

Dynamic Inertia Weight in Particle Swarm Optimization

Bożena Borowska (✉)

Institute of Information Technology, Lodz University of Technology,
Wółczańska 215, 90-924 Łódź, Poland
bozena.borowska@p.lodz.pl

Abstract. This paper proposes a particle swarm optimization method with a novel strategy for inertia weight. Instead of a commonly used linear inertia weight, a nonlinear, dynamic changing inertia weight is applied. The new presented weight is a function of the worst and the best fitness of individuals of a population. In order to investigate the effectiveness of the proposed strategy tests on a set of benchmark function were conducted. The results were compared with those obtained through the LDW-PSO method, EWPSO and the RNW-PSO methods.

Keywords: Inertia weight · Particle swarm optimization
Swarm intelligence · Optimization

1 Introduction

Particle Swarm Optimization (PSO) is an optimization method modeled on the social behavior of the group of organisms in their natural environment [1]. Proposed for the first time by Kennedy and Eberhart in 1995 [2, 3], now belongs to the most frequently used evolutionary methods. Because of its many advantages such as simplicity, few parameters to adjust and easy implementation, it has been applied in almost all fields of science and engineering, where optimization is required [4–9]. A parameter that significantly influences the effectiveness of the particle swarm optimization method is inertia weight. Its role is to control deviation of the particles from their original direction and keep balance between local and global explorations. An incorrectly selected value of inertia weight maintains a balance between local and global exploration and can negatively affect the algorithm performance.

A factor of the inertia weight was first proposed and introduced to the PSO method by Shi and Eberhart [10, 11]. In the subsequent years, a lot of research on inertia weight have been undertaken. Clerc [12] and Trelea [13] suggested that the inertia weight should rather be a constant value. She and Eberhart [11, 14] recommended a linear decreasing inertia weight (LDW). A flexible inertia weight, that can be a positive or negative real number, was used by Han et al. [15]. In order to avoid some troubles of the LDW strategy connected with a poor local search ability at the beginning of the method, and the lack of global search ability at the end of the method, Zhang et al. [16] proposed a random inertia weight (RNW). PSO with random inertia weight was also proposed by Niu et al. [17] and Eberhart and She [18]. Three different concepts of

inertia weight were investigated by Arumugan and Rao [19, 20] and Umapathy et al. [21]. They considered a constant inertia weight (CIW), time-varying inertia weight (TVIW) and a global-local best inertia weight (GLbestIW) and reported that GLbestIW outperforms the other methods in terms of high quality solution, consistency, faster convergence and accuracy. Another approach was developed by Yang et al. [22]. In their study, the inertia weight depends on two parameters: an aggregation degree factor and an evolution speed factor, and is different for each individual of the swarm. A modified inertia weight was also considered by Miao et al. [23]. In this case, weight is updated by dispersion degree and advance degree factors. Performance of PSO with nonlinear strategies was examined by Chauhan et al. [24]. The authors examined an exponential self adaptive (DESIWPSO) and a dynamic (DEDIVPSO) inertia weight based on Gompertz function and a fine grained inertia weight (FGIWPSO). Another exponential inertia weight was also developed by Ememipour et al. [25], Borowska [26] and Ghali et al. [27]. A fuzzy adaptive inertia weight was proposed by Shi and Eberhart [28]. An approach based on the fuzzy systems was also described in [29–33].

This paper presents a modified PSO method named DWPSO in which a new strategy for inertia weight was developed. In DWPSO, the values of inertia weight are dynamically changing and are determined based on fitness function. The proposed weight is a function of the best and the worst fitness of particles. Moreover, the new presented method was tested on a set of benchmark function. Then the results were compared to the RNW-PSO method with a random inertia weight [16], the LDW-PSO method with a linear decreasing inertia weight, and the nonlinear EWPSO method [26].

