ISSN No:-2456-2165

Improvement of Particle Swarm Optimization Using Personal Best Adaptive Weight

[1]Emmanuel Obuobi Addo
Department of Electrical and Electronic Engineering,
Kwame Nkrumah University of Science and Technology
Kumasi, Ghana

[2]Elvis Twumasi
Department of Electrical and Electronic Engineering,
Kwame Nkrumah University of Science and Technology
Kumasi, Ghana

[3]Daniel Kwegyir
Department of Electrical/Electronic Engineering
Kwame Nkrumah University of Science and Technology
Kumasi, Ghana

Abstract:- Improvement of the particle swarm optimization algorithm has become increasingly important to deliver it out of local optima trapping and increase its convergence rate. In this paper a personal best adaptive weight is proposed as a new PSO variant named personal best adaptive weight particle swarm optimization (PBAW-PSO) to choose different inertia weight for different particles in the swarm to update their velocity. The proposed variant was compared with three other inertia weight improved variants on six benchmark functions. The comparison was done based on the best cost, mean cost, simulation time, standard deviation and convergence rate. The overall results showed that the PBAW-PSO variant had a better performance than the other variants.

Keywords:- *Metaheuristic; Inertia Weight; Evolutionary; Particle Swarm Optimization; Convergence.*

I. INTRODUCTION

Over the past decades, a lot of evolutionary optimization techniques have been developed to help optimize problems in Finance, Science and Engineering [1-3]. These evolutionary algorithms have gained much attention amongst researchers due to their high computational efficiency. Popular among these metaheuristic algorithms are particle swarm optimization, genetic algorithm, bee colony optimization among others [4-7]. The particle swarm optimization has been considered as one of the best heuristics because of its simplicity, easy implementation, flexibility and robustness to control parameters and high computational efficiency [5-9]. Also, the PSO is noted for its superior convergence characteristics. However, many researchers have argued that the accuracy of this technique can still be improved. According to [8,9], the particle swarm optimization must be improved in other to further optimize their efficiency, enhance overall search performance, improve its convergence rate, alleviate premature convergence and prevent local trapping in local optima.

Key areas in the metaheuristic processes of the PSO identified by researchers over the years to improve to ensure better accuracy and faster convergence include, the optimal selection of the acceleration constants and selection of inertia weights [10-12]. The paper in [13,14] argued that the inertia weight used in updating the velocities of each particle is one of the most important aspect of the PSO therefore any improvement on it should focus on this parameter. In the original PSO, a constant inertia weight value is used throughout the iteration. Therefore, many researchers have developed various variants of the inertial weight to help improve the particle swarm optimization algorithm. The work in [15] used a constant inertia weight value. The paper in [16] used a trigonometric function to improve the dynamic changes of inertia weight with time. Reference [17] also used a time varying sigmoid increasing inertia weight to optimize the classical PSO. The work in [18] used a random selection of the inertia weight as an improvement to the original PSO. In [19] the random components of inertia weight were generated from Beta distribution. The paper [12] used a double exponential based dynamic inertia weight PSO that linearly decreases the weight after each iteration.

It is realized from the above that, inertia weight variants of the PSO developed over the years have either used, a constant inertia weight, a random inertia weight, a time varying inertia weight or an adaptive inertia weight [15-22]. In the constant weight adjustment, a constant value is chosen within a minimum and maximum value for all the particles to use to update its velocity. In this regard, a high inertia weight constant value is normally chosen to ensure the exploration of the particles and a lower value is normally set for the exploitation of the particles in the search space. However, there remains uncertainty for users to choose a balance value. Also, a randomly generated inertia weight values uses a randomly generated constant weight value in computing the particles velocity. The random selection of the weight supports the exploratory search in the beginning of the PSO optimization process and also increases the population diversity during the search process [20]. However, because the inertia weight is altered randomly, the algorithm may deviate from the optimum solution. Concerning the time varying inertia weight, the weight of the velocity function is changed after each iteration. This would allow a little variation of the particle's velocity after each iteration. However, changing the weights after every iteration but use this same weight value for the entire swarm to update their velocities does not yield good performance because the tendency of each particle to move is different from each other. An adaptive inertia weight modification approach changes each particles inertia weight in response to the parameters of the particles.

