

Research Article

Global Particle Swarm Optimization for High Dimension Numerical Functions Analysis

J. J. Jamian,¹ M. N. Abdullah,^{2,3} H. Mokhlis,² M. W. Mustafa,¹ and A. H. A. Bakar²

¹ Faculty of Electrical Engineering, Universiti Teknologi Malaysia (UTM), 81310 Johor Bahru, Johor, Malaysia

² Faculty of Engineering, University of Malaya, 50603 Kuala Lumpur, Malaysia

³ Faculty of Electrical and Electronics Engineering, Universiti Tun Hussein Onn Malaysia, 86400 Batu Pahat, Johor, Malaysia

Correspondence should be addressed to J. J. Jamian; jasrul@fke.utm.my

Received 21 October 2013; Revised 18 December 2013; Accepted 10 January 2014; Published 25 February 2014

Academic Editor: Marcelo A. Savi

Copyright © 2014 J. J. Jamian et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The Particle Swarm Optimization (PSO) Algorithm is a popular optimization method that is widely used in various applications, due to its simplicity and capability in obtaining optimal results. However, ordinary PSOs may be trapped in the local optimal point, especially in high dimensional problems. To overcome this problem, an efficient Global Particle Swarm Optimization (GPSO) algorithm is proposed in this paper, based on a new updated strategy of the particle position. This is done through sharing information of particle position between the dimensions (variables) at any iteration. The strategy can enhance the exploration capability of the GPSO algorithm to determine the optimum global solution and avoid traps at the local optimum. The proposed GPSO algorithm is validated on a 12-benchmark mathematical function and compared with three different types of PSO techniques. The performance of this algorithm is measured based on the solutions' quality, convergence characteristics, and their robustness after 50 trials. The simulation results showed that the new updated strategy in GPSO assists in realizing a better optimum solution with the smallest standard deviation value compared to other techniques. It can be concluded that the proposed GPSO method is a superior technique for solving high dimensional numerical function optimization problems.

1. Introduction

Nowadays, there are various types of modern optimization techniques, such as Evolutionary Programming (EP), Artificial Immune System (AIS), Ant Colony Optimization (ACO), and Artificial Bees Colony (ABC). All of them can be regarded as heuristic optimization methods, due to the randomization involved in their respective initial steps. Despite the usage of randomized values, the mutation process and other steps in the algorithm render the optimization method capable of solving both linear and nonlinear problems. From the literature, it can be seen that optimization techniques are applicable in many fields. However, among the optimization methods, PSO has become very popular, due to its simplicity and its affinity towards manufacturing and robotics [1–3], electrical power systems [4–8], and engineering [9–11] and in other areas [12–18].

There is no specific optimization algorithm that can reach a global solution for every optimization problem.

Some algorithms can only provide the best solution to a particular problem, but not to others [19]. In classical PSO cases, PSO sometimes fails to find a global optimum solution when dealing with high dimensional problems. This problem occurs due to the fact that the particles are trapped at a local optimum (premature) solution.

In order to address such problems, several variants of PSO have been proposed to enhance the exploration and exploitation capability and convergence speed and to avoid being trapped at the local optimum [20–26]. The modification is made on the PSO algorithm itself or by hybridizing the PSO with other optimization techniques. A new adaptive weight technique that was proposed by Nickabadi et al. [27] was among the recent works involving the modification of the PSO algorithm. There are 3 main constant parameters that will affect the performance of the PSO, which are inertia weight (w), cognitive constant (c_1), and social constant (c_2); the changes of these parameters will result in different performances of the algorithm. In a traditional PSO, these

parameters are unspecified and need to be adjusted several times in order to obtain a suitable value [28, 29]. This implies that there are possibilities for the user to obtain inaccurate settings on the initial values (c_1 , c_2 , and w), which will result in the PSO not converging. Therefore, the tool on time varying inertia weight technique in PSO has been introduced in [30–34] and also the adaptive inertia weight methods in [35–38]. In [27], the authors have used a new adaptive inertia weight approach, which uses the success rate of the swarm as a feedback parameter to determine the particles' situation in the search space to enhance the performance of PSO. The results showed that the proposed method produced superior results compared to other weight adjustment techniques. Although the results in the study were better, it requires a large number of iterations ($\times 10^5$) before achieving the global optimal value compared to other techniques. Thus, for larger dimensions, the adjustment on inertia weight parameter alone might not be sufficient for the algorithm to realize the global optimal value.

In this paper, a novel Global Particle Swarm Optimization (GPSO) is proposed to improve and enhance the convergence speed of classical PSO. The working principle of the proposed algorithm is based on a new proposed updated parameter (E_{best}) in order to enrich the capability of the algorithm in finding a global solution. This new updating strategy can explore more possible solutions in searching space. The performance of GPSO is compared with three other optimization methods, which are traditional PSO, Iteration Particle Swarm Optimization (IPSO) method [39], and Evolutionary Particle Swarm Optimization (EPSO) method [40]. The performance for all these PSO types will be tested on 12 common mathematical test functions in the dimensions of 20, 30, 50, and 100. The maximum number of iterations for the algorithms was set to 1000. The remainder of this paper is organized in the following fashion. In Section 2, the basic operation of PSO, IPSO, and EPSO and their respective updated strategy in obtaining optimal solutions are briefly explained. The details of the proposed GPSO algorithm and the new updating strategy are described in Section 3. The simulation results between GPSO performance with the other PSO algorithms and the influence of parameter setting on GPSO are presented in Section 4, respectively. Finally, Section 5 concludes this paper.

2. Review of Particle Swarm Optimization and Its Variants

2.1. Particle Swarm Optimization (PSO) Algorithm. The original Particle Swarm Optimization (PSO) Algorithm proposed by Kennedy and Eberhart [28] in 1995 is adapted from the food searching behavior of birds and fish (particles). All these particles will move within the searching space towards the location of food (optimal location) in a specific speed and continuously change their respective positions to arrive at the destination. The position and velocity of i th particle in d -dimensional search space are represented as $x_i = (x_{i1}, x_{i2}, \dots, x_{id})$ and $v_i = (v_{i1}, v_{i2}, \dots, v_{id})$,

respectively. Their movement is guided by their own experience ($P_{\text{best}-i}^k$) and the experience of the other particles in a group (G_{best}^k). The best position which has been visited by the i th particle, until k iteration is recorded as $P_{\text{best}-i}^k = (P_{\text{best}-i1}^k, P_{\text{best}-i2}^k, \dots, P_{\text{best}-id}^k)$, while the best position among the $P_{\text{best}-i}^k$ in a group is defined as G_{best}^k . Each particle will update its velocity (v_i^{k+t}) and position (x_i^{k+1}) in the following manner:

$$\begin{aligned} v_i^{k+1} &= wv_i^{k+1} + c_1r_{1i}(P_{\text{best}-i}^k - x_i^k) + c_2r_{2i}(G_{\text{best}}^k - x_i^k), \\ x_i^{k+1} &= v_i^{k+1} + x_i^k, \end{aligned} \quad (1)$$

where k is the number of iterations, w is the inertia weight, v_i is the velocity of i th particle, c_1 and c_2 are the acceleration coefficients for cognitive and social components, respectively, and r_1 and r_2 are independently uniform distributed random numbers between 0 and 1.

2.2. Iteration Particle Swarm Optimization (IPSO) Algorithm. Iteration Particle Swarm Optimization (IPSO) is a modification PSO method proposed by Lee and Chen [39] to improve the solution quality and computing time of the algorithm. In the IPSO, three best values were used to update the velocity and the position of the particles, which are known as G_{best} , P_{best} , and I_{best} . The definition and the method to determine the P_{best} and G_{best} values in the IPSO are similar to the traditional PSO (discussed in Section 2.1). Meanwhile, the new parameter I_{best} is defined as the best value of the fitness function that has been achieved by any particle in the present iteration. In other words, I_{best} value is the P_{best} value that is randomly selected among the existing particles from the current population. In addition, the authors also introduced a dynamic acceleration constant parameter, c_3 value for I_{best} in IPSO. Therefore, the new velocity formula for the IPSO algorithm is [39]

$$\begin{aligned} v_i^{k+1} &= wv_i^{k+1} + c_1r_{1i}(P_{\text{best}-i}^k - x_i^k) + c_2r_{2i}(G_{\text{best}}^k - x_i^k) \\ &\quad + c_3(I_{\text{best}}^k - x_i^k), \\ c_3 &= c_1(1 - e^{-c_1k}), \end{aligned} \quad (2)$$

where c_3 is the acceleration constant that pulls each particle towards I_{best} . The value of c_3 will change according to c_1 and the number of iterations (k).

2.3. Evolutionary Particle Swarm Optimization (EPSO) Algorithm. The Evolutionary Particle Swarm Optimization (EPSO) Algorithm introduced by Angeline [40] utilizes the hybridization concept by combining the Evolutionary Programming (EP) with the traditional PSO. There are few methods in EP optimization methods, and one of them utilized the competition, sorting, and selection processes to select the survival particles and render the process in determining the optimum value quicker. By implementing this concept in PSO algorithm, it causes the particles in EPSO to move faster than its counterpart towards the optimal point.

