

# Swarmed Grey Wolf Optimizer



Sumita Gulati<sup>1,2,\*</sup> and Ashok Pal<sup>1</sup>

<sup>1</sup>Department of Mathematics, Chandigarh University, Gharuan (Mohali) Punjab, India; <sup>2</sup>Department of Mathematics, S. A. Jain College, Ambala City-134003, Haryana

**Abstract: Background:** The Particle Swarm Optimization (PSO) algorithm is amongst the utmost favourable optimization algorithms often employed in hybrid procedures by the researchers considering simplicity, smaller count of parameters involved, convergence speed, and capability of searching global optima. The PSO algorithm acquires memory, and the collaborative swarm interactions enhances the search procedure. The high exploitation ability of PSO, which intends to locate the best solution within a limited region of the search domain, gives PSO an edge over other optimization algorithms. Whereas, low exploration ability results in a lack of assurance of proper sampling of the search domain and thus enhances the chances of rejecting a domain containing high quality solutions. Perfect harmony between exploration and exploitation abilities in the course of selection of the best solution is needed. High exploitation capacity makes PSO trapped in local minima when its initial location is far off from the global minima.

**Objectives:** The intent of this study is to reform this drawback of PSO of getting trapped in local minima. To upgrade the potential of Particle Swarm Optimization (PSO) to exploit and prevent PSO from getting trapped in local minima, we require an algorithm with a positive acceptable exploration capacity.

**Methods:** We utilized the recently developed metaheuristic Grey Wolf Optimizer (GWO), emulating the seeking and hunting techniques of Grey wolves for this purpose. In our way, the GWO has been utilized to assist PSO in a manner to unite their strengths and lessen their weaknesses. The proposed hybrid has two driving parameters to adjust and assign the preference to PSO or GWO.

**Results:** To test the activity of the proposed hybrid, it has been examined in comparison with the PSO and GWO methods. For this, eleven benchmark functions involving different unimodal and multimodal functions have been taken. The PSO, GWO, and SGWO pseudo codes were coded in visual basic. All the functional parameters of PSO and GWO were chosen as:  $w = 0.7$ ,  $c_1 = c_2 = 2$ , population size = 30, number of iterations = 30. Experiments were redone 25 times for each of the methods and for each benchmark function. The methods were compared with regard to their best and worst values besides their average values and standard deviations. The obtained results revealed that in terms of average values and standard deviations, our hybrid SGWO outperformed both PSO and GWO notably.

**Conclusion:** The outcomes of the experiments reveal that the proposed hybrid is better in comparison to both PSO and GWO in the searchability. Though the SGWO algorithm refines result quality, the computational complexity also gets elevated. Thus, lowering the computational complexity would be another issue of future work. Moreover, we will apply the proposed hybrid in the field of water quality estimation and prediction.

**Keywords:** Particle swarm optimizer (PSO), grey wolf optimizer (GWO), exploitation, exploration, swarmed grey wolf optimizer (SGWO), unimodal function.

## 1. INTRODUCTION

Optimization is an eminent method in making opinions and in investigating physical systems. A variety of optimization problems arise in almost all fields of human acts. Mathematically, optimization is a practice of finding the best

solution out of the set of feasible solutions. In the last few years, many nature inspired optimization algorithms have been evolved. These include Particle Swarm Optimization (PSO) Algorithm, Genetic Algorithm, Evolutionary Algorithm, Ant Colony Optimization Algorithm, Firefly Algorithm, etc. Attributes like simple application, computing efficiency, derivation free mechanism, flexibility, local optimal avoidance, and global search make nature inspired algorithms an admirable choice of researchers. Existing nature inspired optimization algorithms are proficient for sev-

\*Address correspondence to this author at the Department of Mathematics, Chandigarh University, Gharuan (Mohali) Punjab, India; E-mail: [sumitabansal2223@gmail.com](mailto:sumitabansal2223@gmail.com)

## ARTICLE HISTORY

Received: September 20, 2021  
Revised: November 02, 2021  
Accepted: November 29, 2021

DOI:  
10.2174/2666782701666220304140720



CrossMark

eral real life and test optimization problems. However, because of the no free lunch theorem [1], there exists no universal algorithm that can be considered most appropriate and finest of all for any of the optimization problem. Besides developing new nature inspired algorithms, another prevailing concept is hybridising distinct algorithms in order to merge their strengths, minimizing weaknesses, and thus, enhancing the search process.

The PSO is amongst the utmost favourable optimization algorithms often employed in hybrid procedures by the researchers considering simplicity, smaller count of parameters involved, convergence speed, and capability of searching global optima. On various problems of different domains, PSO works successfully and this makes us select the PSO as one of the algorithms for our hybrid to propose.

The PSO algorithm acquires memory, and the collaborative swarm interactions enhances the search procedure. The high exploitation ability of the PSO, which intends to locate the best solution within a limited region of the search domain, gives the PSO an edge over other optimization algorithms. Whereas, low exploration ability results in a lack of assurance of proper sampling of the search domain and thus enhances the chances of rejecting a domain containing high quality solutions. Perfect harmony between exploration and exploitation abilities in the course of selection of the best solution is needed. High exploitation capacity makes the PSO get trapped in local minima when its initial location is far off from the global minima [2-5]. The intent of this study is to reform this drawback of the PSO of getting trapped in local minima.

In this direction, we require an algorithm with a positive acceptable exploration capacity. We utilized the recently developed metaheuristic Grey Wolf Optimizer (GWO), emulating the seeking and hunting techniques of grey wolves for this purpose. In our way, the GWO has been utilized to assist the PSO in a manner to unite their strengths and lessen their weaknesses. The proposed hybrid has two driving parameters to adjust and assign the preference to PSO or GWO. This optimization algorithm is simple to use and quite efficient, with less randomness and varying individuals in assigning local and global search processes. In the literature, many existing hybrids utilize other metaheuristics with the PSO for this downside. Zhang *et al.* [6] proposed a hybrid variant by connecting the PSO algorithm with the Back Propagation (BP) algorithm. This variant makes use of dominant global search capacity and persuasive local search capacity of the PSO and the BP algorithms, respectively. The hybrid was proposed to train the weights of feed forward neural networks. In this proposed hybrid, a unique selection strategy of inertial weight was used in which at an initial stage, the inertial weight reduces rapidly to attain global minima and then the inertial weight was reduced smoothly around global optima in order to acquire higher accuracy. Moreover, a heuristic way was utilized to transit the search procedures amongst the involved methods. The authors demonstrated that their hybrid variant outperformed the BP and the adaptive PSO algorithm regarding the quality of solution, convergence speed, and stability.

Combining the strengths of Nelder - Mead Simplex Method variant and the PSO, another hybrid variant was put

forward by Ouyang *et al.* [7] to solve the non-linear system of equations. Authors claimed that their proposed hybrid variant overcomes the strain of selection of initial guess of the Simplex Method and imprecision of the PSO because of easily getting captured into local minima.

Another hybrid was presented by Mirjalili and Hashim's [8] by combining the PSO with the Gravitational Search Algorithm (GSA). The main intention was to assimilate the exploitation capacity of the PSO and the exploration capacity of the GSA. The execution of this combination was verified on some standard functions and compared to the PSO and GSA. The hybrid proved to possess a finer potential to get away from local optima along with rapid convergence in functional optimization.

A newly hybrid PSO variant was developed by Yu *et al.* [9] by merging the modified velocity model with Space Transformation Search (STS). The proposed model involves monitoring fitness values of pbest and gbest. In the case of pbest, gbest shows no improvements, then being considered as getting trapped in local optima, they are given some disturbances to break away the local optima. The variant was verified on eight classical problems and compared with the standard PSO and STS-PSO, and results revealed that the suggested hybrid variant performed well in determining solutions of both unimodal and multi-modal problems.

Esmine *et al.* [10] suggested another hybrid variant of the PSO with the Genetic Algorithm (GA). Both algorithms were merged with the GA mutation technique. The suggested hybrid acquires an automatic balance maintaining potential between global and local search. The hybrid was validated on a variety of benchmark functions compared to the standard PSO. It performed significantly well with regard to the quality of solution, firmness of solution, convergence speed, and attainment of global optima.