Although the adaptive PSO variants have shown some competitiveness in its performance there is always a trade off in either ensuring that the algorithm is delivered out of local optima at the expense of convergence rate or the enhancement ensure a satisfactory convergence rate over the local optima problem. This is certainly undesirable. As such, it is important to develop a state-of-the-art PSO inertia weight variant that is capable of ensuring a satisfactory solution to all the deficiencies inherent in the conventional PSO. This paper addresses this need by developing a personal best adaptive weight that prevents the PSO from local trapping and improve convergence rate.

The rest of the sections of this paper are organized as follows. Section II explains the concept of the particle swarm optimization algorithm, Section III presents the proposed personal best adaptive weight used in giving each particle in the swarm, an inertia weight during each iteration, Section IV analyzes the results of the proposed personal best adaptive PSO as compared with other variants of the inertial weight improved PSO on six benchmark functions. Section V draws the underlining conclusion of the paper.

II. PARTICLE SWARM OPTIMIZATION

The particle swarm optimization is a simple and efficient metaheuristic algorithm which was put forward by Kennedy and Eberhart to mimic the social behavior of birds flocking [23]. According to the technique, there exist an imaginary communication between a swarm of birds searching for food in a field in other to locate the best location of food. In the application of this swarm intelligence, each bird is referred to as a particle. The field is also referred to as the search space, each location that each particle will move to for food in the search space is known as the position of the particle. Among each location that each particle has been to before, the particles have memory of its best location, meaning that each particle in the swarm has the ability to remember its best place that has a lot of food than all the places it has visited and this is known as the particles personal best position.

After each visit to a location, there is an assumption of a swarm intelligence that allows the particles to communicate with each other and move towards the position whose location has a lot of food than all of them. This best location among them is known as the global best of all the particles. In other for each particle to move towards the global best position, each particle will change their current location by updating their velocity considering their personal best and global best. In changing the velocity of each particle there is consideration of the particles, personal position and global position.

Because the intuition of these particles is not certain, there are two random values, r1 and r2 that brings a little randomness in their movement. Also, there are two constants known as the cognitive and social constants that helps each particle to account for the impact of each particle's individual information and the impact of the group of particles respectively. That is the cognitive component (C1) which allows the particle to reappear to its position to ensure a good local search and the social component (C2) whihc encourages the particle to travel to the direction of the overall best position of the swarm, knowledgeable by its vicinity. The velocity of the particle is updated using (1) and the new position of each particle is updated using (2) till the optimum solution is reached.

$$V_i(t+1) = wV_i(t) + r_1c_1(P_{ibest} - X_i(t)) + r_2c_2(G_{best} - X_i(t)) \quad (1)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (2)$$

III. PROPOSED PERSONAL BEST ADAPATIVE WEIGHT

To optimally improve the classical particle swarm optimization, a personal best adaptive weight (PBAW) is introduced. Because inertia weight considers the tendency of a particle to remain in its position of rest or continue moving, the inertia weight of the PSO that considers the behavior of the particle at the previous time step will serve as a determinant for a particle's accurate inertia value at the current time step. Thus, this new variant uses the individual particles' best position and the global position of the swarm to alternate different exploration and exploitation of different particle at each iteration since these positions will give a clear indication of the tendency of each particle to move towards the optimum solution. This will therefore allow the particles to exploit in a confined area to search for promising solution and at the same time explore different regions of the search space during each iteration. The new adaptive weight is defined in this work as (3)

$$w = \frac{0.2 * \|P_{best_i}\|}{\|G_{best(iter)}\|} \quad (3)$$

where P_{best_i} is the personal best of particle i and G_{best_i} is the global best at iteration $iter$.