TABLE 1: High dimensional benchmark functions used in experiments.

Test function	Range
$f_1(x) = \sum_{i=1}^n x_i^2$	$[-100, 100]^n$
$f_2(x) = \sum_{i=1}^n (10^6)^{(i-1)/(n-1)} x_i^2$	$[-100, 100]^n$
$f_3(x) = \sum_{i=1}^n i x_i^2$	$[-10, 10]^n$
$f_4(x) = \sum_{i=1}^n x_i ^{(i+1)}$	$[-10, 10]^n$
$f_5(x) = \sum_{i=1}^n (x_i + 0.5)^2$	$[-100, 100]^n$
$f_6(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	$[-32, 32]^n$
$f_7(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-10, 10]^n$
$f_8(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^n$
$f_9(x) = \sum_{i=1}^n [y_i^2 - 10 \cos(2\pi y_i) + 10] y_i^2 \begin{cases} x_i, & x_i < \frac{1}{2} \\ \frac{\text{round}(2x_i)}{2}, & x_i \geq \frac{1}{2} \end{cases}$	$[-5.12, 5.12]^n$
$f_{10}(x) = \frac{1}{n} \sum_{i=1}^n (x_i^4 - 16x_i^2 + 5x_i)$	$[-5, 5]^n$
$f_{11}(x) = \sum_{i=1}^n x_i \sin(x_i) + 0.1x_i $	$[-10, 10]^n$
$f_{12}(x) = \sum_{i=1}^{n-1} (x_i - 1)^2 1 + x_i \sin^2(3\pi x_i) $	$[-10, 10]^n$

Furthermore, the implementation of EP in PSO will also cause the process of finding the P_{best} and G_{best} for each iteration to become different and simpler [41]. The EPSO algorithm can get the P_{best} and G_{best} values just after the selection process based on the survival particle. Thus, the P_{best} and G_{best} values can be easily determined, and the new velocity and position are updated using (1).

3. A Novel Global Particle Swarm Optimization Algorithm

In this study, a novel Global Particle Swarm Optimization (GPSO) Algorithm is introduced. Certain modifications are made on the velocity formula of the traditional PSO algorithm, so that it would reach and obtain the best solution (global solution) quicker than the traditional PSO.

In the traditional PSO, the G_{best} having a better result is used in the updating process, while the remaining G_{best} is terminated (replaced). This means that all the G_{best} values that have been achieved at the previous iteration will not be referring back. In other words, there is no information sharing between the best values that are achieved at the previous iteration with the current results. By using the “Experience” concept, the knowledge and information about what had previously happened can be used as guides for accurate decisions. Therefore, the G_{best} value at any dimension (variable) and at any iteration will be randomly selected and

used to update the velocity value, and this variable is called E_{best} .

Furthermore, the acceleration coefficient for the E_{best} parameter (c_3) is obtained from the average of cognitive (c_1) and social (c_2) constant parameter values. Thus, the new velocity formula that updates the next position of particles in GPSO is

$$v_i^{k+1} = \omega^k v_i^k + c_1 r_{1i} (P_{\text{best}-i}^k - x_i^k) + c_2 r_{2i} (G_{\text{best}}^k - x_i^k) + c_3 r_{3i} (E_{\text{best}}^k - x_i^k), \quad (3)$$

where the 4th term in the GPSO velocity equation $|c_3 r_{3i} (E_{\text{best}}^k - x_i^k)|$ is referred to as the Improvement Factor (IF).

Figure 1 illustrates an example of the GPSO process to determine the E_{best}^k value. The optimization problem supposedly consists of 3 variables (X_1 , X_2 , and X_3); the number of particles is 6 (the positions of the particles are represented by different colours) and has reached the k th iteration (current iteration). Similar to the PSO algorithm, the G_{best} value for the population at each iteration will be obtained in order to update the next particles’ position. In this example, the population at m th iteration has the G_{best} value that is achieved by the “blue” particle. Therefore, the G_{best} values for each dimension (variable) at the m th iteration are $X_1^{m-\text{blue}}$, $X_2^{m-\text{blue}}$, and $X_3^{m-\text{blue}}$. For the current iteration, the G_{best} value is achieved by the “light purple” particle; thus,

TABLE 2: Best, worst, mean, and standard deviation values obtained by PSO, IPSO, EPSO, and GPSO through 50 independent runs on functions from f_1 to f_3 .

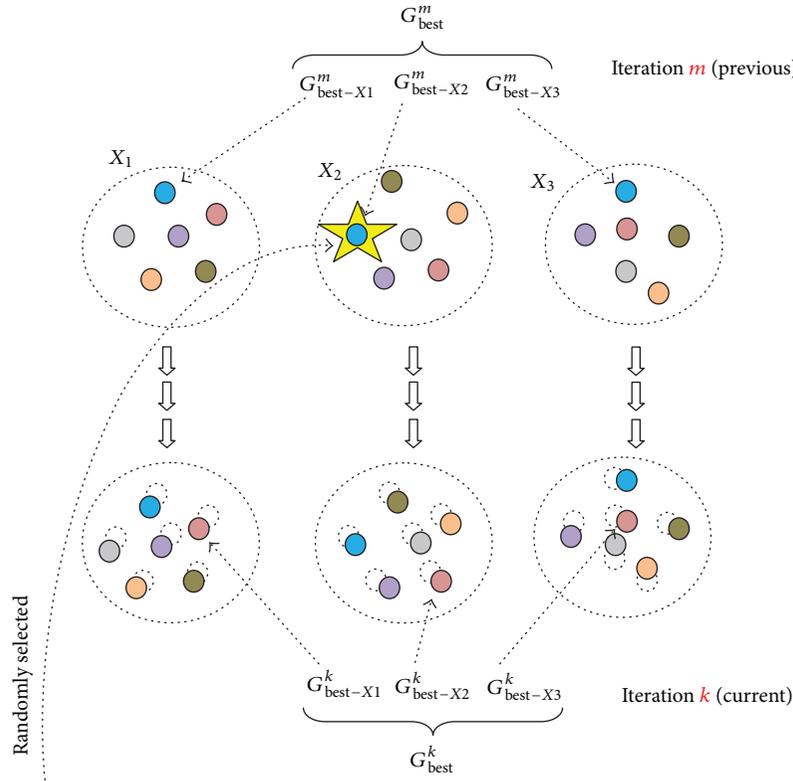
Function	Dim.	Min.		Best	Worst	Mean	SD
f_1	20	0	PSO	$4.09E + 01$	$1.19E + 03$	$3.15E + 02$	$2.68E + 02$
			IPSO	$2.81E + 01$	$5.13E + 02$	$1.99E + 02$	$1.33E + 02$
			EPSO	$1.55E + 03$	$2.17E + 04$	$5.30E + 03$	$3.53E + 03$
			GPSO	$8.18E - 09$	$1.31E - 05$	$2.22E - 06$	$2.58E - 06$
	30		PSO	$1.22E + 03$	$8.23E + 03$	$3.70E + 03$	$1.53E + 03$
			IPSO	$8.71E + 02$	$4.34E + 03$	$2.25E + 03$	$8.05E + 02$
			EPSO	$5.40E + 03$	$2.02E + 04$	$1.14E + 04$	$3.85E + 03$
			GPSO	$1.47E - 10$	$2.47E - 05$	$4.51E - 06$	$6.10E - 06$
	50		PSO	$4.37E + 03$	$1.77E + 04$	$9.67E + 03$	$3.29E + 03$
			IPSO	$3.36E + 03$	$9.31E + 03$	$6.08E + 03$	$1.46E + 03$
			EPSO	$9.70E + 03$	$3.99E + 04$	$2.35E + 04$	$7.05E + 03$
			GPSO	$3.90E - 07$	$1.73E - 05$	$5.45E - 06$	$4.68E - 06$
100	PSO	$5.80E + 03$	$2.73E + 04$	$1.55E + 04$	$5.62E + 03$		
	IPSO	$4.48E + 03$	$1.67E + 04$	$1.12E + 04$	$3.17E + 03$		
	EPSO	$2.84E + 04$	$6.85E + 04$	$4.74E + 04$	$1.06E + 04$		
	GPSO	$5.07E - 07$	$8.33E - 05$	$1.26E - 05$	$1.62E - 05$		
f_2	20	PSO	$2.09E + 06$	$1.33E + 08$	$1.77E + 07$	$2.14E + 07$	
		IPSO	$2.17E + 05$	$1.27E + 07$	$2.90E + 06$	$2.79E + 06$	
		EPSO	$1.36E + 06$	$1.56E + 08$	$4.23E + 07$	$3.83E + 07$	
		GPSO	$5.11E - 03$	$3.09E + 00$	$3.87E - 01$	$5.18E - 01$	
	30	PSO	$5.67E + 06$	$1.07E + 08$	$2.97E + 07$	$2.07E + 07$	
		IPSO	$2.36E + 06$	$3.69E + 07$	$1.33E + 07$	$8.51E + 06$	
		EPSO	$1.37E + 08$	$1.86E + 09$	$6.70E + 08$	$3.65E + 08$	
		GPSO	$5.01E - 04$	$3.26E + 00$	$4.23E - 01$	$6.44E - 01$	
	50	PSO	$2.50E + 07$	$4.77E + 08$	$1.46E + 08$	$9.27E + 07$	
		IPSO	$1.69E + 07$	$2.92E + 08$	$7.23E + 07$	$4.24E + 07$	
		EPSO	$2.14E + 07$	$5.28E + 08$	$1.82E + 08$	$1.16E + 08$	
		GPSO	$6.09E - 03$	$5.35E + 00$	$5.34E - 01$	$8.52E - 01$	
100	PSO	$2.69E + 08$	$1.37E + 09$	$6.64E + 08$	$2.64E + 08$		
	IPSO	$1.58E + 08$	$6.74E + 08$	$3.60E + 08$	$1.19E + 08$		
	EPSO	$1.02E + 09$	$6.44E + 09$	$2.25E + 09$	$1.10E + 09$		
	GPSO	$1.38E - 02$	$7.61E + 00$	$9.31E - 01$	$1.22E + 00$		
f_3	20	PSO	$1.21E + 01$	$3.07E + 02$	$1.09E + 02$	$6.45E + 01$	
		IPSO	$3.50E + 00$	$1.29E + 02$	$4.74E + 01$	$2.94E + 01$	
		EPSO	$6.62E + 01$	$7.40E + 02$	$3.80E + 02$	$1.52E + 02$	
		GPSO	$7.51E - 09$	$6.53E - 06$	$3.66E - 07$	$9.41E - 07$	
	30	PSO	$1.55E + 02$	$1.08E + 03$	$4.86E + 02$	$2.34E + 02$	
		IPSO	$1.12E + 02$	$5.77E + 02$	$2.98E + 02$	$1.12E + 02$	
		EPSO	$5.41E + 02$	$3.68E + 03$	$1.60E + 03$	$6.61E + 02$	
		GPSO	$1.53E - 09$	$4.54E - 06$	$6.32E - 07$	$9.40E - 07$	
	50	PSO	$1.11E + 03$	$3.77E + 03$	$2.05E + 03$	$6.45E + 02$	
		IPSO	$8.07E + 02$	$2.11E + 03$	$1.39E + 03$	$3.72E + 02$	
		EPSO	$2.90E + 03$	$1.08E + 04$	$5.95E + 03$	$1.75E + 03$	
		GPSO	$4.35E - 08$	$5.21E - 06$	$1.48E - 06$	$1.40E - 06$	
100	PSO	$5.15E + 03$	$2.34E + 04$	$1.49E + 04$	$3.92E + 03$		
	IPSO	$3.94E + 03$	$1.27E + 04$	$8.27E + 03$	$2.24E + 03$		
	EPSO	$1.69E + 04$	$4.02E + 04$	$2.74E + 04$	$5.43E + 03$		
	GPSO	$2.65E - 07$	$4.21E - 05$	$8.48E - 06$	$9.90E - 06$		