2 The PSO Method

The PSO algorithm belongs to the group of optimization methods based on the population. In case of PSO this population is called swarm and consists of individuals named particles. Each particle is a point of the space of the feasible solutions. The movement of the particle in this space enables the velocity vector. Initial location and velocity of the particle are randomly generated at the beginning of the algorithm. The quality of the particles is evaluated according to the fitness function of the optimization problem. In each iteration, particles update information about their own best position (named $pbest$) found so far. The knowledge about the particle with the best fitness among all the particles in the whole swarm (named $gbest$) is also remembered and updated in every iteration. The change of the velocity and location of the particles is carried out according to the formula:

$$V_i = wV_i + c_1r_1(pbest_i - X_i) + c_2r_2(gbest - X_i) \quad (1)$$

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

where $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ is a velocity vector of the particle i in the D -dimensional search space. Vector $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ represents a location of the particle i . Factor w is the inertia weight. Vector $pbest_i = (pbest_{i1}, pbest_{i2}, \dots, pbest_{iD})$ means a personal best location of the particle i and $gbest = (gbest_1, gbest_2, \dots, gbest_D)$ denotes

a location of the particle with the best fitness function among all the particles in the whole swarm. The variables c_1 and c_2 are acceleration coefficients. They decide how strong the particle is influenced by its knowledge about its pbest and gbest value. Parameters r_1 and r_2 represent randomly generated numbers between 0 and 1 to maintain diversity of the population.

3 The Proposed DWPSO Algorithm

The proposed DWPSO algorithm is a variant of the particle swarm optimization method in which the new strategy for determination of inertia weight was introduced. In DWPSO a commonly used linear weight was omitted and replaced with an exponential inertia weight. In the proposed approach, the inertia weight is changing dynamically based on a fitness of the particles in the swarm. In each iteration, the particles of the swarm move in the search space according to Eqs. 1–2. After evaluating the quality of the new location of the particles, the individuals with the best and the worst fitness are found and recorded. Then, on their basis, the new value of the weight is counted. The new weight is represented by a nonlinear function of the best and the worst fitness of the particles. In each iteration, a different inertia weight is calculated and applied for the whole swarm. The proposed strategy has been defined as follows:

$$dw = ((gbest - f_{max}/2)^{-1}/((- \ln[f_{min}]))) / 10^5 fh \tag{3}$$

$$w(t + 1) = w(t) - dw(t) \tag{4}$$

where f_{max} and f_{min} are the values of maximal and minimal fitness in the current iteration, respectively. Factor fh is a randomly generated number in the range [0, 1].

4 Results

The simulation tests of the DWPSO method with proposed strategy was carried out on a set of nonlinear benchmark functions depicted in Table 1.

Table 1. Optimization test functions

Function	Formula	Minimum	Range of x
Sphere	$f_1 = \sum_{i=1}^n x_i^2$	0	(-100, 100)
Rosenbrock	$f_3 = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	0	(-30, 30)
Ackley	$f_4 = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	0	(-32, 32)
Rastrigin	$f_5 = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$	0	(-5.12, 5.12)
Griewank	$f_6 = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	0	(-600, 600)

The results of the tests were compared with the performance of PSO with a random number inertia weight (RNWPSO), LDW-PSO with a linear decreasing weight as well as EWPSO with an exponential inertia weight.

The DWPSO algorithm started with the inertia weight of 0.7 and was changing according to the formula 3–4. The acceleration coefficients c_1, c_2 used for executed computation were set to 1.6. The simulations were performed with four dimension sizes $D = 10, 20$ and 30 for $N = 20, 40, 60$ and 80 particles in the swarm respectively. Each experiment was run 50 times. In all cases, the iteration number was 1000.

The exemplary results of the tests performed for 20, 40 and 80 particles of the swarm are illustrated in Tables 2, 3 and 4. The presented values were averaged over 50 trials.