The proposed weight (PBAW) is tested on six (6) benchmark optimization functions. These functions are commonly used optimization test functions for testing the effectiveness of optimization algorithms. The test function used were taken from [24]. The test function used in the experiment are Rosenbrock, Rastrigin, Dekkers-Aarts, Michwalze, Greiwank and Baele. The details of the test functions are shown table 1. All selected functions are multimodal except Rosenbrock. Unimodal functions have only one optimum value. On the other hand, the optimum value of multimodal functions increases with the dimension of the function. Hence multimodal functions are complex and difficult to obtain optimum solutions. These help to assess the strength of the new proposed weight definition since they present varied level of difficulty. Rastrigin, Dekkers-Aarts, Greiwank, and Baele are evaluated on 2-dimensions whilst Michalewicz and Rosenbrock are evaluated on a dimension of 10 and 50 respectively. The optimum value of all functions is 0 except Deckkers-Aarts and Michalewicz with optimum values of −24771.09 and -9.66015 respectively.

Table 1:Description of benchmark datasets

Function name	Limits	Dimension	Global optima solution	Description
Rosenbrock	[-5,10]	50	0	Unimodal
Rastrigin	[-5.12,5.12]	2	0	Multimodal
Dekkers-Aarts	[-20,20]	2	−24771.09	Multimodal
Michalewicz	$[0, \pi]$	10	-9.66015	Multimodal
Greiwank	[-600.600]	2	0	Multimodal
Baele	[-4.5,4.5]	2	0	Multimodal

A. Benchmarking of Proposed Personal Best Adaptive Weight

The proposed method is benchmarked against three most efficient weight definitions in the current literature (random inertia weight, chaotic inertia weight and modified inertia weight). The weights definitions are presented in equations (4), (5) and (6) respectively.

Random inertia weight [18]

$$(w) = 0.5 + \frac{rand()}{2} \tag{4}$$

Chaotic inertia weight [25]

$$(w) = (w_1 - w_2) \times \frac{MAXiter - iter}{MAXiter} + w_2 \times z \tag{5}$$

where w_1 and w_2 are inertia constant, $MAXiter$ is the maximum number of iteration and z is logistic mapping.

Modified inertia weight [12]

$$w(t+1) = \exp\left(-\exp\left(-F(t)\right)\right)$$

(6) where $F(t) = \left(\frac{\max t - t}{\max t}\right)$

B. PSO Parameters and Simulation in Matlab Software

The parameters of the particle swarm optimization for the testing of the proposed weight definition are shown in table 2. All the four weight definitions for the PSO were run on the same computer. The specifications of the computer are as follows: Intel (R) Core TM i7-10750H with CPU of 2.60 GHz and 16.0GB RAM. The computer uses a windows-based operating system. The weight definitions for each test function were run five times. It should be noted that the random generator in MATLAB was set to default restarting each run with the same random numbers. The performance of the proposed weight was assessed in terms of best cost, average cost, the standard deviation of cost, and simulation execution time after five runs. The standard deviation is used to check how each iteration's cost spread around the optimum cost.

Table 2: PSO parameters

Parameter	Value
Population size	100
Inertia weight (w_1)	0.9
Inertia weight (w_2)	0.4
Acceleration factor (c_1)	2
Acceleration factor (c_2)	2
Maximum iteration	1000
Maximum run	5

IV. RESULTS AND ANALYSIS

The results of the experiment are presented under three heading as; best cost of optimum cost, mean cost and standard deviation of the cost obtained and the convergence rate. The results presented are recorded after five separate runs of each benchmark function. For the sake of comparison, abbreviation representing each of the weight definitions is used for discussion of results and the best values are also boldened in each table of results. These are personal best adaptive weight (PSO-PBAW), modified inertia weight (PSO-MIW), chaotic inertia weight (PSO-CIW) and random weight (PSO-RW).