A multi-objective model to minimize water shortages and maximize hydropower generation of the Tao River in China, utilizing an adjustable PSO-GA hybrid algorithm, was conducted by Chang *et al.* [11]. This hybrid merges the potencies of both PSO and GA to balance the process of sharing good knowledge and natural selection, enabling robustness and effective search procedure. The hybrid was validated by comparing it with the involved techniques. The results showed this hybrid as a promising algorithm with rapid convergence speed. The authors also mentioned that this method has the potential for significant application in large water resource systems.

In another study, Yu *et al.* [12] recommended a new hybrid variant of the PSO by combining it with the Differential Evolution (DE). In this variant, the PSO and the DE were mixed with the help of a symmetric parameter. In this hybrid, when the population gets clustered around a local optima, the current population conducts adaptive mutation. This hybrid, along with maintaining diversity, relishes the advantages of both of the involved algorithms. The conduct of this hybrid variant was then verified on a number of benchmark test problems and compared with the PSO, the DE, and their variants. The authors have proved that this hybrid variant outperformed all of them regarding the quality of solution, frequency of quality solutions, and works effectively.

Abd-Elazim and Ali [13] developed a novel hybrid variant for tuning static var compensator for multi-machine power system by merging the PSO with the Bacterial Foraging Optimization Algorithm (BFOA) named Bacterial Swarm Optimizer (BSO). In this proposed variant, the personal and global best locations of the PSO algorithm orient the search directions of each tumble behavioural bacterium. The hybrid BSO outperformed both the standard PSO and BFOA. Simulation results validate the proposed tuning approach for static var compensator in comparison to tuning with standard PSO and BFOA. Furthermore, results demonstrate the effectiveness of the suggested technique in improving the stability of power systems varying over a varied range of loading conditions.

With the aim of discovering the optimum solution, Kumar and Vidhyarthi [14] developed another hybrid of the PSO, blending it with the GA. They started the search procedure with the PSO algorithm in order to reduce the search domain and then progressed with the GA method. In the suggested hybrid, the PSO provides diversification and the GA enhances intensification. The proposed PSO-GA blend was validated for task scheduling of basic linear algebra problems, namely LU decomposition and Gauss-Jordan elimination, and also on other heuristics with known solutions. The results revealed that the proposed hybrid performed quite effectively for scheduling problems.

Further, another penalty guided PSO-GA hybrid was proposed by Garg [15] by fusing the strengths of both for constrained optimization problems. In the PSO, the whole population continued to take part in each iteration, whereas the GA intends to transfer only satisfactory individuals of the population to the succeeding generation. In this hybrid, two algorithms were combined in such a way that the GA technique formulated the next generation to be used by the PSO at each iteration by executing crossover, mutation, and selection. To validate the proposed technique, an engineering design problem was investigated. The proposed algorithm was compared with other evolutionary algorithms and the experimental results revealed superiority of the proposed technique and proved the approach more potent in engineering problems.

Ghasemi *et al.* [16] proposed a novel approach for finding peak particle velocity in open pit mines due to bench blasting. In this approach, the Artificial Neuro Fuzzy Inference System (ANFIS) is merged in a way that they, with the help of the PSO algorithm, optimize the ANFIS arrangement. To validate the effectiveness of the proposed approach, a model based on the Support Vector Regression was developed. The obtained results revealed that the proposed ANFIS-PSO algorithm provides better results in this way; they reduced the gap between the results secured with the ANFIS training and the real results.

Another algorithm was proposed by Javidrad and Nazari [17] by uniting the PSO with the Simulated Annealing (SA). They adapted a turn – based transformation approach to merge them. The process of optimization initiates with the PSO process and transformed to the SA process at the time when the PSO was not adept at getting a superior solution than its previous ones. In a similar way, the SA algorithm continued the optimization process and the turn transfer to

the PSO when the SA could not find a better fitted solution. This transformation between the two algorithms continued till a stopping condition was achieved or a desired count of iterations were performed. The proposed PSO-SA algorithm showed better performance in terms of solutions to the problems considered compared to other evolutionary techniques. Moreover, the authors exhibit their algorithm as more reliable and effective for various optimization problems.

In another study, a three phase combination of the PSO and the GA methods was developed by Ali and Tawhid [18]. In the first phase, a stable fitness value was calculated by the PSO algorithm. The second phase involved the division of the population into sub-population using the GA crossover operation and thus increasing the diversity. Moreover, the GA's mutation operation hampered trapping to local minima. The optimization process was shifted back to the PSO algorithm again in the final stage, and the PSO was executed till the stopping condition was met or the maximum count of iterations was attained. The algorithm was compared with the standard PSO for solving complex global optimization problems and with nine benchmark functions for validating the expertise in determining molecule's potential energy function. The experimental results revealed that the suggested methodology is promising, efficient, and rapid in getting global minimum for molecular energy function.

Hasanipanah *et al.* [19] combined the Support Vector Regression with the PSO to anticipate air-over pressure generated by mine-blasting. The PSO method was used to compute hyper-parameters of each of the kernels and the Support Vector Regression procedure was tested with three distinct kernels. To check the accuracy of the suggested approach, multi-linear regression was utilized. The outcomes of the study revealed the reliability of the suggested algorithm in training the Support Vector Regression model.

To enhance the network lifetime of wireless sensor networks, a hybrid combining the PSO algorithm with the Ant Colony Algorithm (ACA) was developed by Kaur and Mahajan [20]. In this hybrid, the ACA was used first to evolve the population and then optimization was progressed with the PSO algorithm. At first, the clusters were constructed depending upon remaining energy, and then, the hybrid dependent clustering was carried out. The results of the study demonstrated enhanced network lifetime in comparison to the other techniques.

Furthermore, analogous to our proposed hybrid, other hybrid variants blend strengths of the PSO algorithm with the GWO algorithm in different ways. Chopra *et al.* [21] proposed a hybrid in which the PSO and the GWO algorithms were employed in succession. The population attained by one algorithm was utilized by the other in its upcoming iteration. The motive of the hybrid is to utilize the strengths of both methods. Kamboj [22] also presented a similar hybrid in which both the PSO and the GWO run in succession. Unlike Chopra's hybrid in which the whole of the population was handed over from the first algorithm to the second at each iteration, in this, the whole population was renewed by one algorithm with the best ones of the other algorithm secured in the previous iteration. Another hybrid was proposed by Singh and Singh [23] in which instead of running the algorithms in succession, both algorithms

work in parallel using the mixture of algorithm commanding equations. Şenel *et al.* [24] conducted another study in which a PSO-GWO hybrid variant was developed with the optimization control process under the PSO algorithm. In this approach, few individuals of the PSO algorithm were selected and replaced with the best individuals of the GWO algorithm. The GWO algorithm runs for a limited population and for a limited number of iterations in order to create partially better individuals. These partially improved individuals possess the potential of avoiding getting trapped in local minima. With intact PSO algorithm stability, this hybrid takes the support of the GWO for exploration. Teng *et al.* [25] proposed another hybrid variant combining the GWO with the PSO. To initiate the individual's location, Tent Chaotic sequence was utilized to increase the diversification of the wolf pack, and nonlinearity of the control parameter was utilized for enhancing local and global exploration along with improving the convergence speed of the algorithm. The notion of the PSO utilizes the pre-eminent position of the individual and prevents the wolf pack from falling into local optima.

In a similar direction, we proposed a hybrid variant of the PSO and the GWO. The main features of our study are: (1) In our hybrid variant, two driving parameters are employed to adjust the control of the optimization process between the PSO and the GWO. (2) The aim of this approach is to merge the strengths of the PSO and the GWO in a way that improves local search ability and accelerates algorithm optimization. (3) Non-linear control parameter strategy is adopted to have coordination between the exploration and exploitation ability.