TABLE 3: Best, worst, mean, and standard deviation values obtained by PSO, IPSO, EPSO, and GPSO through 50 independent runs on functions from f_4 to f_6 .

Function	Dim.	Min.	Best	Worst	Mean	SD	
f_4	20	0	PSO	$8.46E + 00$	$5.17E + 05$	$1.25E + 04$	$7.29E + 04$
			IPSO	$1.11E - 03$	$5.41E + 03$	$3.32E + 02$	$9.93E + 02$
			EPSO	$3.69E + 02$	$2.70E + 08$	$1.15E + 07$	$4.24E + 07$
			GPSO	$5.60E - 13$	$7.61E - 09$	$4.98E - 10$	$1.16E - 09$
	30		PSO	$1.54E + 02$	$9.45E + 10$	$2.40E + 09$	$1.35E + 10$
			IPSO	$3.80E + 01$	$4.13E + 05$	$5.36E + 04$	$1.00E + 05$
			EPSO	$5.51E + 04$	$3.65E + 15$	$1.43E + 14$	$5.92E + 14$
			GPSO	$2.69E - 14$	$4.96E - 09$	$3.54E - 10$	$7.68E - 10$
	50		PSO	$1.16E + 10$	$1.00E + 23$	$2.01E + 21$	$1.41E + 22$
			IPSO	$1.34E + 05$	$3.02E + 14$	$1.72E + 13$	$5.91E + 13$
			EPSO	$1.56E + 13$	$1.00E + 29$	$2.01E + 27$	$1.41E + 28$
			GPSO	$1.60E - 12$	$2.07E - 09$	$3.33E - 10$	$4.83E - 10$
100	PSO	$1.06E + 26$	$1.02E + 54$	$2.04E + 52$	$1.44E + 53$		
	IPSO	$9.77E + 15$	$4.13E + 36$	$9.76E + 34$	$5.86E + 35$		
	EPSO	$1.05E + 39$	$7.97E + 62$	$1.67E + 61$	$1.13E + 62$		
	GPSO	$1.97E - 12$	$3.48E - 09$	$3.89E - 10$	$5.45E - 10$		
f_5	20	PSO	$2.06E + 02$	$3.05E + 03$	$1.12E + 03$	$6.33E + 02$	
		IPSO	$1.25E + 02$	$2.23E + 03$	$6.55E + 02$	$4.50E + 02$	
		EPSO	$2.17E + 03$	$1.17E + 04$	$5.78E + 03$	$2.26E + 03$	
		GPSO	$6.47E - 09$	$3.65E - 05$	$2.91E - 06$	$5.62E - 06$	
	30	PSO	$1.85E + 03$	$1.17E + 04$	$3.78E + 03$	$1.76E + 03$	
		IPSO	$7.13E + 02$	$4.11E + 03$	$2.04E + 03$	$7.82E + 02$	
		EPSO	$3.71E + 03$	$1.91E + 04$	$1.06E + 04$	$3.34E + 03$	
		GPSO	$5.22E - 08$	$2.20E - 05$	$3.63E - 06$	$4.91E - 06$	
	50	PSO	$4.47E + 03$	$1.63E + 04$	$9.70E + 03$	$2.91E + 03$	
		IPSO	$2.88E + 03$	$1.02E + 04$	$6.11E + 03$	$1.70E + 03$	
		EPSO	$1.11E + 04$	$4.71E + 04$	$2.45E + 04$	$7.81E + 03$	
		GPSO	$7.60E - 08$	$5.50E - 05$	$7.55E - 06$	$9.15E - 06$	
100	PSO	$1.19E + 04$	$4.48E + 04$	$2.70E + 04$	$7.65E + 03$		
	IPSO	$9.05E + 03$	$2.36E + 04$	$1.57E + 04$	$2.74E + 03$		
	EPSO	$2.36E + 04$	$9.09E + 04$	$5.08E + 04$	$1.58E + 04$		
	GPSO	$3.02E - 08$	$1.17E - 04$	$1.20E - 05$	$1.95E - 05$		
f_6	20	PSO	$7.34E + 00$	$1.57E + 01$	$1.08E + 01$	$1.88E + 00$	
		IPSO	$4.68E + 00$	$1.08E + 01$	$7.95E + 00$	$1.57E + 00$	
		EPSO	$1.25E + 01$	$1.84E + 01$	$1.53E + 01$	$1.55E + 00$	
		GPSO	$4.20E - 05$	$1.80E - 03$	$4.45E - 04$	$3.77E - 04$	
	30	PSO	$9.48E + 00$	$1.69E + 01$	$1.27E + 01$	$1.45E + 00$	
		IPSO	$7.48E + 00$	$1.27E + 01$	$1.00E + 01$	$1.16E + 00$	
		EPSO	$1.30E + 01$	$1.80E + 01$	$1.63E + 01$	$1.14E + 00$	
		GPSO	$5.59E - 05$	$1.38E - 03$	$4.23E - 04$	$2.83E - 04$	
	50	PSO	$1.13E + 01$	$1.59E + 01$	$1.39E + 01$	$1.02E + 00$	
		IPSO	$1.02E + 01$	$1.35E + 01$	$1.19E + 01$	$8.40E - 01$	
		EPSO	$1.47E + 01$	$1.88E + 01$	$1.73E + 01$	$9.16E - 01$	
		GPSO	$6.53E - 05$	$1.28E - 03$	$4.25E - 04$	$2.91E - 04$	
100	PSO	$1.34E + 01$	$1.75E + 01$	$1.52E + 01$	$9.02E - 01$		
	IPSO	$1.08E + 01$	$1.48E + 01$	$1.28E + 01$	$7.88E - 01$		
	EPSO	$1.52E + 01$	$1.95E + 01$	$1.76E + 01$	$9.96E - 01$		
	GPSO	$2.62E - 05$	$1.85E - 03$	$3.48E - 04$	$3.05E - 04$		

TABLE 4: Best, worst, mean, and standard deviation values obtained by PSO, IPSO, EPSO, and GPSO through 50 independent runs on functions from f_7 to f_9 .