A. Best Cost And Execution Time

The Best costs (optimum cost) obtained by the four weights definitions for the particle swarm optimization algorithm (PSO) for each benchmark function is shown in table 3. These costs are the best obtained after five separate runs. Since the benchmark costs for these functions are zeros (except that for the Michwalze and Dekker-Aarts function), the closer an obtained value is to zero, the better. The PSO-PBAW obtained the best cost for Rosenbrock, and Beale functions with a very low value of 7.61E-16 and 1.60E-28 respectively. PSO-PBAW also had the best cost with PSO-MIW for Dekkers-Aarts function with a value of -2477.00 which is very close to the optimum value of −24777.09. Again, PSO-PBAW was the second-best performing weight for Griewank and Rastrigin functions with the cost of 4.75E-15 and 2.84E-15 respectively. For the Michwalze function, PSO-PBAW had the third-best cost of -9.2053 whilst PSO-MIW had the best cost of -9.5564.

Also, table 3 compares the execution time for each weighted PSO for obtaining their respective best cost for each benchmark function. The lower the time the better the execution time. It is noted that PSO-PBAW had the best and least time for all test functions except Rosenbrock where it had the second least time of 2.31s. The time taken by PSO-PBAW to obtain the cost for Rastrigin, Dekkers-Aarts, Michwalze, Griewank, and Beale were 0.25s, 0.87s,1.64s, 1.17s, and 0.61s respectively. The personal best adaptive weight definition has improved the convergence cost of the particle swarm optimization algorithm to a large extent as has been shown in the test results of the benchmark functions in table 3. Significantly, it obtained this performance with the least time relative to the other weight definitions.

Table 3: Best cost comparison of benchmark functions

Function	Weight definition	Minimum Cost	Execution time (s)
Rosenbrock	MIW	3.57E-10	3.20
	CIW	1.97E-04	3.18
	RW	1.13E-01	**2.31**
	PBAW	**7.61E-16**	2.67
Rastrigin	MIW	3.44E-15	0.39
	CIW	3.55E-15	0.56
	RW	**2.84E-15**	0.75
	PBAW	8.69E-09	**0.25**
Dekkers-Aarts	MIW	**-24777.00**	1.85
	CIW	-24776.52	2.45
	RW	-24776.52	3.14
	PBAW	**-24777.00**	**0.87**
Michwalze	MIW	**-9.5564**	3.39
	CIW	-9.4265	3.12
	RW	-7.1226	4.17
	PBAW	-9.2053	**1.64**
Griewank	MIW	6.61E-13	1.35
	CIW	3.15E-11	1.96
	RW	**4.75E-15**	1.71
	PBAW	3.15E-10	**1.17**
Beale	MIW	9.87E-19	1.07
	CIW	8.27E-15	1.48
	RW	1.28E-11	1.36
	PBAW	**1.60E-28**	**0.61**

B. Mean Cost

Table 4 compares the mean cost obtain by each weighted PSO for the benchmark functions. The closer the mean cost is to the optimum value of the test function the better. It shows the consistency of the algorithm in obtaining the optimum cost. It is noted from table 4 that the mean cost obtained by PSO-PBAW for Rosenbrock, Dekkers-Aarts, Michwalze, Griewank, and Beale is lower than that obtained by PSO-MIW, PSO-WIW, and PSO-RW and second lowest for Rastrigin function. The mean costs obtained by PSO-PBAW were 8.37E-02, -24777.00, -8.0199, 7.90E-03 and 2.95E-13 respectively and 1.98E-02 for Rastrigin function. PSO-PBAW clearly outperformed the other three weight definitions in terms of the mean cost.

Table 4: Mean cost comparison of benchmark functions

Function	Weight definition	Mean Cost
Rosenbrock	MIW	9.10E-01
	CIW	3.06E+02
	RW	3.46E+02
	PBAW	**8.37E-02**
Rastrigin	MIW	2.40E-01
	CIW	**2.30E-03**
	RW	1.77E-01
	PBAW	1.98E-02
Deckkers-Aarts	MIW	-24776.52
	CIW	-22052.18
	RW	-22772.47
	PBAW	**-24777.00**
Michwalze	MIW	-7.5438
	CIW	-6.9790
	RW	-4.8368
	PBAW	**-8.0199**
Griewank	MIW	1.42E-02
	CIW	1.63E-02
	RW	8.60E-03
	PBAW	**7.90E-03**
Beale	MIW	2.56E-04
	CIW	7.50E-06
	RW	6.80E-04
	PBAW	**2.95E-13**