Table 2. Performance of the LDW-PSO, RNW-PSO, EWPSO and DWPSO algorithms for Rosenbrock function

Population size	Dimension	Algorithm	LDW-PSO	RNW-PSO	EWPSO	DWPSO
20	10	Mean	4.0933e+001	1.8401e+001	1.5489e+001	4.9787e+000
		St. Dev.	2.8127e+001	2.3554e+001	2.0147e+001	6.0743e+000
		Min	4.3607e+000	8.0913e-001	2.2132e-001	1.5060e-001
	20	Mean	8.3192e+001	1.9984e+002	7.2504e+001	5.0429e+001
		St. Dev.	4.5308e+001	2.8759e+002	1.3315e+002	4.1276e+001
		Min	1.5216e+001	1.5318e+001	9.0325e-001	2.1551e+000
	30	Mean	1.3507e+002	4.2856e+002	1.7210e+002	7.2359e+001
		St. Dev.	1.5563e+002	5.3721e+002	1.5334e+002	5.5085e+001
		Min	2.3142e+001	2.9608e+001	2.3512e+001	9.3401e+000
40	10	Mean	2.5379e+001	1.7240e+001	1.4231e+001	4.1631e+000
		St. Dev.	1.7455e+001	3.2178e+001	5.3622e+000	9.1435e+000
		Min	2.8927e-002	1.0806e+000	1.4523e-002	4.8498e-002
	20	Mean	5.7344e+001	5.6102e+001	4.7376e+001	4.0605e+001
		St. Dev.	5.3637e+001	4.9630e+001	4.8744e+001	6.4665e+001
		Min	6.1431+000	5.7073e+000	1.9198e+000	1.5801e-001
	30	Mean	7.1917e+001	6.9343e+001	6.7008e+001	5.2569e+001
		St. Dev.	1.4425e+002	3.5032e+001	7.5692e+001	2.9028e+001
		Min	1.1633e+001	1.0225e+001	6.8376e+000	2.5110e+000
80	10	Mean	1.8261e+001	1.5728e+001	1.2415e+001	5.5479e+000
		St. Dev.	2.9027e+001	2.3569e+001	2.2178e+001	1.8712e+001
		Min	1.2415e-001	2.8725e-003	9.4684e-003	2.2647e-002
	20	Mean	4.4502e+001	3.3485e+001	2.2570e+001	2.0819e+001
		St. Dev.	3.2417e+001	2.7279e+001	2.9342e+001	3.0246e+001
		Min	1.5655e-001	5.2635e-001	1.5904e-005	1.5185e+000
	30	Mean	6.6138e+001	4.2504e+001	3.9453e+001	3.8589e+001
		St. Dev.	1.3050e+002	3.2997e+001	2.6149e+001	3.4396e+001
		Min	9.5624e-001	1.1682e+001	1.2736e+001	1.3776e-002

Table 3. Performance of the LDW-PSO, RNW-PSO, EWPSO and DWPSO algorithms for Rastrigin function

Population size	Dimension	Algorithm	LDW-PSO	RNW-PSO	EWPSO	DWPSO
20	10	Mean	6.4812e+000	6.3174e+000	6.1075e+000	7.7109e+000
		St. Dev.	4.5160e+000	7.2541e+000	4.1338e+000	4.8600e+000
		Min	3.3097e+000	3.0705e+000	9.9775e-001	9.9496e-001
	20	Mean	4.5202e+001	5.3877e+001	4.3706e+001	3.7609e+001
		St. Dev.	1.6925e+001	1.6786e+001	1.4537e+001	1.2697e+001
		Min	2.2130e+001	3.1973e+001	2.5166e+001	1.9899e+001
	30	Mean	8.1753e+001	8.8445e+001	8.3524e+001	7.1985e+001
		St. Dev.	2.2488e+001	2.9007e+001	2.4530e+001	2.1340e+001
		Min	4.5357e+001	6.2319e+001	3.8712e+001	4.5768e+001
40	10	Mean	4.3519e+000	4.3854e+000	4.1303e+000	4.7657e+000
		St. Dev.	3.9689e+000	2.2077e+000	3.6194e+000	3.1695e+000
		Min	1.0526e+000	2.8343e+000	1.2376e+000	1.9899e+000
	20	Mean	3.6697e+001	3.4463e+001	3.1793e+001	2.3779e+001
		St. Dev.	1.1455e+001	9.0821e+000	1.3508e+001	7.9528e+000
		Min	1.1243e+001	1.5354e+001	1.9487e+001	2.9849e+000
	30	Mean	7.3137e+001	7.2862e+001	6.7972e+001	6.3329e+001
		St. Dev.	3.3256e+001	2.0617e+001	2.5146e+001	1.8331e+001
		Min	3.6972e+001	4.1565e+001	3.3887e+001	2.7859e+001
80	10	Mean	2.7314e+000	2.5931e+000	2.4145e+000	2.5801e+000
		St. Dev.	4.3898e+000	2.8769e+000	1.8860e+000	2.2603e+000
		Min	9.7515e-001	9.9004e-001	9.7073e-001	0.0000e+000
	20	Mean	3.1305e+001	3.0798e+001	2.8712e+001	2.1292e+001
		St. Dev.	9.4789e+000	9.7543e+000	1.1676e+001	8.4036e+000
		Min	1.6002e+001	1.6200e+001	1.0841e+001	8.9546e+000
	30	Mean	5.7351e+001	5.5917e+001	5.5712e+001	5.3379e+001
		St. Dev.	1.6879e+001	1.7884e+001	1.6234e+001	1.2905e+001
		Min	4.3125e+001	3.3266e+001	3.6258e+001	2.9849e+001