Function	Dim.	Min.	Best	Worst	Mean	SD	
f_7	20	0	PSO	$4.64E + 02$	$9.78E + 03$	$2.51E + 03$	$2.23E + 03$
			IPSO	$1.44E + 02$	$7.65E + 03$	$1.21E + 03$	$1.22E + 03$
			EPSO	$4.33E + 03$	$2.37E + 05$	$4.14E + 04$	$4.54E + 04$
			GPSO	$1.10E - 07$	$1.35E - 04$	$2.60E - 05$	$3.20E - 05$
	30		PSO	$1.79E + 03$	$3.22E + 04$	$1.17E + 04$	$7.31E + 03$
			IPSO	$7.16E + 02$	$3.34E + 04$	$5.47E + 03$	$5.08E + 03$
			EPSO	$1.62E + 04$	$4.05E + 05$	$9.36E + 04$	$7.21E + 04$
			GPSO	$5.36E - 07$	$3.67E - 04$	$3.26E - 05$	$5.57E - 05$
	50		PSO	$6.72E + 03$	$1.44E + 05$	$4.39E + 04$	$2.78E + 04$
			IPSO	$5.71E + 03$	$8.60E + 04$	$2.27E + 04$	$1.46E + 04$
			EPSO	$7.24E + 04$	$1.01E + 06$	$3.07E + 05$	$1.95E + 05$
			GPSO	$1.23E - 07$	$1.15E - 04$	$3.97E - 05$	$3.53E - 05$
100	PSO	$7.49E + 04$	$6.85E + 05$	$2.18E + 05$	$1.47E + 05$		
	IPSO	$2.28E + 04$	$2.43E + 05$	$9.55E + 04$	$4.66E + 04$		
	EPSO	$2.52E + 05$	$2.37E + 06$	$8.59E + 05$	$4.58E + 05$		
	GPSO	$1.97E - 07$	$6.07E - 04$	$8.64E - 05$	$1.34E - 04$		
f_8	20	PSO	$3.93E + 01$	$1.15E + 02$	$7.70E + 01$	$1.55E + 01$	
		IPSO	$2.74E + 01$	$8.14E + 01$	$4.95E + 01$	$1.20E + 01$	
		EPSO	$9.37E + 01$	$2.02E + 02$	$1.38E + 02$	$2.58E + 01$	
		GPSO	$1.17E - 07$	$1.78E - 04$	$2.56E - 05$	$3.74E - 05$	
	30	PSO	$9.38E + 01$	$2.24E + 02$	$1.48E + 02$	$2.63E + 01$	
		IPSO	$6.06E + 01$	$1.89E + 02$	$1.19E + 02$	$2.41E + 01$	
		EPSO	$1.36E + 02$	$3.05E + 02$	$2.35E + 02$	$3.69E + 01$	
		GPSO	$4.77E - 07$	$2.41E - 04$	$2.96E - 05$	$4.28E - 05$	
	50	PSO	$1.90E + 02$	$3.99E + 02$	$3.19E + 02$	$4.43E + 01$	
		IPSO	$1.87E + 02$	$3.09E + 02$	$2.70E + 02$	$2.87E + 01$	
		EPSO	$3.17E + 02$	$5.58E + 02$	$4.54E + 02$	$4.93E + 01$	
		GPSO	$8.72E - 08$	$2.75E - 04$	$4.82E - 05$	$6.43E - 05$	
100	PSO	$6.41E + 02$	$9.16E + 02$	$7.97E + 02$	$5.75E + 01$		
	IPSO	$5.85E + 02$	$7.71E + 02$	$7.01E + 02$	$4.23E + 01$		
	EPSO	$9.00E + 02$	$1.16E + 03$	$1.03E + 03$	$5.83E + 01$		
	GPSO	$2.36E - 07$	$5.13E - 04$	$8.11E - 05$	$1.11E - 04$		
f_9	20	PSO	$3.06E + 01$	$1.18E + 02$	$6.57E + 01$	$1.79E + 01$	
		IPSO	$1.39E + 01$	$8.43E + 01$	$3.99E + 01$	$1.79E + 01$	
		EPSO	$6.05E + 01$	$1.70E + 02$	$1.15E + 02$	$2.41E + 01$	
		GPSO	$1.01E - 06$	$5.34E - 04$	$9.27E - 05$	$9.80E - 05$	
	30	PSO	$8.00E + 01$	$2.08E + 02$	$1.36E + 02$	$3.46E + 01$	
		IPSO	$3.96E + 01$	$1.58E + 02$	$8.35E + 01$	$2.78E + 01$	
		EPSO	$1.40E + 02$	$3.17E + 02$	$2.16E + 02$	$3.72E + 01$	
		GPSO	$7.94E - 06$	$7.71E - 04$	$1.42E - 04$	$1.59E - 04$	
	50	PSO	$2.08E + 02$	$3.96E + 02$	$3.11E + 02$	$4.19E + 01$	
		IPSO	$1.09E + 02$	$3.34E + 02$	$2.04E + 02$	$3.91E + 01$	
		EPSO	$3.12E + 02$	$5.76E + 02$	$4.22E + 02$	$6.29E + 01$	
		GPSO	$2.81E - 07$	$1.07E - 03$	$1.85E - 04$	$2.08E - 04$	
100	PSO	$6.32E + 02$	$1.03E + 03$	$8.08E + 02$	$8.19E + 01$		
	IPSO	$4.91E + 02$	$7.10E + 02$	$5.94E + 02$	$5.05E + 01$		
	EPSO	$7.23E + 02$	$1.20E + 03$	$9.73E + 02$	$9.09E + 01$		
	GPSO	$8.56E - 06$	$1.48E - 03$	$3.24E - 04$	$3.52E - 04$		



$$E_{best}^k = \text{random}(G_{best}^1, G_{best}^2, \dots, G_{best}^m, \dots, G_{best}^{k-1}, G_{best}^k)_{X1, X2, X3} = G_{best-X2}^m$$

FIGURE 1: Illustration for Global Particle Swarm Optimization concept.

the G_{best} value can be presented as $X_j^{k-\text{light purple}}$, where $j = 1, 2,$ and 3 (number of dimensions).

However, in the GPSO, an additional parameter is needed for the updating process, which is the E_{best} value. From the previous definition, the E_{best} value is obtained from the G_{best} value that is randomly selected at any iteration and dimension. By using the Gaussian distributed random function, the G_{best} value from 2nd dimension (2nd variable) at the m th iteration is selected to be the E_{best}^k value for the current iteration. Thus, in this example, the IF for the current iteration population is

$$\begin{aligned} IF^k &= c_3 r_{3i} (G_{best-X2}^m - x_i^k) \\ &= \left[\frac{c_2 + c_2}{2} \right] r_{3i} (G_{best-X2}^m - x_i^k). \end{aligned} \tag{4}$$

From the above description, it can be noticed that the E_{best} value for each iteration will be different for each time. Unlike the G_{best} concept, where the comparison process is needed to determine the surviving G_{best} value for the current iteration, the E_{best} parameter will not require any comparison, and the possibility of having similar E_{best} values in a next iteration is minuscule, especially for high dimensional problems. Therefore, the particles will learn and know about different “Experience” for each process. Furthermore, the Improvement Factor (IF) will also help the algorithm have

the suitable velocity value for updating the next particles’ positions. All these factors made the GPSO superior to its original counterpart. Figure 2 and Algorithm 1 show the flow chart for the GPSO algorithm in solving the high dimensional optimization problem and the corresponding MATLAB codes for the IF parameter.

4. Result and Discussion

Twelve (12) well-known classical benchmark functions that consist of unimodal, multimodal, separable, and non-separable types are used to evaluate the performance of the CPSO, IWPSO, IPSO and GPSO algorithms, respectively, as shown in Table 1.

In order to have a fair comparison, all of the PSO methods in the analysis utilized the same parametric values. The population size for the particles is set to 20 ($N = 20$), with the cognitive and social components (c_1 and c_2) values set to 0.5. The algorithm will run until the maximum iteration (max. iter. = 1000). Furthermore, the performance of these PSOs will also be tested in 4 different dimensions, which are 20, 30, 50, and 100, and each test set is executed 50 times with the new random population values.