C. Standard Deviation

Table 5 compares the standard deviations (SD) of the cost obtained by each weighted PSO. The SD shows the spread within the cost and it is expected to be low to denote better performance of the PSO algorithm. The standard deviations were low for Dekkers-Aarts, Griewank, and Beale for all weights compared to the cost obtained in other functions. However, PSO-PBAW had the lowest SDs for Beale, Michwalze, Rosenbrock, and Dekkers-Aarts functions compared to PSO-MIW, PSO-WIW, and PSO-RW with values of 6.62E-13, 1.185, 5.72E-10, and 1.60E-01 respectively. Again, PSO-PBAW was the least performing PSO for the Rastrigin function where PSO-CIW had the lowest standard deviation of 4.10E-02. The lowest standard deviations obtained by the PSO-PBAW variant confirms the low values of the mean cost obtained by PSO-PBAW as shown in table 4.

Table 5: Standard deviation comparison of cost of benchmark functions

Function	Weight definition	Standard deviation
Rosenbrock	MIW	1.72E+00
	CIW	6.48E+02
	RW	6.08E+02
	PBAW	**1.60E-01**
Rastrigin	MIW	6.08E-01
	CIW	**4.10E-02**
	RW	5.44E-01
	PBAW	8.34E-01
Deckkers-Aarts	MIW	3.13E-09
	CIW	6894.36
	RW	4971.68

	PBAW	**5.72E-10**
Michwalze	MIW CIW RW PBAW	1.479 3.449 1.751 **1.185**
Griewank	MIW CIW RW PBAW	5.94E-02 6.10E-02 **2.94E-02** 4.45E-02
Beale	MIW CIW RW **PBAW**	5.73E-04 1.68E-05 1.50E-03 **6.62E-13**

D. Convergence Rate

Figures 1, 2,3,4,5, and 6 show convergence curves for the Beale, Michwalze, Rosenbrock, Griewank, and Rastrigin with the application of the PSO-PBAW, PSO-MIW, PSO-WIW, and PSO-RW algorithms. It is observed from all figures that the PSO-PBAW had the fastest convergence (i.e., least number of iterations), with the least cost in Griewank, Rosenbrock, and Beale function. The PSO-PBAW avoided premature convergence and stagnation in the search process for these functions. Particularly, it exhibited an extraordinary fastest convergence in the Michwalze function with the third-lowest convergence cost compared to PSO-MIW, PSO-WIW, and PSO-RW. Again PSO-PBAW exhibited the fastest convergence for Rastrigin and Griewank functions with the second-lowest convergence cost compared to PSO-MIW, PSO-WIW, and PSO-RW. The convergence rate and cost of the particle swarm optimization algorithm have been improved with the personal best adaptive weight definition.

