org/10.1109/CISW.2007.4425479.

[54] 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. 1–12, 2013, https://doi.org/10.1155/2013/860289.