The remaining paper is organized as follows:

Section 2 illustrates the engaged optimization algorithms, depicts the proposed hybrid variant and illustrates the benchmark functions used to assess our approach. Section 3 represents the experimental outcomes, including discussion. Section 4 concludes the paper and outlines the subsequent scope.

## 2. MATERIALS AND METHODS

### 2.1. Optimization Algorithms

#### 2.1.1. Particle Swarm Optimization (PSO) Algorithm

The PSO algorithm was initially suggested by Kennedy and Eberhart [26] in the year 1995. The PSO is considered an initial optimization algorithm to be studied under Swarm Intelligence. The PSO algorithm is a population depending search algorithm and was originally invented to imitate the social attitude and conduct of birds within a swarm. In basic PSO, individuals in a swarm of particles displace following a very clear attitude; to imitate the achievements of neighboring entities and their self-achievements. The particles glide through a hyper-dimensional search domain, and the position of every particle is modified and regulated in accordance with its own experience and experiences of its neighbors. To put it another way, particles keep track of their personal best position (known as *pbest*) and the best position in their immediate vicinity (known as *gbest*). The course of optimization is derived by the velocity vector. The velocity vector reflects the experimental information (which

is recognized as a cognitive constituent) and socially exchanged information (which is recognized as the social constituent).

For *gbest* PSO, the particle velocity is evaluated by

$$v_{ij}(t+1) = v_{ij}(t) + c_1 \text{rand}() (pbest_{ij} - x_{ij}(t)) + c_2 \text{Rand}() (gbest_{ij} - x_{ij}(t)) \quad (1)$$

where,  $v_{ij}(t)$  describes velocity of  $i$ th particle at time  $t$  in the dimension  $j=1, 2, \dots, n_x$ ;  $c_1, c_2$  are positive acceleration constants utilized to scale the extent of cognitive and social constituents respectively;  $\text{rand}()$ ,  $\text{Rand}()$  are arbitrary values within the range  $[0, 1]$ . These arbitrary values add hypothetical nature to the algorithm.

The position of the particle is then restored by the equation

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \quad (2)$$

Here,  $v_{ij}$  is usually bounded in some range.

There are a large number of variants of the standard basic PSO algorithm, but the most effective and notable is probably the concept of inertia weight proposed by Shi and Eberhart [27], in which the term  $v_{ij}(t)$  is retrieved by  $wv_{ij}(t)$  where the inertia function  $w$  takes value lying between 0 and 1. The value of  $w$  ensures convergence and settles the process of exploration and exploitation. The smaller values of  $w$  support local search, whereas larger values support global search. The introduction of inertia also excludes the need for re-setting of  $v_{\max}$  each time during the execution of the PSO algorithm. As the iteration progresses, the particle system converges towards global optima.

#### 2.1.2. Grey Wolf Optimization (GWO) Algorithm

Mirjalili *et al.* [28] suggested the GWO algorithm emulating the seeking and hunting techniques of grey wolves. Known as apex predators, grey wolves are part of the Canidae family. They mostly favor living in packs of five to twelve wolves on average. The pack is split into four different groups, namely alpha ( $\alpha$ ), beta ( $\beta$ ), delta ( $\delta$ ), and omega ( $\omega$ ). They have a very rigid social grouping, with each wolf having a role to play in the pack.

The  $\alpha$  are the leaders and are generally responsible for making decisions about habitat, sleeping, time to wake, hunting, *etc.*, in the pack. Their orders should be obeyed in the pack. The  $\beta$  are the second best and are the assistant wolves that support and assist  $\alpha$  wolves in making decisions and other group activities. They direct and instruct the other lower level wolves along with reinforcing the instructions of  $\alpha$  wolves. Playing the role of space goat, the third level  $\omega$  wolves are the lowest ranking grey wolves. They must constantly acknowledge all other leading wolves. In case a wolf is not  $\alpha$ ,  $\beta$ , or  $\omega$  wolf, then he/she is known as an assistant  $\delta$  wolf. They assist  $\alpha$  and  $\beta$  wolves, but they dictate  $\omega$  wolves.

Besides social grouping, the technique of hunting in a group is another aspect of this family of grey wolves. In GWO, their hunting technique and social grouping have been mathematically modelled to perform optimization.

##### 2.1.2.1. Social Grouping

The fittest solution is contemplated as  $\alpha$ , whereas the second and third competent solutions are expressed as  $\beta$  and

$\delta$ , respectively. The rest of the candidate solutions are regarded as  $\omega$ . The hunting is governed by  $\alpha$ ,  $\beta$ ,  $\delta$  and  $\omega$  wolves assist them.

### 2.1.2.2. Encircling Prey

This behaviour of each agent of the pack is calculated by the succeeding mathematical equations:

$$D = |c \cdot x_p(t) - x(t)| \quad (3)$$

$$x(t + 1) = x_p(t) - A \cdot D \quad (4)$$

Here  $t$  indicates the current iteration,  $x_p$  and  $x$  are the position vectors of the prey and grey wolf respectively,  $A$  and  $c$  are coefficient vectors and are determined as follows:

$$A = 2a \cdot r_1 - a \quad (5)$$

and

$$c = 2r_2 \quad (6)$$

where  $a$  decreases from 2 to 0 over the progress of iterations and  $r_1, r_2$  are arbitrary vectors in  $[0, 1]$ .

### 2.1.2.3. Hunting

The hunting is directed by  $\alpha$ , whereas  $\beta$  and  $\delta$  participate occasionally in hunting. The hunting is simulated mathematically as follows:

$$D_\alpha = |c_1 \cdot x_\alpha - x|$$

$$D_\beta = |c_2 \cdot x_\beta - x|$$

$$D_\delta = |c_3 \cdot x_\delta - x|$$

$$\text{and } x_1 = x_\alpha - A_1 \cdot D_\alpha$$

$$x_2 = x_\beta - A_2 \cdot D_\beta$$

$$x_3 = x_\delta - A_3 \cdot D_\delta$$

$$\text{and } x(t + 1) = \frac{x_1 + x_2 + x_3}{3}$$

### 2.1.2.4. Searching Prey (Exploration) and Attacking Prey (Exploitation)

$A$  is a random value in the interval  $[-2a, 2a]$ . For values of  $A$ , with  $|A| < 1$ , the wolves are directed to attack the prey, and when  $|A| \geq 1$ , the wolves are directed to diverge from prey. They diverge from one another for searching prey and converge for attacking prey. With the progress of iterations, the value of  $A$  decreases and half of the iterations are dedicated to exploration with  $|A| \geq 1$  and the rest half to exploitation with  $|A| < 1$ .

Another component favouring exploration is  $c$ , which has arbitrary values in  $[0, 2]$ . Value of  $c$  is not linearly reduced in contrast to  $A$  but has random values to facilitate exploration throughout the process. The GWO has two parameters  $a$  and  $c$ , to adjust.

## 2.2. Swarmed Grey Wolf Optimization (SGWO) Algorithm

We developed our hybrid Swarmed Grey Wolf Optimizer (SGWO) algorithm without altering the general standard operations of both PSO and GWO algorithms. The PSO algorithm can attain desired outcomes for most of the opti-

mization problems. But getting trapped into a local minima is a major drawback of this algorithm. In our proposed SGWO algorithm, the GWO is supporting the PSO in a way that merges the strengths of both and reduces the chances of the PSO getting trapped into local minima. In this approach, an adjustable technique is utilized to adjust the control amongst the two algorithms used in hybrid. To allocate a preference amongst the PSO or the GWO, two driving parameters have been designed. These parameters are  $k_1$  and  $k_2$ , with  $k_1$  as influence term for the PSO and  $k_2$  for the GWO. The algorithm runs in the ratio  $k_1:k_2$  such that at each iteration,  $k_1 + k_2 = 1$ . When influence term  $k_1 = 0$  and  $k_2 = 1$  then PSO has no impact on the population, and similarly when  $k_1 = 1$  and  $k_2 = 0$ , then GWO has no impact on the population. For intermediate values of influence terms  $k_1$  and  $k_2$ , the hybrid algorithm executes the basic position and velocity update equations with the top (population size  $\times k_1$ ) individuals moving to the next iteration through the PSO algorithm while rest of the (population size  $\times k_2$ ) individuals moving to next iteration through the GWO algorithm. Here we are taking population size = maximum number of iterations.