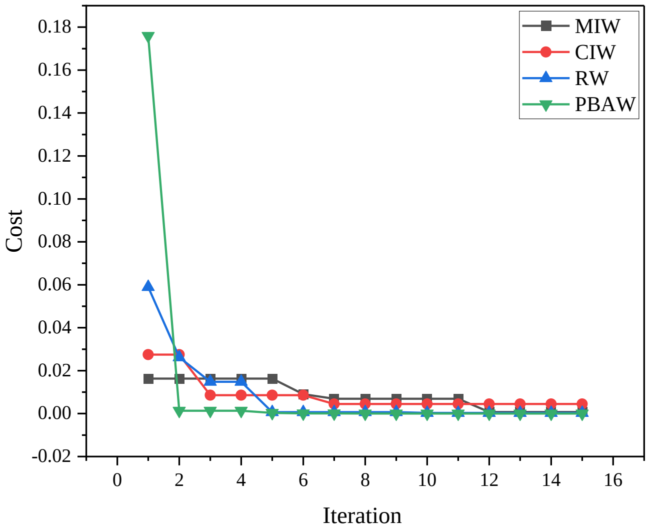

Figure 1: Convergence curve of Beale

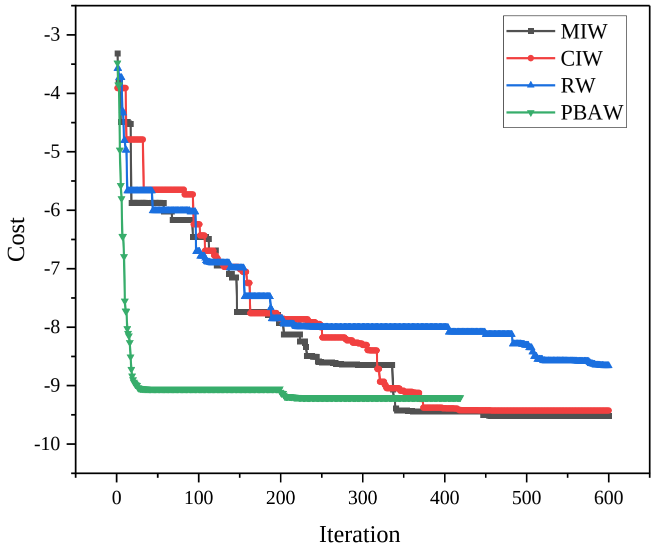

Figure 2: Convergence curve of Michwalze

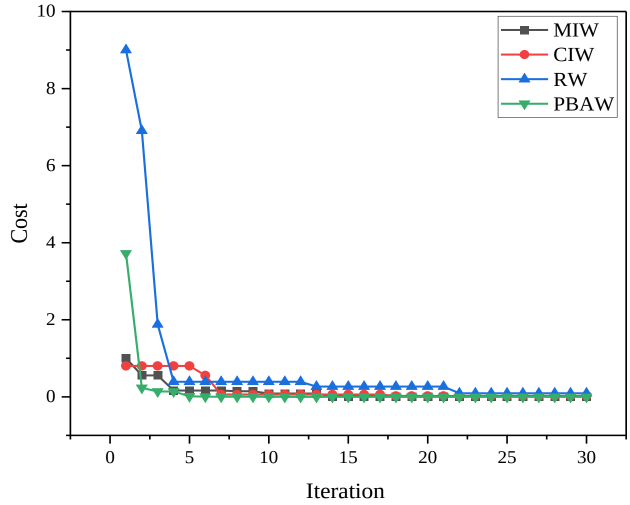

Figure 3: Convergence curve of Rosenbrock

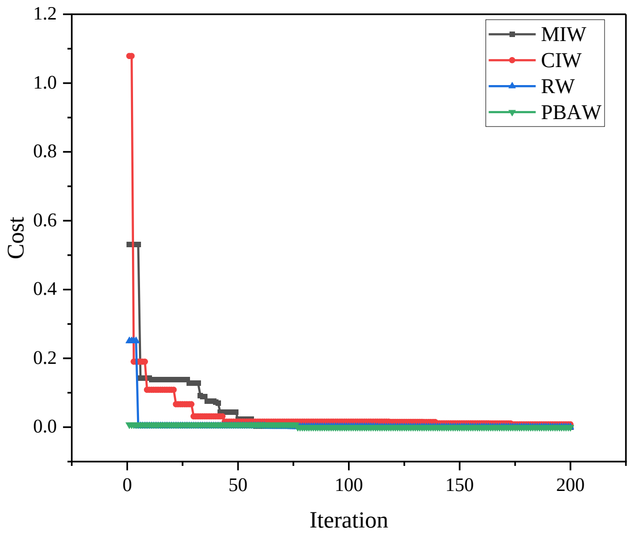

Figure 4: Convergence curve of Griewank

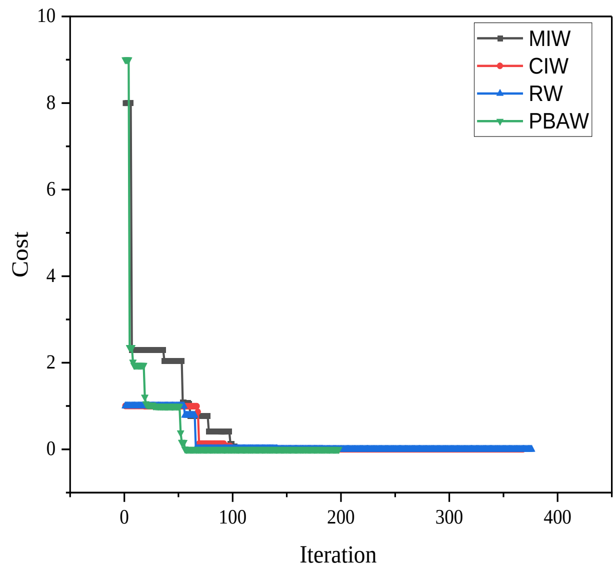

Figure 5: Convergence curve of Rastrigin

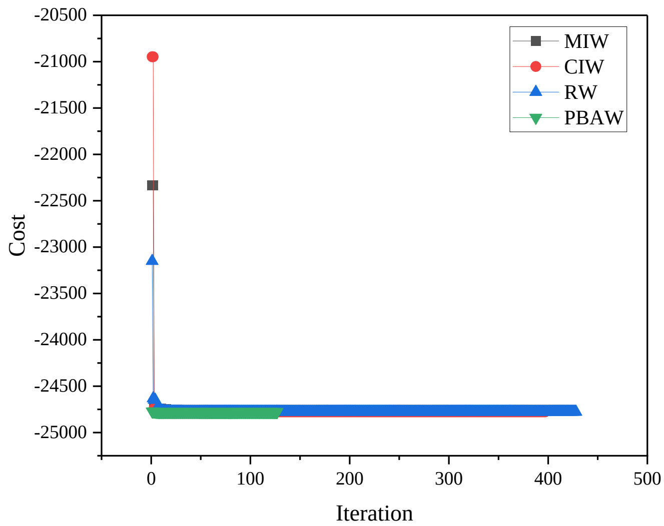

Figure 6: Convergence curve of Dekker and Aarts

V. CONCLUSION

A personal best adaptive weight is presented in this work as a new variant to improve the accuracy of the classical particle swarm optimization. The performance of the proposed improvement was tested using six benchmark functions. The proposed adaptive weight showed a high degree of accuracy when its performance was compared against three other PSO improved variants in literature. This improved algorithm approach can be adopted to solve any optimization problem

REFERENCES

[1]. Janga Reddy M, Nagesh Kumar D, "Evolutionary algorithms, swarm intelligence methods, and their applications in water resources engineering: a state-of-the-art review," H2Open Journal, 3(1):135-88, Jan 2021.