4.1. Results from Benchmark Simulation. Tables 2, 3, 4, and 5 show the results of the best, worst, mean, and standard

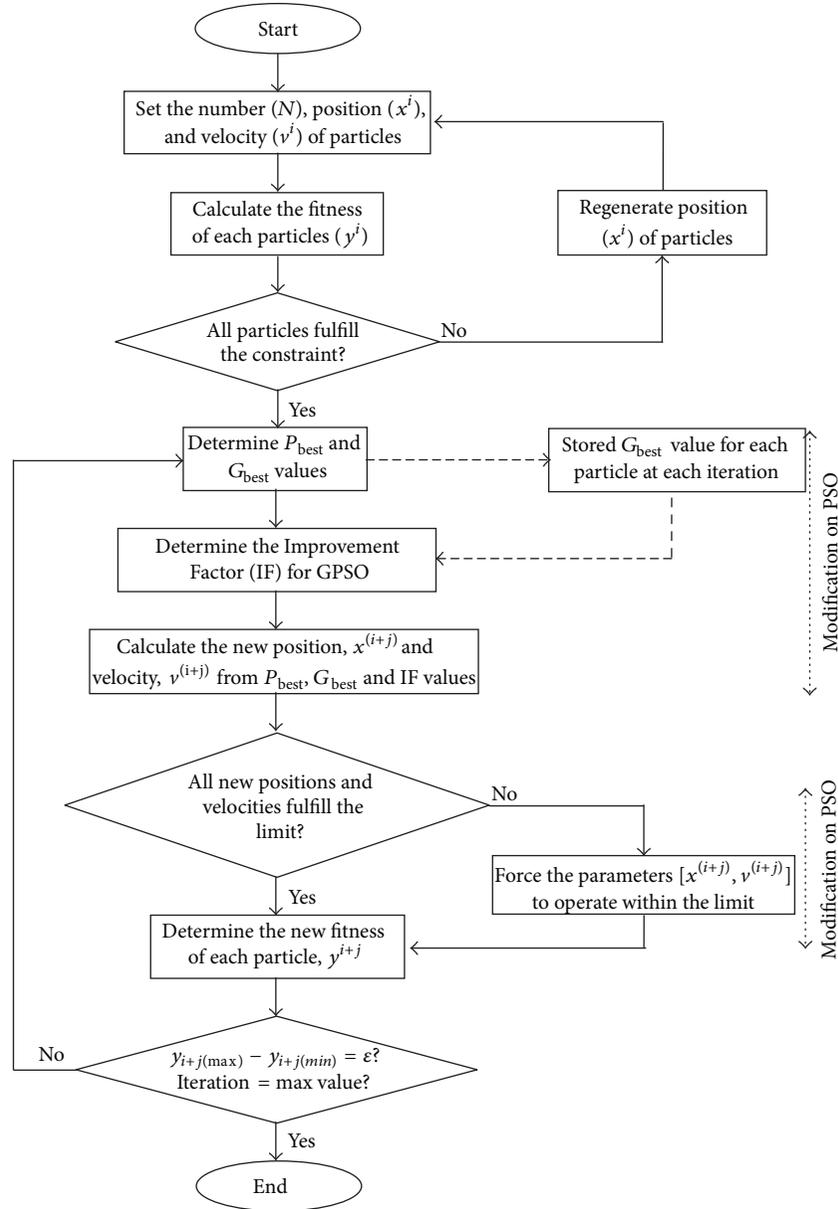


FIGURE 2: The process of Global Particle Swarm Optimization (GPSO) Algorithm.

deviation (SD) values that are achieved by the PSO, IPSO, EPSO, and GPSO for all 12 benchmark functions in four different dimensional sizes, respectively. The standard optimal value for all functions in this analysis is zero except for function 10 (f_{10}), where the standard optimal value for f_{10} is -78.33 . Since the dimension of each test function is very high (20, 30, 50, and 100 dimensions), the PSO, EPSO, and IPSO algorithms could not provide the global optimal value in the same manner that the GPSO could.

Figure 3 shows the example of the best results that are achieved by all PSO methods in different functions and dimensions. From the results, it can be seen that the concept of “Experience” in GPSO has helped the algorithm obtain the lowest convergence value compared to other PSOs. Not only

that, but the GPSO has also reached the global value faster than others, either when the global minimum value is zero (Figures 3(a) and 3(b)) or not (Figure 3(c)). For example, in Figure 3(c), the EPSO is the slowest algorithm that makes the high dimensional problem arrive near its convergence solution (nearly 200 iterations) compared to PSO and IPSO, whilst GPSO only required a small number of iteration (12 iterations) to do the same. This makes GPSO the fastest algorithm in terms of obtaining convergence solution in high dimensional problems and the lowest optimal results when compared to other PSOs methods.

On top of that, the limitation on 1000 iterations is not the cause for the PSO, EPSO, and IPSO methods for not realizing the best optimal value. Although the numbers of iterations

TABLE 5: Best, worst, mean, and standard deviation values obtained by PSO, IPSO, EPSO, and GPSO through 50 independent runs on functions from f_{10} to f_{12} .

Function	Dim.	Min.	Best	Worst	Mean	SD	
f_{10}	20	-78.3324	PSO	-70.2000	-52.8000	-61.2000	3.5700
			IPSO	-68.3000	-57.7000	-62.0000	2.5100
			EPSO	-60.6000	-40.3000	-51.7000	4.8822
			GPSO	-78.3000	-75.1000	-77.9000	0.6716
	30		PSO	-63.2295	-49.5756	-56.2019	3.0588
			IPSO	-64.0370	-51.9089	-57.1815	2.7452
			EPSO	-56.0959	-39.0376	-47.4010	3.6965
			GPSO	-78.3323	-78.3323	-78.3323	0.0000
	50		PSO	-60.4579	-44.9971	-51.2624	3.1199
			IPSO	-56.2791	-46.6235	-51.8837	2.3331
			EPSO	-50.3900	-35.8658	-43.2473	3.2610
			GPSO	-78.3323	-78.3323	-78.3323	0.0000
100	PSO	-51.1000	-40.1000	-45.0000	2.5976		
	IPSO	-53.2000	-43.0000	-47.4000	2.0405		
	EPSO	-38.7000	-31.9000	-35.4000	1.7196		
	GPSO	-78.3000	-76.1000	-78.0000	0.3840		
f_{11}	20	PSO	3.06E + 00	1.29E + 01	7.89E + 00	2.31E + 00	
		IPSO	1.50E + 00	8.32E + 00	3.51E + 00	1.33E + 00	
		EPSO	9.03E + 00	2.56E + 01	1.45E + 01	3.76E + 00	
		GPSO	1.27E - 05	1.46E - 04	5.62E - 05	3.10E - 05	
	30	PSO	6.60E + 00	2.74E + 01	1.67E + 01	3.87E + 00	
		IPSO	4.28E + 00	1.44E + 01	9.31E + 00	2.13E + 00	
		EPSO	1.38E + 01	4.03E + 01	2.58E + 01	6.66E + 00	
		GPSO	1.80E - 05	1.64E - 04	7.93E - 05	4.18E - 05	
	50	PSO	2.32E + 01	5.53E + 01	3.60E + 01	7.14E + 00	
		IPSO	1.58E + 01	3.40E + 01	2.31E + 01	3.56E + 00	
		EPSO	3.10E + 01	7.10E + 01	5.45E + 01	7.73E + 00	
		GPSO	2.48E - 05	3.85E - 04	1.26E - 04	8.52E - 05	
100	PSO	5.91E + 01	1.32E + 02	8.94E + 01	1.35E + 01		
	IPSO	4.66E + 01	7.38E + 01	5.98E + 01	5.72E + 00		
	EPSO	8.55E + 01	1.54E + 02	1.17E + 02	1.66E + 01		
	GPSO	3.97E - 05	7.27E - 04	2.68E - 04	1.58E - 04		
f_{12}	20	PSO	1.23E + 01	1.03E + 02	4.55E + 01	2.02E + 01	
		IPSO	4.72E + 00	4.97E + 01	2.22E + 01	9.20E + 00	
		EPSO	3.10E + 01	2.09E + 02	1.17E + 02	3.92E + 01	
		GPSO	1.41E - 07	2.29E - 05	5.91E - 06	6.27E - 06	
	30	PSO	5.68E + 01	2.56E + 02	1.15E + 02	4.38E + 01	
		IPSO	1.72E + 01	1.05E + 02	5.24E + 01	1.76E + 01	
		EPSO	9.89E + 01	3.54E + 02	2.11E + 02	6.05E + 01	
		GPSO	8.08E - 08	6.49E - 05	1.14E - 05	1.34E - 05	
	50	PSO	1.14E + 02	3.91E + 02	2.14E + 02	5.72E + 01	
		IPSO	7.01E + 01	1.98E + 02	1.30E + 02	2.79E + 01	
		EPSO	1.89E + 02	6.68E + 02	4.38E + 02	1.23E + 02	
		GPSO	5.00E - 07	1.50E - 04	2.25E - 05	2.59E - 05	
100	PSO	3.55E + 02	9.73E + 02	5.96E + 02	1.40E + 02		
	IPSO	2.47E + 02	4.99E + 02	3.57E + 02	5.63E + 01		
	EPSO	5.21E + 02	1.38E + 03	1.03E + 03	1.70E + 02		
	GPSO	4.74E - 07	4.71E - 04	4.28E - 05	7.74E - 05		