If  $T_{max}$  is the maximum number of iterations, then

$$k_1 = 1 - \frac{n}{T_{max}}, \text{ where } n = 0, 1, 2, \dots, T_{max} - 1 \quad (7)$$

A non-linear control parameter strategy is employed in this hybrid proposed by Teng *et al.* [25]. This non-linear parameter enhances the exploration and exploitation capabilities of the GWO. We used the value of the control parameter as

$$a = 2 \left( 1 - \left( \frac{n}{T_{max}} \right)^2 \right) \quad (8)$$

The detailed process of SGWO is described as follows, and the relevant flowchart is shown in Fig. (1).

**Step 1: Initialization.** The population is initialized and maximum number of iterations,  $T_{max}$  is taken equal to population size. The parameters  $c_1, c_2$ , and  $w$ , are initialized for the PSO and for the GWO,  $A$ , and  $c$  given by the equations (5) and (6), in which  $a$  is given by equation (8).

**Step 2: Calculate fitness value.** Select the appropriate benchmark function as an objective function and calculate the fitness value of the whole population.

**Step 3: Run PSO.** Select population size  $X$   $k_1$  individuals where  $k_1$  is given by the equation (7) and then update velocity and position of those individuals by the equations (1) and (2), respectively.

**Step 4: Run GWO.** Select population size  $X$   $k_2$  individuals where  $k_2 = 1 - k_1$  and then update the position of those individuals by the equations (3) and (4).

**Step 5: Test the stopping condition.** Determine whether a maximum number of iterations,  $T_{max}$  is reached, otherwise return to Step 2 and continue searching.

### 2.2.1. Benchmark Functions

To test the activity of the proposed hybrid, it has been examined in comparison with the PSO and the GWO methods.

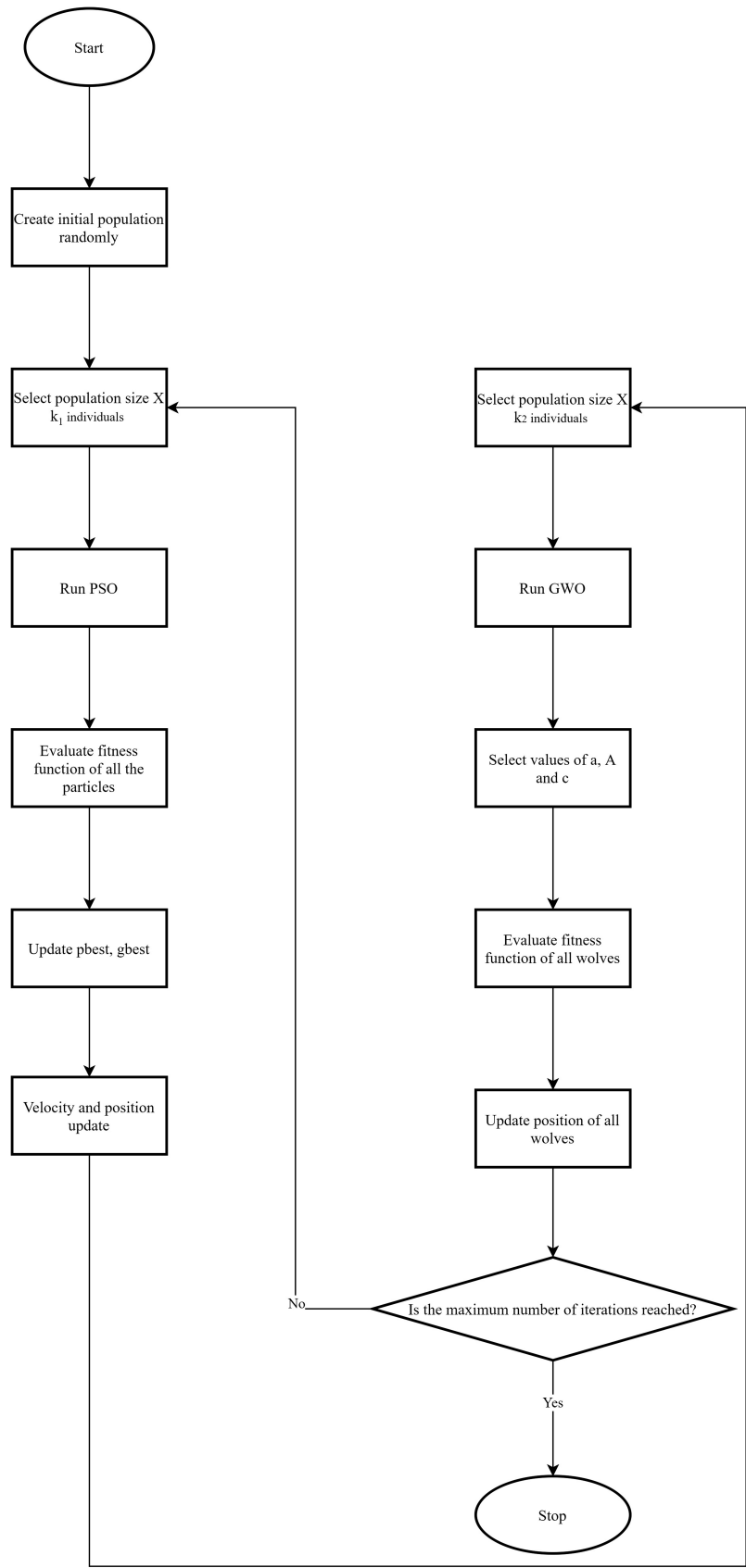


Fig. (1). Flow chart of the proposed SGWO algorithm.

For this, eleven benchmark functions involving different unimodal and multimodal functions have been taken. This set of benchmark functions involves two kinds of functions. First kind is of scalable functions (F1-F6) in which the dimension can be increased or decreased by choice, and mostly complexity increases with an increase in dimension. Second kind involves functions with fixed dimensions (F7-F11). Details of these benchmark functions are given in Table 1.

### 3. RESULTS AND DISCUSSION

The PSO, GWO, and SGWO pseudo codes were coded in visual basic. In all the functions, parameters of PSO and GWO were chosen as [23, 24]:  $w = 0.7$ ,  $c_1 = c_2 = 2$ , population size = 30, number of iterations = 30.

Experiments were redone 25 times for each of the methods and for each of the benchmark functions. The methods were compared with regard to their best, worst, and average values as well as their mean absolute deviation (MAD) and standard deviation (Std. Dev.). The numerical and statistical

results are illustrated in Table 2. The graphs of the values obtained in 25 runs for PSO, GWO, and SGWO for each of the test functions are shown in Fig. (2).

The obtained results revealed that in terms of average values, Std. Dev. and MAD, our hybrid SGWO outperformed both PSO and GWO notably, with only F<sub>2</sub> and F<sub>3</sub> supplying better results in terms of the average for the GWO, in terms of Std. Dev. and MAD, only F<sub>2</sub> supplied better results for the GWO. Also with regard to best value, the GWO gives better results than our hybrid for functions F<sub>2</sub>, F<sub>3</sub>, F<sub>4</sub>, and F<sub>5</sub> only. However, our method performed superior to PSO in all respects. So, we can conclude that SGWO worked fairly well in comparison to both PSO and GWO. This certainly marks the success of our hybridization technique. The results indicate that our hybrid is directed towards improving the downside of the PSO of getting trapped in local minima by merging it with the GWO in this adjustable manner.