[2]. Jahandideh-Tehrani M, Bozorg-Haddad O, Loáiciga HA, "A review of applications of animal-inspired evolutionary algorithms in reservoir operation modelling," Water and Environment Journal, 35(2):628-46, 2021 May.

[3]. Deepika D., GS Tomar, "Interactive Biogeography Particle Swarm Optimization for Content Based Image Retrieval," Wireless Personal Communications, 2019, Vol.106, No.4, pp.1-15.

[4]. Jiang F, Zhang Y, Zhang Y, Liu X, Chen C, "An adaptive particle swarm optimization algorithm based on guiding strategy and its application in reactive power optimization," Energies, 12(9):1690, Jan 2019.

[5]. Kessentini S., Barchiesi D., T. Grosges, Moreau L. G., and Lamy de la Chapelle M., "Adaptive non-uniform particle swarm optimizetion: application to plasmonic design," International Journal of Applied Metaheuristic Computing," vol. 2, pp. 18-28, 2011.

[6]. Amoshahy MJ, Shamsi M, Sedaaghi MH, "A novel flexible inertia weight particle swarm optimization algorithm, PloS one," 11(8):e0161558, Aug 2016.

[7]. Ab Wahab MN, Nefti-Meziani S, Atyabi A, "A comprehensive review of swarm optimization algorithms," PloS one, 10(5):e0122827, May 2016.