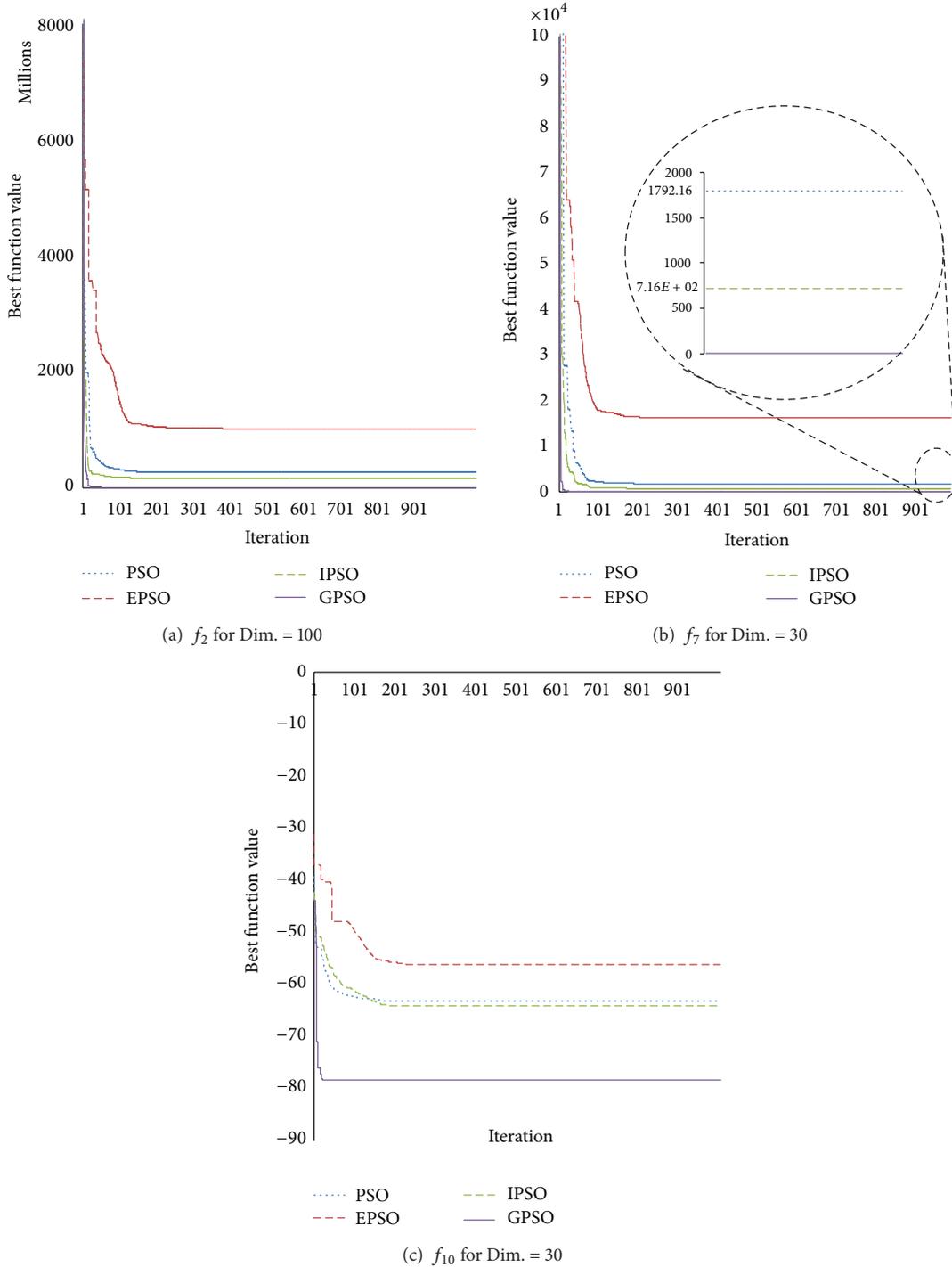


FIGURE 3: The performance of PSO, EPSO, IPSO, and GPSO in solving high dimension problems.

increase to 20000, the results of optimization problem is still give the same optimal value given by 1000 iterations. It means that, for the high dimensional problems, these three algorithms tend to be trapped at the local optimal solution. Therefore, the E_{best} parameter not only gives a faster convergence solution, it also helped the algorithm to avoid being trapped at the local optimal solution. Thus, from the

“best” results that are archived by all PSO methods in Tables 2 to 5, the GPSO can provide the optimal results very close to or similar to the standard minimum value regardless of the dimension size of the problems.

Furthermore, the GPSO can provide consistent results for all 50-sample tests. This can be seen from the standard deviation (SD) value where the SD values for GPSO are

TABLE 6: The average (mean) and standard deviation values of GPSO with various acceleration coefficients for all functions.

Function	Performance of GPSO for different $c_1, c_2,$ and c_3 values in 50 trials for Dim. = 50 cases									
	$c_1 = c_2 = c_3 = 0.2$		$c_1 = c_2 = c_3 = 0.5$		$c_1 = c_2 = c_3 = 1.0$		$c_1 = c_2 = c_3 = 1.5$		$c_1 = c_2 = c_3 = 2.0$	
	Average	Std Dev.	Average	Std Dev.	Average	Std Dev.	Average	Std Dev.	Average	Std Dev.
f_1	$1.02E-06$	$1.40E-06$	$5.45E-06$	$4.68E-06$	$7.37E-04$	$8.00E-04$	$4.69E-01$	$5.46E-01$	$3.42E+01$	$3.24E+01$
f_2	$2.32E-07$	$2.41E-07$	$1.48E-06$	$1.40E-06$	$1.60E-04$	$2.16E-04$	$2.27E-01$	$3.10E-01$	$9.45E+00$	$7.47E+00$
f_3	$7.75E-11$	$1.44E-10$	$3.33E-10$	$4.83E-10$	$3.79E-08$	$5.18E-08$	$1.12E-05$	$1.35E-05$	$1.13E-03$	$1.19E-03$
f_4	$9.39E-07$	$8.96E-07$	$7.55E-06$	$9.15E-06$	$5.83E-04$	$6.42E-04$	$5.56E-01$	$6.05E-01$	$3.23E+01$	$3.16E+01$
f_5	$1.84E-04$	$9.71E-05$	$4.25E-04$	$2.91E-04$	$3.57E-03$	$1.97E-03$	$2.22E-01$	$1.86E-01$	$2.39E+00$	$7.81E-01$
f_6	$8.81E-06$	$1.30E-05$	$3.97E-05$	$3.53E-05$	$2.97E-03$	$2.63E-03$	$3.30E-01$	$4.93E-01$	$2.80E+01$	$2.76E+01$
f_7	$1.79E-06$	$3.42E-06$	$4.82E-05$	$6.43E-05$	$3.63E-03$	$4.23E-03$	$1.30E-01$	$3.87E-01$	$1.26E+01$	$1.46E+01$
f_8	$1.05E-05$	$2.04E-05$	$1.85E-04$	$2.08E-04$	$2.92E-03$	$3.61E-03$	$5.42E-02$	$7.07E-02$	$6.27E+00$	$6.51E+00$
f_9	$-7.83E+1$	$2.85E-09$	$-7.83E+1$	$2.21E-08$	$-7.83E+1$	$7.68E-07$	$-7.83E+1$	$8.78E-04$	$-7.77E+1$	$4.01E+00$
f_{10}	$8.02E-05$	$4.92E-05$	$1.26E-04$	$8.52E-05$	$9.97E-04$	$7.55E-04$	$1.68E-02$	$9.63E-03$	$3.15E-01$	$3.05E-01$
f_{11}	$2.01E-06$	$2.66E-06$	$2.25E-05$	$2.59E-05$	$1.43E-03$	$1.75E-03$	$9.45E-01$	$9.93E-01$	$1.10E+01$	$5.69E+00$
f_{12}	$8.81E-06$	$1.30E-05$	$3.97E-05$	$3.53E-05$	$2.97E-03$	$2.63E-03$	$3.30E-01$	$4.93E-01$	$2.80E+01$	$2.76E+01$

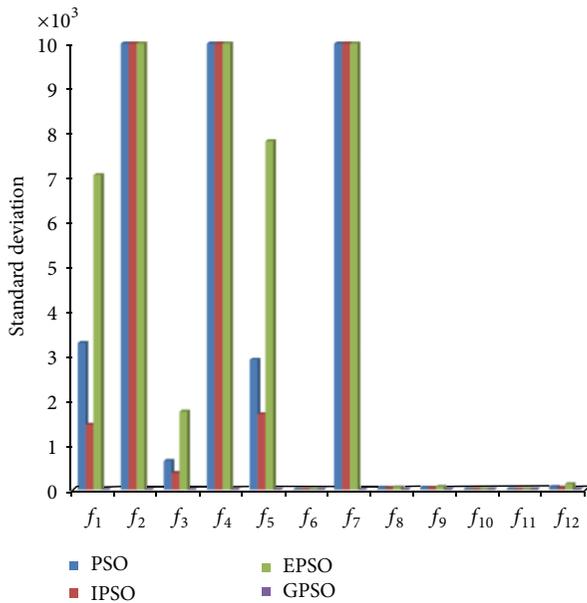


FIGURE 4: Comparison of standard deviation between GPSO and other PSOs for all functions with Dim. = 50.

the lowest compared to the results from the other three PSOs (Tables 2–5: column “SD”). Since the SD value indicates how close the results that are obtained from the 50 different trials to its average value (mean value), the largest of the SD value shows that the algorithm could not provide a consistent result in the analysis. This occurred due to the algorithms that are always trapped in the local results, especially for high dimensional problem. Figure 4 shows the SD values for all functions when the dimension is 50.