Table 4. Performance of the LDW-PSO, RNW-PSO, EWPSO and DWPSO algorithms for Griewank function

Population size	Dimension	Algorithm	LDW-PSO	RNW-PSO	EWPSO	DWPSO
20	10	Mean value	9.3627e-002	1.3901e-001	8.1573e-002	8.2364e-002
		St. Dev.	7.9415e-002	4.8233e-002	3.5967e-002	5.5734e-002
		Min	2.0163e-002	3.0994e-002	1.3029e-002	9.8573e-003
	20	Mean value	5.8212e-002	5.6745e-002	4.3195e-002	3.1448e-002
		St. Dev.	6.4934e-002	4.2267e-002	2.7830e-002	2.9442e-002
		Min	2.0407e-005	00000e+000	00000e+000	0.0000e+000
	30	Mean value	1.9235e-001	4.1733e-000	1.7908e-001	6.6079e-002
		St. Dev.	2.7506e-001	3.7786e-000	2.6839e-001	1.2421e-001
		Min	2.9618e-009	1.3545e-001	1.1424e-005	3.4376e-010

(continued)

Table 4. (continued)

40	10	Mean value	7.6583e-002	7.2025e-002	6.8570e-002	6.9333e-002
		St. Dev.	5.9771e-002	4.3162e-002	1.6405e-002	3.7984e-002
		Min	6.7088e-003	5.9476e-002	1.8311e-002	2.7037e-002
	20	Mean value	5.4734e-002	4.3979e-002	4.0720e-002	2.0134e-002
		St. Dev.	5.1682e-002	3.1848e-002	2.8183e-002	1.7913e-002
		Min	8.6707e-004	00000e+000	0.0000e+000	6.5454e-003
	30	Mean value	3.4395e-002	3.2599e-002	2.4759e-002	1.8896e-002
		St. Dev.	4.1504e-002	3.8676e-002	3.5383e-002	1.5956e-002
		Min	6.2816e-003	7.9302e-003	3.3418e-013	2.1289e-005
80	10	Mean value	7.2003e-002	6.5499e-002	5.8677e-002	6.2356e-002
		St. Dev.	4.1688e-002	5.4212e-002	3.2431e-002	2.2412e-002
		Min	5.4053e-003	3.3827e-002	8.5972e-004	2.4603e-002
	20	Mean value	2.8475e-002	2.5688e-002	2.0084e-002	1.0083e-002
		St. Dev.	1.7142e-002	2.0213e-002	1.5755e-002	1.5652e-002
		Min	0.0000e+000	00000e+000	0.0000e+000	0.0000e+000
	30	Mean value	2.1659e-002	1.7706e-002	9.5268e-003	8.8063e-003
		St. Dev.	1.8374e-002	1.8195e-002	1.5169e-002	1.3302e-002
		Min	0.0000e+000	00000e+000	0.0000e+000	0.0000e+000

The average best fitness in the following iterations for both DWPSO, EWPSO, RNW-PSO algorithms and LDW-PSO model for 40 particles (swarm size) and 30 dimensions is illustrated in Figs. 1, 2, 3 and 4. The vertical coordinates indicate the average best fitness in a logarithmic scale.