[8]. Xu L, Song B, Cao M, "An improved particle swarm optimization algorithm with adaptive weighted delay velocity," Systems Science & Control Engineering, 9(1):188-97, Jan 2021.

[9]. M. Jakubcová, P. Máca, and P. Pech, "A comparison of selected modifications of the particle swarm optimization algorithm," Journal of Applied Mathematics, vol. 2014, Article ID 293087, 10 pages, 2014.

[10]. Tripathi DP, Jena UR, "Cognitive and social information based PSO," International Journal of Engineering, Science and Technology, 8(3):64-75, 2016.

[11]. Abdullah MN, Bakar AH, Rahim NA, Mokhlis H, Illias HA, Jamian JJ, "Modified particle swarm optimization with time varying acceleration coefficients for economic load dispatch with generator constraints," Journal of Electrical Engineering and Technology, 9(1):15-26, 2014.

[12]. Sabir M, Ahmad A, Ahmed A, Siddique S, Hashmi UA, "A Modified Inertia Weight Control of Particle Swarm Optimization for Optimal Reactive Power Dispatch Problem," (pp.1-6), April 2021 [2021 International Conference on Emerging Power Technologies (ICEPT)].

[13]. Meziane MA. Y. mouloudi, B. Bouchiba, and A. Laoufi,"Impact of inertia weight strategies in particle swarm optimization for solving economic dispatch problem," Indonesian Journal of Electrical Engineering and Computer Science, 13(1):377-83, 2019.

[14]. Wang S, Zhou F, Wang F, "Effect of Inertia Weight ω on PSO-SA Algorithm, "International Journal of Online Engineering," 1;9, July 2013.

[15]. Shi, Y. and Eberhart, R., “A modified particle swarm optimizer,” In 1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence, 69-73 1998, May.
[16]. Hu Z, Zou D, Kong Z, Shen X, “A particle swarm optimization algorithm with time varying parameters,” IEEE, (pp. 4555-4561), June 2018 [2018 Chinese Control and Decision Conference (CCDC)].
[17]. J. C. Bansal, P. K. Singh, M. Saraswat, A. Verma, S. S. Jadon and A. Abraham, “Inertia Weight strategies in Particle Swarm Optimization,” pages. 633- 640, 2011 [Third World Congress on Nature and Biologically Inspired Computing, Salamanca].
[18]. Eberhart, R. C., & Shi, Y. “Tracking and optimizing dynamic systems with particle swarms,” In Proceedings of the 2001 congress on evolutionary computation. Vol. 1, pp. 94-100, May 2001.
[19]. Maca P, Pech P, “The inertia weight updating strategies in particle swarm optimisation based on the beta distribution,” Mathematical Problems in Engineering. 17;2015, March 2015.
[20]. J. C. Bansal, P. K. Singh, M. Saraswat, A. Verma, S. S. Jadon, and A. Abraham, “Inertia weight strategies in particle swarm optimization,” pp. 633–640, October 2011 [3rd World Congress on Nature and Biologically Inspired Computing (NaBIC '11)].
[21]. Kessentini S, Barchiesi D, “Particle swarm optimization with adaptive inertia weight,” International Journal of Machine Learning and Computing, 5(5):368, October 2015.
[22]. Li M, Chen H, Wang X, Zhong N, Lu S,” An improved particle swarm optimization algorithm with adaptive inertia weights,” International Journal of Information Technology & Decision Making, 18(03):833-66, May 2019.
[23]. Kennedy J, Eberhart R, “Particle swarm optimization,” IEEE, (Vol. 4, pp. 1942-1948), Nov 1995 [ICNN'95- international conference on neural networks].
[24]. Bansal, J. C., Sharma, H., Jadon, S. S., and Clerc, M. “Spider monkey optimization algorithm for numerical optimization”. Memetic computing, 6(1), 31-47, 2014.
[25]. Feng, Y., Teng, G. F., Wang, A. X., and Yao, Y. M. “Chaotic inertia weight in particle swarm optimization,” In Second International Conference on Innovative Computing, Informatio and Control , 475-475, September 2007.