**Table 1. Benchmark functions.**

Function Name	Function Expression	Type	Range	f <sub>min</sub>
Sphere function	$F_1(x) = \sum_{i=1}^n x_i^2$	Unimodal	[-100, 100]	0
Schwefel 2.22 function	$F_2(x) = \sum_{i=1}^n  x_i  + \prod_{i=1}^n  x_i $	Unimodal	[-10, 10]	0
Schwefel 1.2 function	$F_3(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	Unimodal	[-100, 100]	0
Schwefel 2.21 function	$F_4(x) = \max\{ x_i ; 1 \leq i \leq n\}$	Unimodal	[-100, 100]	0
Step 2 function	$F_5(x) = \sum_{i=1}^n (x_i + 0.5)^2$	Unimodal	[-100, 100]	0
Quartic Noise	$F_6(x) = \sum_{i=1}^n ix_i^4 + \text{rand}[0, 1)$	Unimodal	[-1.28, 1.28]	0
Schwefel 2.6 function	$F_7(x) = \max\{ x_1 + 2x_2 - 7 ,  2x_1 + x_2 - 5 \}$	Unimodal	[-100, 100]	0
Matyas function	$F_8(x) = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2$	Unimodal	[-10, 10]	0
Booth function	$F_9(x) = (x_1 + 2x_2 - 7)^2 + (2x_1 + x_2 - 5)^2$	Unimodal	[-10, 10]	0
Himmelblare function	$F_{10}(x) = (x_1^2 + x_2 - 11)^2 + (x_1 + x_2^2 - 7)^2$	Multimodal	[-5, 5]	0
Camel function Three hump	$F_{11}(x) = 2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6} + x_1x_2 + x_2^2$	Multimodal	[-5, 5]	0

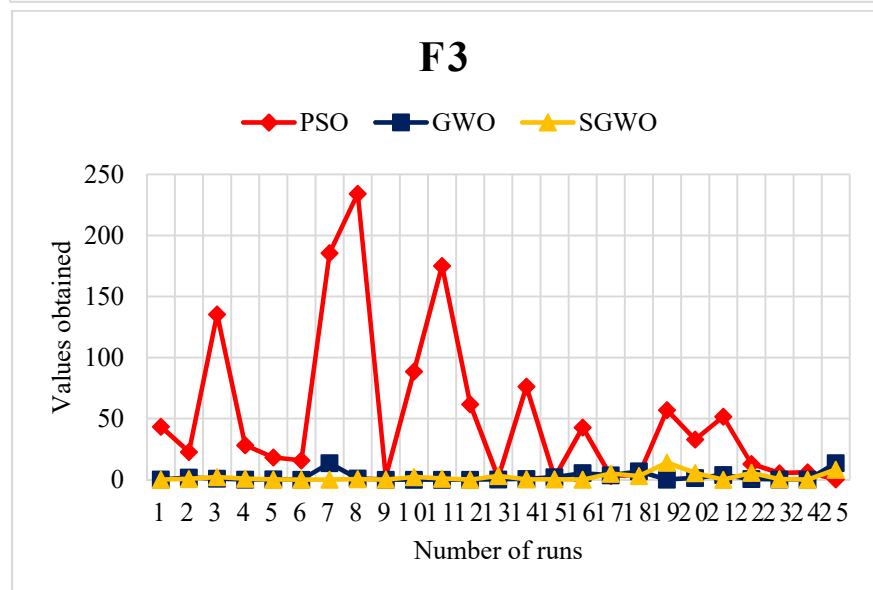
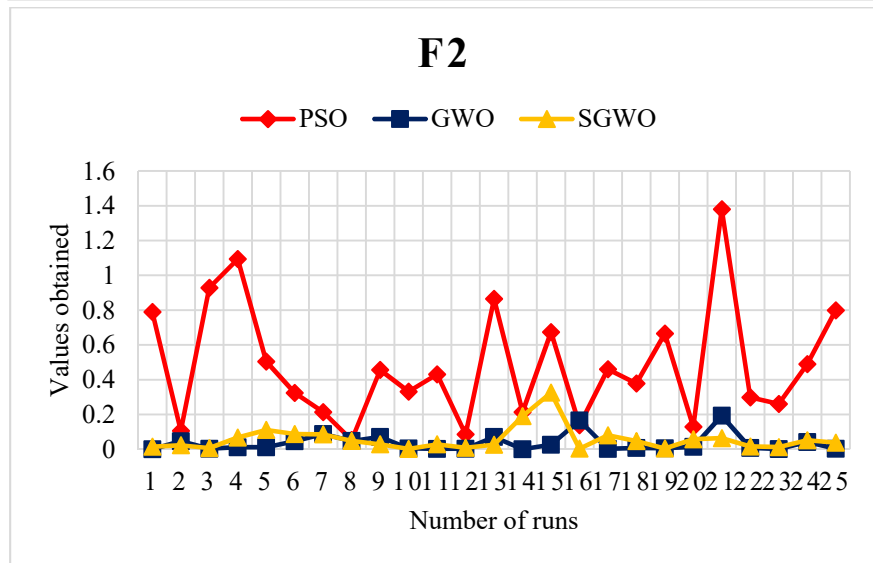
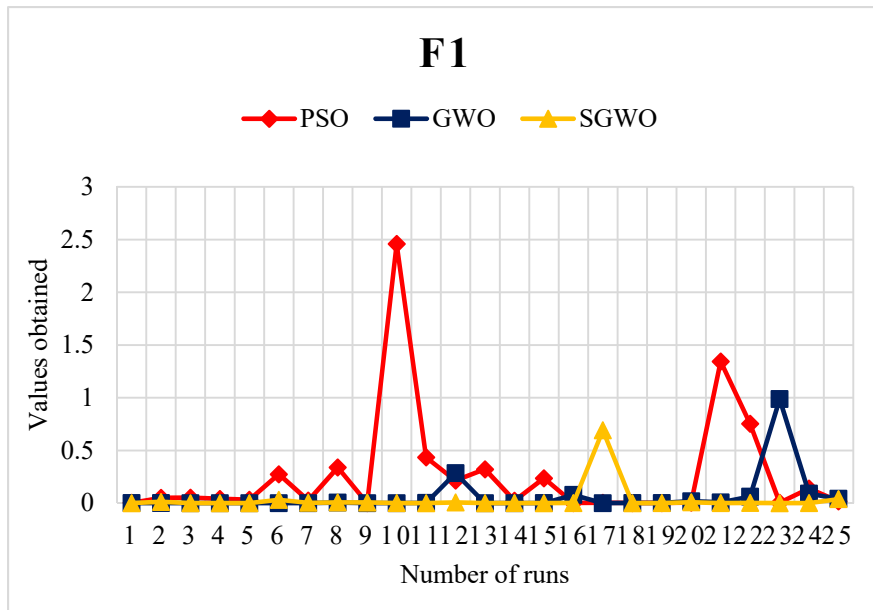
**Table 2. PSO, GWO, and SGWO numerical and statistical results of benchmark functions.**

Function Name	Metrics	PSO	GWO	SGWO
F1	Best	0.0003	0	0
	Worst	2.459	0.9865	0.691
	Average	0.2711	0.0632	0.0324
	Std. Dev.	0.5365	0.1974	0.1375
	MAD	0.3223	0.0657	0.0531
F2	Best	0.0529	0	0.002
	Worst	1.3802	0.193	0.3254
	Average	0.4825	0.0348	0.0569
	Std. Dev.	0.3351	0.0496	0.0689
	MAD	0.2689	0.0365	0.0444