The results of simulations show that the proposed DWPSO method is more effective than the other methods investigated in this study. The dynamic strategy for inertia weight introduced for DWPSO facilitates the algorithm to maintain a diversity of the individuals in the search space and helps overcome the problem premature convergence.

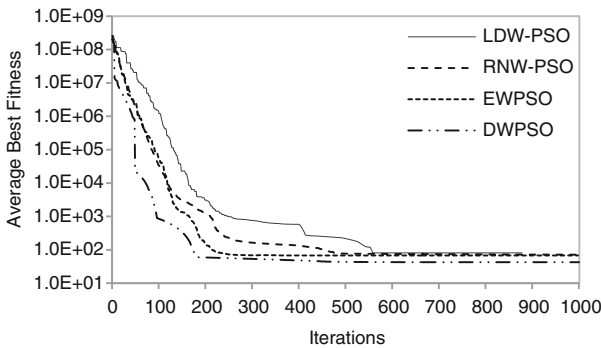


Fig. 1. The average best fitness for Rosenbrock30 and the population of 40 particles

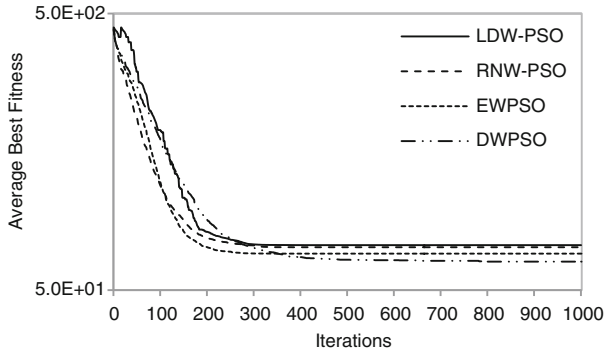


Fig. 2. The average best fitness for Rastrigin30 and the population of 40 particles

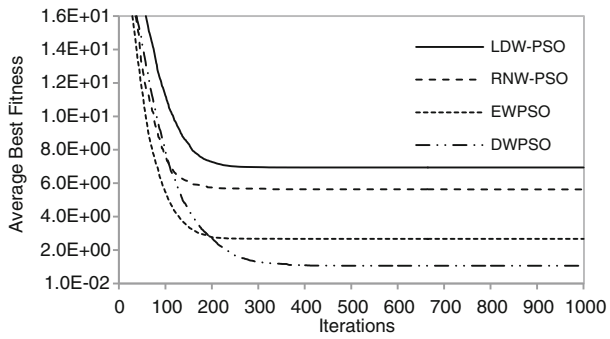


Fig. 3. The average best fitness for Ackley30 and the population of 40 particles

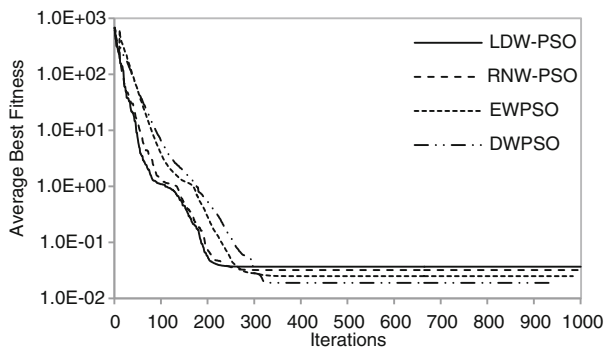


Fig. 4. The average best fitness for Griewank30 and the population of 40 particles

In almost all cases (except $D = 10$), the average function values found by DWPSO were lower than the results achieved by LDW-PSO and RNW-PSO, and lower or in rare cases comparable to those obtained by EWPSO. Moreover, the lowest standard deviation for DWPSO reported in most cases indicates its better stability compared to remaining investigated method. Furthermore, the minimum value was also lower in case of the DWPSO algorithm.. Additionally, in most simulations, DWPSO converged faster than LDW-PSO, RNW-PSO and EWPSO (Figs. 1, 2, 3 and 4) and only at the beginning, in the first two hundred iterations, the algorithm converged a bit slower than EWPSO or RNW-PSO (After first two hundred iterations DWPSO was the fastest). The RNW-PSO algorithm converged slower than EWPSO but still better than LDW-PSO.