It can be clearly seen that the SD values for $f_2, f_4,$ and f_7 for PSO, EPSO, and IPSO, respectively, are very high compared to GPSO results. From Table 3, the “best” values that is achieved by PSO, EPSO, and IPSO in f_4 for dimension 50 are $1.16 \times 10^{10}, 1.34 \times 10^5,$ and $1.56 \times 10^{13},$ while the “worst”

value that were achieved by the algorithms are $1.00 \times 10^{23}, 3.02 \times 10^{14},$ and $1.00 \times 10^{29},$ respectively. The large difference between the “best” and the “worst” values influences the SD values. However, for functions $f_6, f_{10},$ and $f_{11},$ the SD results for all the PSO methods are smaller than the other functions results. Therefore, the PSO, EPSO, and IPSO can still give consistent results in some of the high dimensional problems, but not as well as the GPSO’s results. Overall, since the SD value for GPSO in all of the test functions and in all of the dimensions is very small (≈ 0), it can be concluded that the GPSO can provide consistent results in any high dimensional problem, as shown in Tables 2–5 (columns “best” and “worst”).

In summary, the GPSO algorithm can reach the global optimal value within 1000 iterations for all cases on high dimensional test functions. Not only that, but the GPSO also has a very low standard deviation value compared to the PSO, EPSO, and IPSO. Therefore, the GPSO has shown very high efficiency in solving any high dimensional numerical optimization problem.

4.2. Impacts of Acceleration Coefficient Selection for GPSO Algorithm. From the previous sections, the GPSO algorithm demonstrated the best performance in solving the high dimension numerical optimization problems. However, based on the velocity formula in (3), the settings of the acceleration value will also influence GPSO’s performance, albeit in a different manner. Thus, the investigation on the suitable range for coefficient acceleration in solving high dimensional problems is needed. The dimension for all the benchmark functions that are used in this analysis is 50. The maximum iteration number still remains the same, which is 1000 iterations, and the average results and the standard deviation values from 50 runs are presented in Table 6.

From Table 6, the GPSO algorithm can provide the average results near the optimal solution when the acceleration (c_i) value is between 0.2 and 1.5, where $i = 1, 2,$ and 3. For case $c_i = 0.2, c_i = 0.5,$ and $c_i = 1.0,$ all the average results for

```

for  $i = 1 : N$ 
    Ch_iter = fix(rand*t) + 1;
    Ch_part = fix(rand*D) + 1;
    while (Ch_part  $\neq$  i)
        Ch_part = fix(rand*D) + 1;
    end;
    IF ( $i, :, t$ ) =  $c_3$ *rand*(G_best(1, Ch_part, Ch_iter) - X( $i, :, t$ ));
End
V( $:, :, t + 1$ ) = W( $t$ )*V( $:, :, t$ ) +  $c_1$ *rand*(P_best( $:, :, t$ ) - X( $:, :, t$ ))
+  $c_2$ *rand*(G( $:, :, t$ ) - X( $:, :, t$ )) + IF( $:, :, t$ );
Where  $N$ : number of iteration,  $t$ : current iteration,  $D$ : Dimension,
 $i$ : particle's number,  $X$ : variable,  $G$ : the current  $g_{best}$  result

```

ALGORITHM 1: The coding for determined IF and updated position in GPSO.

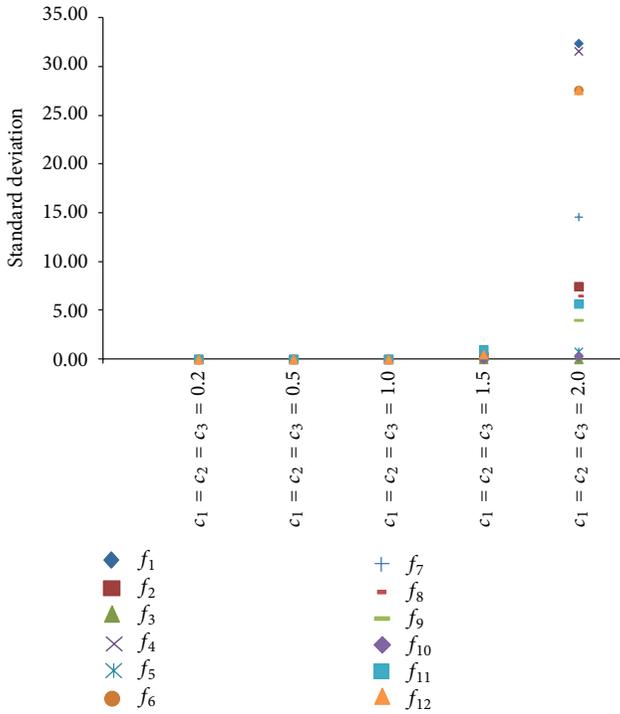


FIGURE 5: Standard deviation results for GPSO in varying the acceleration values.

f_1 to f_8 and f_{10} to f_{12} are between 10^{-11} and 10^{-3} . Since the standard optimal value for these functions is zero, the GPSO gave the average results that are very close to the standard optimal value. For the acceleration coefficient $c_i = 1.5$, the f_1 to f_8 and f_{10} to f_{12} functions gave the optimal value from 10^{-5} to 10^{-1} . For the f_8 function, the average results given by $c_i = 0.2$ until $c_i = 1.5$ are similar, which is -7.83×10^1 , and equal to the standard optimal results (-78.33). However, when the acceleration coefficients are set to 2.0, the results of the GPSO are not as good as the results when the acceleration values are less than 1.5, where some of the average value that is obtained by GPSO are in the tens, despite the fact that the optimal results should be zero.

Thus, in order to optimize the GPSO's performance in solving high dimensional problems, the acceleration coefficient that had been set by the user must be less than 1.5.

Figure 5 shows the standard deviation value for all of the functions when the acceleration varies from 0.2 and 2.0. From the results, when the acceleration coefficient values for GPSO are between 0.2 to 1.0 ($c_i = 0.2$, $c_i = 0.5$, and $c_i = 1.0$), the SD value for the algorithm is equal to zero for all functions.

In other words, GPSO is capable of providing consistent results, when the acceleration coefficient values are within range. However, when $c_i = 1.5$, some of the functions will have higher SD values, such as f_{11} , where the SD value is near to 1.0. The worst condition occurred when $c_i = 2.0$, where almost all functions have SD values higher than 0, except for functions 3 and 10. Therefore, by considering the average value in 50-test sets' performance and the SD results, it can be concluded that the suitable value on acceleration coefficient for the GPSO algorithm must be less than 1.0 in order to realize the best performance for GPSO in solving high dimensional problems.

5. Conclusion

This paper proposed a new algorithm for Particle Swarm Optimization, which is known as Global Particle Swarm Optimization (GPSO), to determine the global optimal value for high dimensional optimization problems. The performance of GPSO is tested on 12 classical benchmark high dimensional test functions and compared with 3 other PSO methods, which are Original Particle Swarm Optimization (PSO), Evolutionary Particle Swarm Optimization (EPSO), and Iteration Particle Swarm Optimization (IPSO).

The simulation results showed that the GPSO method performed well in solving high dimensional problems. The implementation of E_{best} in the GPSO did not only help the algorithm reach the global optimal results in a few iterations, but it also prevented the GPSO from being trapped in local optimal results. Furthermore, the GPSO algorithm also gave consistent results, even with different initial values based on the smallest standard deviation obtained in 50-test sets. From the acceleration coefficient varying analysis, the suitable value for the GPSO algorithm must be less than 1.0 in order to

realize the best performance in solving high dimensional problems.