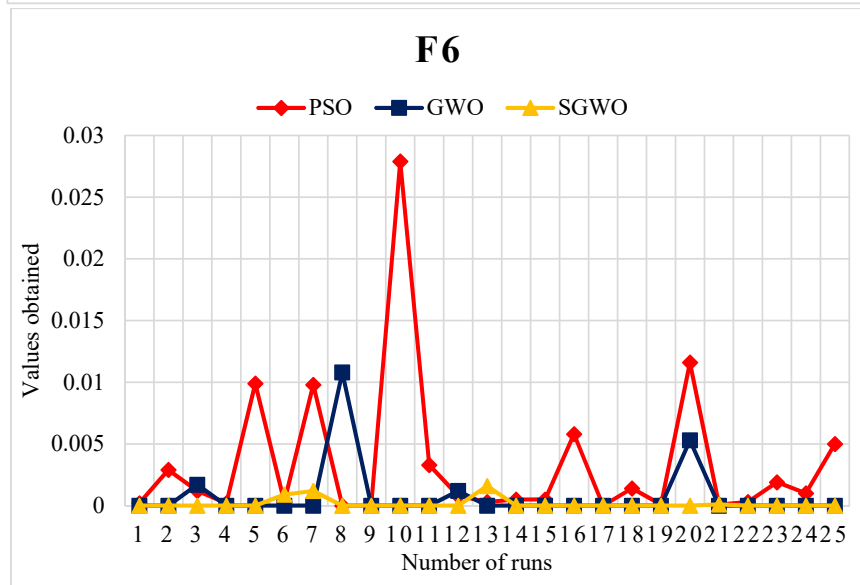
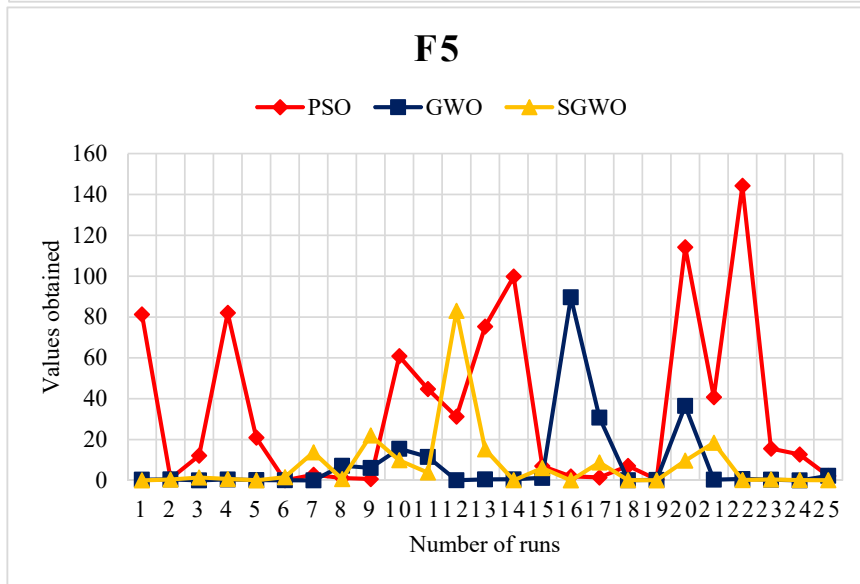
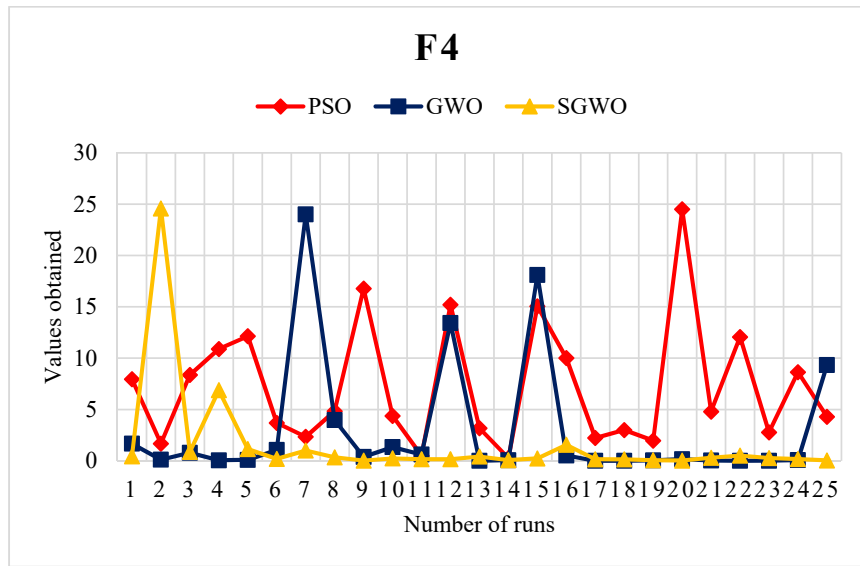
(Table 2) Contd....

Function Name	Metrics	PSO	GWO	SGWO
F3	Best	0.3339	0.0003	0.0018
	Worst	234.2435	13.6236	14.0324
	Average	52.3065	1.7892	2.1245
	Std. Dev.	63.3525	3.787	3.2862
	MAD	47.6586	2.6710	2.3864
F4	Best	0.2592	0.0001	0.0176
	Worst	24.4979	24.0077	24.5714
	Average	7.2613	3.0324	1.5937
	Std. Dev.	5.9441	6.2112	4.8751
	MAD	4.9395	4.2943	2.2608
F5	Best	0.0919	0.0027	0.0165
	Worst	144.2744	89.6492	82.9846
	Average	34.4655	8.2077	7.8621
	Std. Dev.	41.2663	19.0782	16.6252
	MAD	34.6430	11.4289	9.4222
F6	Best	0.000	0.000	0.000
	Worst	0.0279	0.0108	0.0016
	Average	0.0034	0.0008	0.0001
	Std. Dev.	0.006	0.0023	0.0004
	MAD	0.0040	0.0013	0.0003
F7	Best	1.9422	0.5759	0.0896
	Worst	491.0176	75.2575	5.1885
	Average	54.9624	18.0954	1.8054
	Std. Dev.	101.0054	21.6529	1.3951
	MAD	60.3061	15.8158	1.1148
F8	Best	0.0004	0.0000	0.0000
	Worst	2.0618	0.0789	0.0514
	Average	0.3034	0.0092	0.0045
	Std. Dev.	0.5683	0.0183	0.0104
	MAD	0.3973	0.0123	0.0058
F9	Best	0.0915	3.9282	0.0053
	Worst	3705.305	214.8598	6.3827
	Average	188.7342	55.5385	0.9144
	Std. Dev.	721.9475	43.1752	1.4488
	MAD	290.9268	31.9983	0.9970
F10	Best	0.0102	11.2868	0.0045
	Worst	313.5382	178.3368	4.2725
	Average	53.9882	150.3835	1.2291
	Std. Dev.	92.1818	39.2054	1.2839
	MAD	71.4062	28.9306	1.0398
F11	Best	0.0065	0.0000	0.0000
	Worst	9.6538	1.5922	0.3353
	Average	0.6986	0.1919	0.0286
	Std. Dev.	1.8986	0.4359	0.0685
	MAD	0.9102	0.2765	0.0390