Different performance of the algorithms have been noticed only for Griewank and Rastrigin functions and small dimensions. In case of Griewank function with $D = 10$ dimension size, the DWPSO algorithm performed a bit worse than EWPSO but better than RNW-PSO and LDW-PSO. For Rastrigin function with $D = 10$ for 20 and 40 particles, DWPSO achieve worse results than the remaining algorithms even when the minimum value was lower.

5 Summary

In this study, a modified particle swarm optimization algorithm named DWPSO with a novel strategy for inertia weight has been proposed. In the considered approach, instead of a commonly used constant or linear decreasing inertia weight, a dynamically changing weight was adopted. Values of the inertia weight coefficient depend on fitness of the individuals of the population. The effectiveness of the proposed strategy was tested on a set of benchmark test functions. The results of the simulations were compared with those obtained through the nonlinear EWPSO method, the RNW-PSO method with a random inertia weight and the LDW-PSO method with a linear decreasing inertia weight.

The use of the proposed strategy helps maintain diversity of individuals of the population and performs better than the other investigated methods. Furthermore, the DWPSO algorithm is also faster and more efficient in avoiding the premature convergence, compared to the other methods.

References

1. Robinson, J., Rahmat-Samii, Y.: Particle swarm optimization in electromagnetics. *IEEE Trans. Antennas Propag.* **52**, 397–407 (2004)
2. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: *IEEE International Conference on Neural Networks*, Perth, Australia, pp. 1942–1948 (1995)
3. Kennedy, J., Eberhart, R.C., Shi, Y.: *Swarm Intelligence*. Morgan Kaufmann Publishers, San Francisco (2001)
4. Guedria, N.B.: Improved accelerated PSO algorithm for mechanical engineering optimization problems. *Appl. Soft Comput.* **40**, 455–467 (2016)