Therefore, from all the results shown by the GPSO algorithm, it can be concluded that GPSO is the superior method in solving high dimensional numerical optimization problems compared to PSO, EPSO, and IPSO algorithms.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] T. Navalertporn and N. V. Afzulpurkar, "Optimization of tile manufacturing process using particle swarm optimization," *Swarm and Evolutionary Computation*, vol. 1, no. 2, pp. 97–109, 2011.
- [2] G. S. Chyan and S. G. Ponnambalam, "Obstacle avoidance control of redundant robots using variants of particle swarm optimization," *Robotics and Computer-Integrated Manufacturing*, vol. 28, no. 2, pp. 147–153, 2012.
- [3] Q. Tang and P. Eberhard, "A PSO-based algorithm designed for a swarm of mobile robots," *Structural and Multidisciplinary Optimization*, vol. 44, no. 4, pp. 483–498, 2011.
- [4] D. N. Jeyakumar, T. Jayabarathi, and T. Raghunathan, "Particle swarm optimization for various types of economic dispatch problems," *International Journal of Electrical Power and Energy Systems*, vol. 28, no. 1, pp. 36–42, 2006.
- [5] Q. Kang, T. Lan, Y. Yan, L. Wang, and Q. Wu, "Group search optimizer based optimal location and capacity of distributed generations," *Neurocomputing*, vol. 78, no. 1, pp. 55–63, 2012.
- [6] J.-Y. Kim, K.-J. Mun, H.-S. Kim, and J. H. Park, "Optimal power system operation using parallel processing system and PSO algorithm," *International Journal of Electrical Power and Energy Systems*, vol. 33, no. 8, pp. 1457–1461, 2011.
- [7] Y. W. Ying, J. Zhou, C. Z. Chao Zhou, Y. W. Wang, H. Q. Qin, and Y. L. Lu, "An improved self-adaptive PSO technique for short-term hydrothermal scheduling," *Expert Systems with Applications*, vol. 39, no. 3, pp. 2288–2295, 2012.
- [8] G. Isazadeh, R.-A. Hooshmand, and A. Khodabakhshian, "Modeling and optimization of an adaptive dynamic load shedding using the ANFIS-PSO algorithm," *Simulation*, vol. 88, no. 2, pp. 181–196, 2012.
- [9] H. Liu, Z. Cai, and Y. Wang, "Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization," *Applied Soft Computing Journal*, vol. 10, no. 2, pp. 629–640, 2010.
- [10] A. Farshidianfar, A. Saghafi, S. M. Kalami, and I. Saghafi, "Active vibration isolation of machinery and sensitive equipment using H_{∞} control criterion and particle swarm optimization method," *Meccanica*, vol. 47, no. 2, pp. 437–453, 2012.
- [11] S. M. Seyedpoor, J. Salajegheh, and E. Salajegheh, "Shape optimal design of materially nonlinear arch dams including dam-water-foundation rock interaction using an improved PSO algorithm," *Optimization and Engineering*, vol. 13, no. 1, pp. 79–100, 2012.
- [12] A. M. El-Zonkoly, "Optimal placement of multi-distributed generation units including different load models using particle swarm optimization," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 50–59, 2011.
- [13] R. J. Kuo and C. Y. Yang, "Simulation optimization using particle swarm optimization algorithm with application to assembly line design," *Applied Soft Computing Journal*, vol. 11, no. 1, pp. 605–613, 2011.
- [14] E. Pekşen, T. Yas, A. Y. Kayman, and C. Özkan, "Application of particle swarm optimization on self-potential data," *Journal of Applied Geophysics*, vol. 75, no. 2, pp. 305–318, 2011.
- [15] W.-T. Pan, "Combining PSO cluster and nonlinear mapping algorithm to perform clustering performance analysis: take the enterprise financial alarming as example," *Quality and Quantity*, vol. 45, no. 6, pp. 1291–1302, 2011.
- [16] L. Goel, D. Gupta, and V. K. Panchal, "Hybrid bio-inspired techniques for land cover feature extraction: a remote sensing perspective," *Applied Soft Computing Journal*, vol. 12, no. 2, pp. 832–849, 2012.
- [17] H. M. Jiang, C. K. Kwong, W. H. Ip, and T. C. Wong, "Modeling customer satisfaction for new product development using a PSO-based ANFIS approach," *Applied Soft Computing Journal*, vol. 12, no. 2, pp. 726–734, 2012.
- [18] J.-W. Li, Y.-C. Chang, C.-P. Chu, and C.-C. Tsai, "A self-adjusting e-course generation process for personalized learning," *Expert Systems with Applications*, vol. 39, no. 3, pp. 3223–3232, 2012.
- [19] V. Vassiliadis and G. Dounias, "Nature-inspired intelligence: a review of selected methods and applications," *International Journal on Artificial Intelligence Tools*, vol. 18, no. 4, pp. 487–516, 2009.
- [20] H. Liu and A. Abraham, "An hybrid fuzzy variable neighborhood particle swarm optimization algorithm for solving quadratic assignment problems," *Journal of Universal Computer Science*, vol. 13, no. 7, pp. 1032–1054, 2007.
- [21] J. Jie, J. Zeng, C. Han, and Q. Wang, "Knowledge-based cooperative particle swarm optimization," *Applied Mathematics and Computation*, vol. 205, no. 2, pp. 861–873, 2008.
- [22] S. M. Morkos and H. A. Kamal, "Optimal tuning of PID controller using adaptive hybrid particle swarm optimization algorithm," *International Journal of Computers, Communications and Control*, vol. 7, no. 1, pp. 101–114, 2012.
- [23] L.-Y. Chuang, S.-W. Tsai, and C.-H. Yang, "Chaotic catfish particle swarm optimization for solving global numerical optimization problems," *Applied Mathematics and Computation*, vol. 217, no. 16, pp. 6900–6916, 2011.
- [24] W.-M. Lin, H.-J. Gow, and M.-T. Tsai, "Hybridizing particle swarm optimization with signal-to-noise ratio for numerical optimization," *Expert Systems with Applications*, vol. 38, no. 11, pp. 14086–14093, 2011.
- [25] M. I. Menhas, M. Fei, L. Wang, and L. Qian, "Real/binary co-operative and co-evolving swarms based multivariable PID controller design of ball mill pulverizing system," *Energy Conversion and Management*, vol. 54, no. 1, pp. 67–80, 2012.
- [26] M. Xi, J. Sun, and W. Xu, "An improved quantum-behaved particle swarm optimization algorithm with weighted mean best position," *Applied Mathematics and Computation*, vol. 205, no. 2, pp. 751–759, 2008.
- [27] A. Nickabadi, M. M. Ebadzadeh, and R. Safabakhsh, "A novel particle swarm optimization algorithm with adaptive inertia weight," *Applied Soft Computing Journal*, vol. 11, no. 4, pp. 3658–3670, 2011.
- [28] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Perth, Australia, December 1995.

- [29] Y. Shi and R. Eberhart, "Modified particle swarm optimizer," in *Proceedings of the IEEE International Conference on Evolutionary Computation (ICEC'98)*, pp. 69–73, Anchorage, Alaska, May 1998.
- [30] A. Chatterjee and P. Siarry, "Nonlinear inertia weight variation for dynamic adaptation in particle swarm optimization," *Computers and Operations Research*, vol. 33, no. 3, pp. 859–871, 2006.
- [31] A. Ratnaweera, S. K. Halgamuge, and H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 240–255, 2004.
- [32] M. S. Arumugam, G. R. Murthy, M. V. C. Rao, and C. K. Loo, "A novel effective particle swarm optimization like algorithm via extrapolation technique," in *Proceedings of the International Conference on Intelligent and Advanced Systems (ICIAS'07)*, pp. 516–521, Kuala Lumpur, Malaysia, November 2007.
- [33] W. Fan, Z. Cui, Y. Chen, and Y. Tan, "Nonlinear time-varying stability analysis of particle swarm optimization," in *Proceedings of the International Conference on Computational Aspects of Social Networks (CASoN'10)*, pp. 3–6, Taiyuan, China, September 2010.
- [34] Y.-l. Zheng, L.-h. Ma, L.-y. Zhang, and J.-x. Qian, "Empirical study of particle swarm optimizer with an increasing inertia weight," in *Proceedings of the Congress on Evolutionary Computation (CEC'03)*, pp. 221–226, Canberra, Australia, 2003.
- [35] C. Dong, G. Wang, Z. Chen, and Z. Yu, "A method of self-adaptive inertia weight for PSO," in *Proceedings of the International Conference on Computer Science and Software Engineering (CSSE'08)*, pp. 1195–1198, Hubei, China, December 2008.
- [36] M. Lin and Z. Hua, "Improved PSO algorithm with adaptive inertia weight and mutation," in *Proceedings of the WRI World Congress on Computer Science and Information Engineering (CSIE'09)*, pp. 622–625, Los Angeles, Calif, USA, April 2009.
- [37] W. Lin, C. Jiang, and J. Qian, "The identification research of nonlinear system based on PSO with fuzzy adaptive inertia weight," in *Proceedings of the 5th World Congress on Intelligent Control and Automation (WCICA'04)*, pp. 267–271, Hangzhou, China, June 2004.
- [38] Z. Wu and J. Zhou, "A self-adaptive particle swarm optimization algorithm with individual coefficients adjustment," in *Proceedings of the International Conference on Computational Intelligence and Security (CIS'07)*, pp. 133–136, Heilong Jiang, China, December 2007.
- [39] T.-Y. Lee and C.-L. Chen, "Unit commitment with probabilistic reserve: an IPSO approach," *Energy Conversion and Management*, vol. 48, no. 2, pp. 486–493, 2007.
- [40] P. J. Angeline, "Using selection to improve particle swarm optimization," in *Proceedings of the IEEE International Conference on Evolutionary Computation Proceedings, IEEE World Congress on Computational Intelligence*, pp. 84–89, Anchorage, Alaska, USA, May 1998.
- [41] J. J. Jamian, M. W. Mustafa, H. Mokhlis, and M. N. Abdullah, "Comparative study on distributed generator sizing using three types of particle swarm optimization," in *Proceedings of the 3rd International Conference on Intelligent Systems Modelling and Simulation (ISMS'12)*, pp. 131–136, Sabah, Malaysia, February 2012.