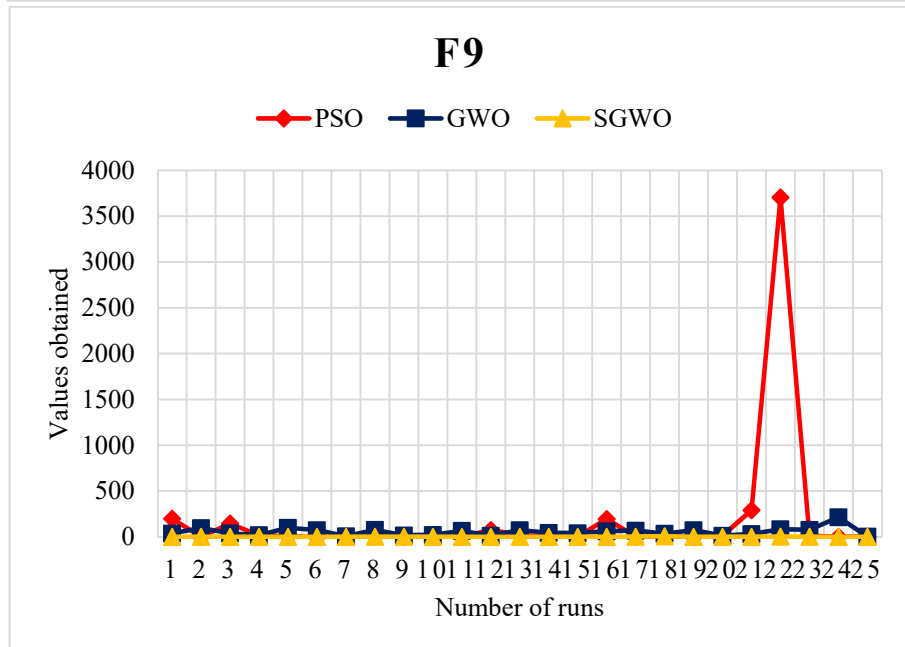
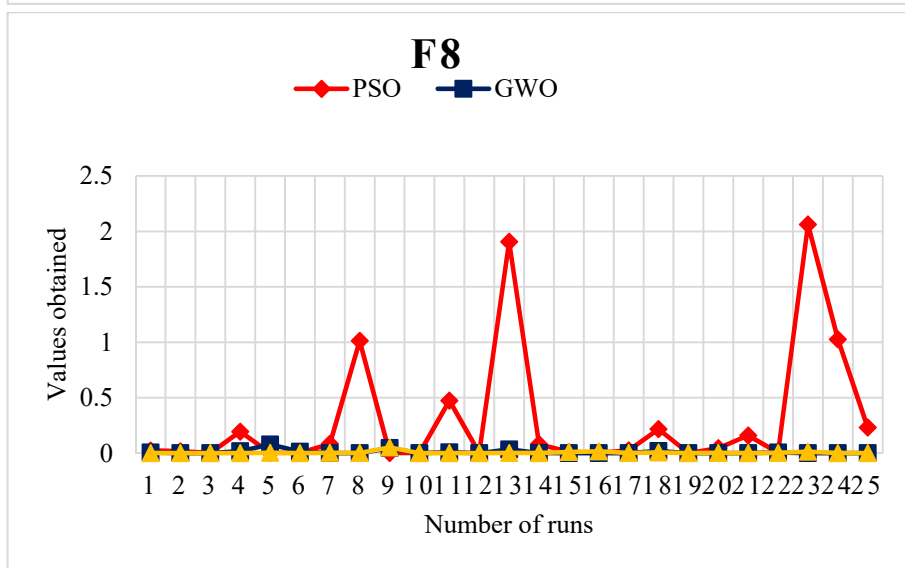
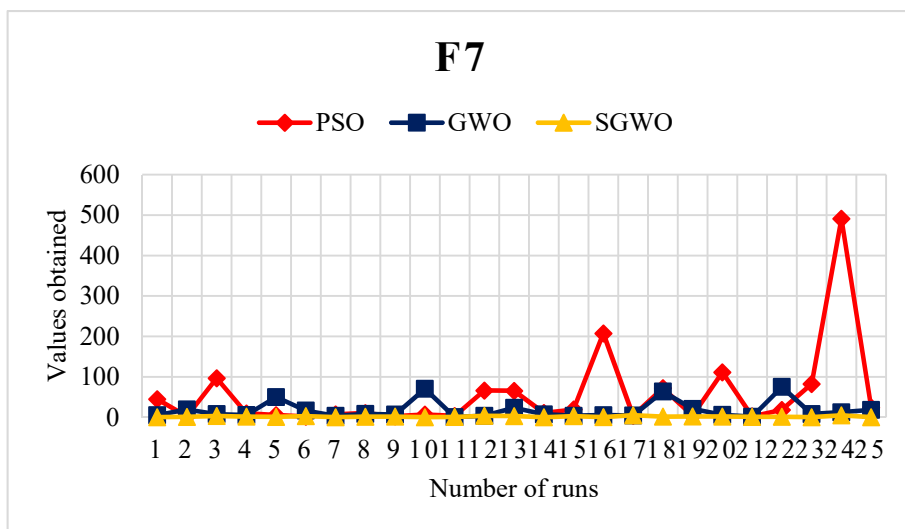




(Fig 2) Contd...



(Fig 2) Contd...



(Fig 2) Contd...

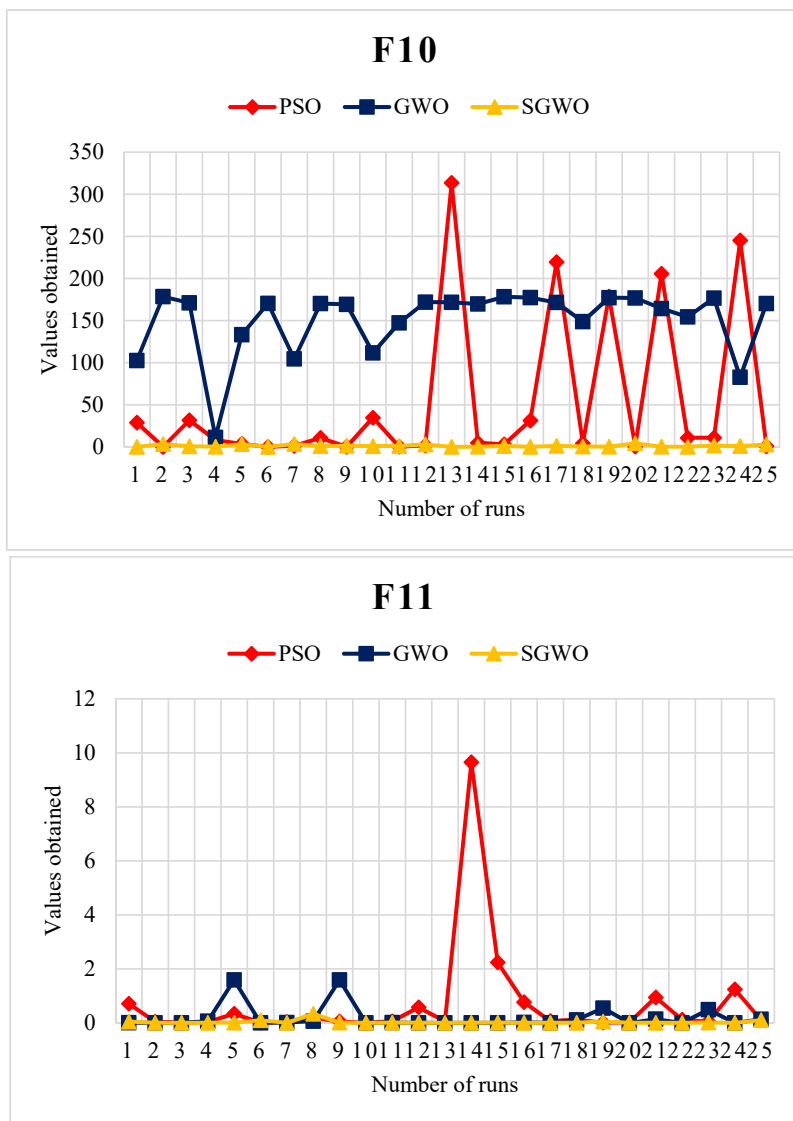


Fig. (2). (F1-F11) Comparison graphs of values obtained by benchmark functions. (A higher resolution / colour version of this figure is available in the electronic copy of the article).

**CONCLUSION**

In this paper, an adjustable hybrid of the PSO and the GWO has been proposed. Our strategy aims to prevent the PSO from getting trapped in local minima and merging the strengths of both. With this intention, two driving parameters are employed to adjust the control of the optimization process between the PSO and the GWO. We have analysed our algorithm on eleven benchmark functions and compared it with the PSO and the GWO. The numerical and statistical results of the experiments reveal that the proposed hybrid is better in comparison to both PSO and GWO in the searchability, quality of solution, stability of solution, and ability to seek global optima. The proposed hybrid can be used for solving various real life optimization problems. Though the SGWO algorithm refines result quality, the computational complexity also gets elevated. Thus, lowering the computational complexity would be another issue of future work.

Moreover, we will apply the proposed hybrid for tuning the parameters of the fuzzy logic controller to carry out water quality estimation and prediction.

**CONSENT FOR PUBLICATION**

Not applicable.

**AVAILABILITY OF DATA AND MATERIALS**

The data supporting the findings of this study is available in this article.

**FUNDING**

None.

**CONFLICT OF INTEREST**

The authors declare no conflict of interest, financial or otherwise.

## ACKNOWLEDGEMENTS

Declared none.