5. Dolatshahi-Zand, A., Khalili-Damghani, K.: Design of SCADA water resource management control center by a bi-objective redundancy allocation problem and particle swarm optimization. *Reliab. Eng. Syst. Saf.* **133**, 11–21 (2015)
6. Mazhoud, I., Hadj-Hamou, K., Bigeon, J., Joyeux, P.: Particle swarm optimization for solving engineering problems: a new constraint-handling mechanism. *Eng. Appl. Artif. Intell.* **26**, 1263–1273 (2013)
7. Yildiz, A.R., Solanki, K.N.: Multi-objective optimization of vehicle crashworthiness using a new particle swarm based approach. *Int. J. Adv. Manuf. Technol.* **59**, 367–376 (2012)
8. Hajforoosh, S., Masoum, M.A.S., Islam, S.M.: Real-time charging coordination of plug-in electric vehicles based on hybrid fuzzy discrete particle swarm optimization. *Electr. Power Syst. Res.* **128**, 19–29 (2015)
9. Yadav, R.D.S., Gupta, H.P.: Optimization studies of fuel loading pattern for a typical pressurized water reactor (PWR) using particle swarm method. *Ann. Nucl. Energy.* **38**, 2086–2095 (2011)
10. Shi, Y., Eberhart, R.: A modified particle swarm optimizer. In: *IEEE International Conference on Evolutionary Computation*, pp. 69–73 (1998)
11. Shi, Y., Eberhart, R.C.: Parameter selection in particle swarm optimization. In: *Proceedings of the Seventh Annual Conference on Evolutionary Programming*, New York, pp. 591–600 (1998)
12. Clerc, M.: The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In: *Proceedings of ICEC*, Washington, D.C., pp. 1951–1957 (1999)
13. Trelea, I.C.: The particle swarm optimization algorithm: convergence analysis and parameter selection. *Inf. Process. Lett.* **85**, 317–325 (2003)
14. Shi, Y., Eberhart, R.C.: Empirical study of particle swarm optimization. In: *Proceedings of the Congress on Evolutionary Computation*, vol. 3, pp. 1945–1950 (1999)
15. Han, Y., Tang, J., Kaku, I., Mu, L.: Solving uncapacitated multilevel lot-sizing problems using a particle swarm optimization with flexible inertial weight. *Comput. Math Appl.* **57**, 1748–1755 (2009)
16. Zhang, L., Yu, H., Hu, S.: A new approach to improve particle swarm optimization. In: *Proceedings of the International Conference on Genetic and Evolutionary Computation*, pp. 134–139. Springer, Berlin (2003)
17. Niu, B., Zhu, Y.L., He, X., Wu, H.: A multi-swarm cooperative particle swarm optimizer. *Appl. Math. Comput.* **185**, 1050–1062 (2007)
18. Eberhart, R.C., Shi, Y.: Tracking and optimizing dynamic systems with particle swarms. In: *IEEE Congress on Evolutionary Computation*, Seoul, Korea, pp. 94–97 (2001)
19. Arumugan, M.S., Rao, M.V.C.: On the performance of the particle swarm optimization algorithm with various inertia weight variants for computing optimal control of a class of hybrid systems. *Discrete Dyn. Nat. Soc.* **2006**, 1–17 (2006)
20. Arumugan, M.S., Rao, M.V.C., Chandramohan, A.: A new and improved version of particle swarm optimization algorithm with global-local best parameters. *Knowl. Inf. Syst.* **16**, 331–357 (2008)
21. Umaphathy, P., Venkateshaiah, C., Arumugam, M.S.: Particle swarm optimization with various inertia weight variants for optimal power flow solution. *Discrete Dyn. Nat. Soc.* **2010**, 1–15 (2010)
22. Yang, X., Yuan, J., Yuan, J., Mao, H.: A modified particle swarm optimizer with dynamic adaptation. *Appl. Math. Comput.* **189**, 1205–1213 (2007)
23. Miao, A., Shi, X., Zhang, J., Wang, E., Peng, S.: A modified Particle Swarm Optimizer with Dynamical Inertia Weight, pp. 767–776. Springer, Berlin (2009)
24. Chauhan, P., Deep, K., Pant, M.: Novel inertia weight strategies for particle swarm optimization. *Memetic Comput.* **5**, 229–251 (2013)

25. Ememipour, J., Nejad, M.M.S., Rezanejad, M.M.J.: Introduce a new inertia weight for particle swarm optimization. In: Proceedings of Fourth International Conference on Computer Sciences and Convergence Information Technology, pp. 1650–1653 (2009)
26. Borowska, B.: Exponential inertia weight in particle swarm optimization. In: Advances in Intelligent Systems and Computing, vol. 524, pp. 265–275. Springer International Publishing, Heidelberg (2017)
27. Ghali, I., El-Dessouki, N., Mervat, A.N., Bakrawi, L.: Exponential particle swarm optimization approach for improving data clustering. *Int. J. Electr. Electron. Eng.* **3–4**, 208–212 (2009)
28. Shi, Y., Eberhart, R.C.: Fuzzy adaptive particle swarm optimization. In: Proceedings of the Congress on Evolutionary Computation, vol. 1, pp. 101–106 (2001)
29. Tian, D., Li, N.: Fuzzy particle swarm optimization algorithm. In: International Joint Conference on Artificial Intelligence, pp. 263–267 (2009)
30. Neshat, M.: FAIPSO: Fuzzy Adaptive Informed Particle Swarm Optimization. *Neural Comput. Appl.* **23**, 95–116 (2013)
31. Chen, T., Shen, Q., Su, P., Shang, C.: Fuzzy rule weight modification with particle swarm optimization. In: *Soft Computing*, pp. 1–15 (2015)
32. Mohiuddin, M.A., Khan, S.A., Engelbrecht, A.P.: Fuzzy particle swarm optimization algorithms for the open shortest path first weight setting problem. *Appl. Intell.* 1–24 (2016)
33. Chaturvedi, D.K., Kumar, S.: Solution to electric power dispatch problem using fuzzy particle swarm optimization algorithm. *J. Inst. Eng. India* **96**, 101–106 (2015)