## REFERENCES

- [1] Wolpert, D.H.; Macready, W.G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.*, **1997**, *1*(1), 67-82. <http://dx.doi.org/10.1109/4235.585893>
- [2] Yang, X.S. Swarm intelligence based algorithms: a critical analysis. *Evol. Intell.*, **2014**, *7*(1), 17-28. <http://dx.doi.org/10.1007/s12065-013-0102-2>
- [3] Arani, B.O.; Mirzabeygi, P.; Panahi, M.S. An improved PSO algorithm with a territorial diversity-preserving scheme and enhanced exploration-exploitation balance. *Swarm Evol. Comput.*, **2013**, *11*, 1-15. <http://dx.doi.org/10.1016/j.swevo.2012.12.004>
- [4] Wang, Y.; Li, B.; Weise, T.; Wang, J.; Yuan, B.; Tian, Q. Self-adaptive learning based particle swarm optimization. *Inf. Sci.*, **2011**, *181*(20), 4515-4538. <http://dx.doi.org/10.1016/j.ins.2010.07.013>
- [5] Kessentini, S.; Barchiesi, D. In: *A new strategy to improve particle swarm optimization exploration ability*, Proceedings of the IEEE International Conference on 2010 Second WRI Global Congress on Intelligent Systems, Wuhan, China, December 16-17, 2010; IEEE: 2011; Vol. 1, pp. 27-30. <http://dx.doi.org/10.1109/GCIS.2010.147>
- [6] Zhang, J.R.; Zhang, J.; Lok, T.M.; Lyu, M.R. A hybrid particle swarm optimization-back-propagation algorithm for feedforward neural network training. *Appl. Math. Comput.*, **2007**, *185*(2), 1026-1037. <http://dx.doi.org/10.1016/j.amc.2006.07.025>
- [7] Ouyang, A.; Zhou, Y.; Luo, Q. In: *Hybrid particle swarm optimization algorithm for solving systems of nonlinear equations*, Proceedings of the IEEE International Conference on Granular Computing (GRC '09), Nanchang, China, August 17-19, 2009; IEEE: 2009; pp. 460-465. <http://dx.doi.org/10.1109/GRC.2009.5255079>
- [8] Mirjalili, S.; Hashim, S.Z.M. In: A new hybrid PSO-GSA algorithm for function optimization, Proceedings of ICCIA 2010 - 2010 International Conference on Computer and Information Application, Tianjin, China, November 2-4, 2010; Torrens University Australia, pp. 374-377.
- [9] Yu, S.; Wu, Z.; Wang, H.; Chen, Z.; Zhong, H. A hybrid particle swarm optimization algorithm based on space transformation search and a modified velocity model. *Int. J. Numer. Anal. Model.*, **2012**, *9*(2), 522-527. [http://dx.doi.org/10.1007/978-3-642-11842-5\\_73](http://dx.doi.org/10.1007/978-3-642-11842-5_73)
- [10] Esmine, A.A.; Matwin, S. HPSOM: A hybrid particle swarm optimization algorithm with genetic mutation. *Int. J. Innov. Comput., Inf. Control*, **2013**, *9*(5), 1919-1934.
- [11] Chang, J.X.; Bai, T.; Huang, Q.; Yang, D.W. Optimization of water resources utilization by PSO-GA. *Water Resour. Manage.*, **2013**, *27*(10), 3525-3540. <http://dx.doi.org/10.1007/s11269-013-0362-8>
- [12] Yu, X.; Cao, J.; Shan, H.; Zhu, L.; Guo, J. An adaptive hybrid algorithm based on particle swarm optimization and differential evolution for global optimization. *ScientificWorldJournal*, **2014**, *2014*, 215472. <http://dx.doi.org/10.1155/2014/215472> PMID: 24688370
- [13] Abd-Elazim, S.M.; Ali, E.S. A hybrid particle swarm optimization and bacterial foraging for power system stability enhancement. *Complexity*, **2015**, *21*(2), 245-255. <http://dx.doi.org/10.1002/cplx.21601>
- [14] Kumar, N.; Vidyarthi, D.P. A novel hybrid PSO-GA meta-heuristic for scheduling of DAG with communication on multiprocessor systems. *Eng. Comput.*, **2016**, *32*(1), 35-47. <http://dx.doi.org/10.1007/s00366-015-0396-z>
- [15] Garg, H. A hybrid PSO-GA algorithm for constrained optimization problems. *Appl. Math. Comput.*, **2016**, *274*, 292-305. <http://dx.doi.org/10.1016/j.amc.2015.11.001>
- [16] Ghasemi, E.; Kalhori, H.; Bagherpour, R. A new hybrid ANFIS-PSO model for prediction of peak particle velocity due to bench blasting. *Eng. Comput.*, **2016**, *32*(4), 607-614. <http://dx.doi.org/10.1007/s00366-016-0438-1>
- [17] Javidrad, F.; Nazari, M. A new hybrid particle swarm and simulated annealing stochastic optimization method. *Appl. Soft Comput.*, **2017**, *60*, 634-654. <http://dx.doi.org/10.1016/j.asoc.2017.07.023>
- [18] Ali, A.F.; Tawhid, M.A. A hybrid particle swarm optimization and genetic algorithm with population partitioning for large scale optimization problems. *Ain Shams Eng. J.*, **2017**, *8*(2), 191-206. <http://dx.doi.org/10.1016/j.asej.2016.07.008>
- [19] Hasanipanah, M.; Shahnazar, A.; Amnieh, H.B.; Armaghani, D.J. Prediction of air-overpressure caused by mine blasting using a new hybrid PSO-SVR model. *Eng. Comput.*, **2017**, *33*(1), 23-31. <http://dx.doi.org/10.1007/s00366-016-0453-2>
- [20] Kaur, S.; Mahajan, R. Hybrid meta-heuristic optimization based energy efficient protocol for wireless sensor networks. *Egyptian Inform. J.*, **2018**, *19*(3), 145-150. <http://dx.doi.org/10.1016/j.eij.2018.01.002>
- [21] Chopra, N.; Kumar, G.; Mehta, S. Hybrid GWO-PSO algorithm for solving convex economic load dispatch problem. *Int J Res Adv Technol*, **2016**, *4*(6), 37-41.
- [22] Kamboj, V.K. A novel hybrid PSO-GWO approach for unit commitment problem. *Neural Comput. Appl.*, **2016**, *27*(6), 1643-1655. <http://dx.doi.org/10.1007/s00521-015-1962-4>
- [23] Singh, N.; Singh, S.B. Hybrid algorithm of particle swarm optimization and grey wolf optimizer for improving convergence performance. *J. Appl. Math.*, **2017**, *2017*, 2030489. <http://dx.doi.org/10.1155/2017/2030489>
- [24] Şenel, F.A.; Gökçe, F.; Yüksel, A.S.; Yiğit, T. A novel hybrid PSO-GWO algorithm for optimization problems. *Eng. Comput.*, **2019**, *35*(4), 1359-1373. <http://dx.doi.org/10.1007/s00366-018-0668-5>
- [25] Teng, Z.J.; Lv, J.L.; Guo, L.W. An improved hybrid grey wolf optimization algorithm. *Soft Comput.*, **2019**, *23*(15), 6617-6631. <http://dx.doi.org/10.1007/s00500-018-3310-y>
- [26] Kennedy, J.; Eberhart, R. In: Particle swarm optimization, Proceedings of ICNN'95-international conference on neural networks, Perth, WA, Australia, 27 November - 1 December, 1995; IEEE, 2002; Vol. 4, pp. 1942-1948. <http://dx.doi.org/10.1109/ICNN.1995.488968>
- [27] Shi, Y.; Eberhart, R. In: *A modified particle swarm optimizer*, Proceedings of 1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360), Anchorage, AK, USA, May 4-9, 1998; IEEE, 2002; pp. 69-73. <http://dx.doi.org/10.1109/ICEC.1998.699146>
- [28] Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.*, **2014**, *69*, 46-61. <http://dx.doi.org/10.1016/j.advengsoft.2013.12.007>