Applied Soft Computing 66 (2018) 232–249

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A hybrid firefly and particle swarm optimization algorithm for computationally expensive numerical problems

İbrahim Berkan Aydilek

Department of Computer Engineering, Faculty of Engineering, Harran University, Şanlıurfa, Turkey

ARTICLE INFO

Article history:
Received 24 December 2016
Received in revised form 9 December 2017
Accepted 16 February 2018
Available online 21 February 2018

Keywords:
Hybrid optimization
Firefly algorithm
Particle swarm optimization

ABSTRACT

Optimization in computationally expensive numerical problems with limited function evaluations provides computational advantages over constraints based on runtime requirements and hardware resources. Convergence success of a metaheuristic optimization algorithm depends on directing and balancing of its exploration and exploitation abilities. Firefly and particle swarm optimization are successful swarm intelligence algorithms inspired by nature. In this paper, a hybrid algorithm combining firefly and particle swarm optimization (HFPSO) is proposed. The proposed algorithm is able to exploit the strongpoints of both particle swarm and firefly algorithm mechanisms. HFPSO try to determine the start of the local search process properly by checking the previous global best fitness values. In experiments, several dimensional CEC 2015 and CEC 2017 computationally expensive sets of numerical and engineering, mechanical design benchmark problems are used. The proposed HFPSO is compared with standard particle swarm, firefly and other recent hybrid and successful algorithms in limited function evaluations. Runtimes and convergence accuracies are statistically measured and evaluated. The solution results quality of this study show that the proposed HFPSO algorithm provides fast and reliable optimization solutions and outperforms others in unimodal, simple multimodal, hybrid, and composition categories of computationally expensive numerical functions.

1. Introduction

Finding optima in many real-world optimization problems requires expensive evaluations in terms of computation. Because of some limitations in studies like project time requirements and computation resource constraints, the optimization process should be conducted quickly and it should not be too complex [1]. Many standard optimization algorithms require a large number of function evaluations. These algorithms usually give satisfactory results by use of their special information transfer mechanisms with a number of first candidate solutions in a number of fitness evaluations. Due to the evaluation of each candidate resolution, these processes often require computer-intensive computer resources and runtime. For this reason, the study and development of successful optimization algorithms for evaluating a limited number of functions is emerging and developing study area. In recent years, many new approaches have been introduced and published. Some satisfactory results have been obtained from these methods with limited function evaluations [1–3].

Nature-inspired algorithm is a computational technique inspired by observation in nature [4–6]. Swarm intelligence is a set of social behaviors pre-determined by some rules. Some of the single individuals do not act wisely enough. However, behavior of all individuals in together can be wise enough with the help of the swarm cooperation [7]. One of the most well-known and researched nature-inspired and swarm intelligence methods is particle swarm optimization (PSO) algorithm. PSO inspired by living or hunting life styles of birds and fish. It is widely used in difficult and complex optimization problems. Since the first introduction of PSO [8], various PSO algorithm variants have been developed by participants to find better algorithm in some specific problems. These developments in PSO variants can be classified roughly into four categories: The first category algorithms are based on parameter settings focusing on the optimization of inertia weight and acceleration coefficients parameters. The second variant algorithm category is based on the neighborhood topology, which describes the connection of particles to each other. The algorithms of the third variant category are based on learning strategies consisting of teaching and peer learning of the particle bests and global best positions and the last fourth category is hybrid variants using the mixture of PSO with other suitable optimization algorithms [9]. Primary objectives of hybrid PSO variants are to establish a

E-mail address: berkanaydilek@harran.edu.tr

https://doi.org/10.1016/j.asoc.2018.02.025
1568-4946/

balance between exploration and exploitation and to avoid premature convergence. Moreover, hybridization can contribute to the use of the PSO's powerful capabilities and to remove the weakness [10]. Known strong PSO abilities are easy implementation, less computing resource requirements, and fast convergence. On the other hand, it also has some weaknesses such as premature convergence or trapping in local optima and slow convergence rate in the exploitation where particles are close to each other or global optimum [4]. Firefly Algorithm (FA) is another nature-inspired and swarm intelligence optimization algorithm that mimics fireflies in the nature. FA has some advantages over PSO [11] algorithm. These advantages are: FA does not have an individual best or a certain global best and this prevents trapping in local minima or premature convergence disadvantages. Also, the fireflies of FA algorithm do not have velocity characteristic. Thus, other problems that are based on fast or slow velocity, can be prevented [12]. Although FA has good characteristic in local search, sometimes it is unable to escape from local search completely and traps in a local minima [13].

Hybrid optimization technique is a successful combination of metaheuristic algorithm with another optimization algorithm that can display a more robust behavior and exhibit greater flexibility against complex and difficult problems [14]. Local search algorithms iteratively scan the search space to find a preferable solution than existing solution using appropriately defined neighborhood mechanism [14]. Metaheuristic is composed of some iterative generation operations that efficiently combines different sub-heuristics to discover a search space. Some learning strategies are used to find global optimum areas [14,15]. Population-based metaheuristics are natural approaches that explore the search field by manipulating the population and final results highly depend on their unique manipulating methods [14]. Population-based metaheuristics methods are better at describing local optima than other trajectory methods which can be easily influenced by local optima. Therefore, metaheuristic hybrids, which can combine the strengths of both population-based and trajectory methods in a proper way, are usually very efficient and successful [14]. For example, Li et al. [16] proposed an approach that combines, global search ability of genetic algorithm (GA) and the fast convergence mechanism of PSO to find global optimum more precisely. A numerous of studies propose different types of hybrid optimization techniques that combine their powerful mechanisms. For this reason, they are often more productive in terms of running time and/or solution results quality [17]. Memetic algorithms are based on exploitation systematic and the combination of population-based and trajectory metaheuristics synergistic. A known technique to create this hybridation is to include local search add-ons in an evolutionary algorithm [18].

In this study, CEC 2015 and CEC 2017 [1,19], realistic engineering design optimization benchmark problems are used in a limited number of function evaluations. Therefore, fast convergence ability of PSO and good local search ability of FA are used in a hybrid way and obtained results are compared with other recent hybrid PSO and FA approaches. Rest of the paper is organized as follows: Literature review is given in Section 2. PSO and FA algorithms are briefly explained in Sections 3 and 4, respectively. Other recent hybrid-related works are analyzed and explained in Section 5. Proposed hybrid algorithm is explained in Section 6. Detailed flowchart and pseudo codes of the proposed algorithm are shown and given. Computationally expensive CEC 2015 and CEC 2017, engineering and mechanical design problem sets are explained and presented in Section 7. Experimental setup, results and discussions are also given by using of figures and tables. Finally, conclusions of the whole paper are detailed in Section 8.

2. Literature review

Petalas et al. [20] propose a new efficient and robust memetic particle swarm optimization (MPSO) algorithm that improves local search ability of the standard particle swarm algorithm. The proposed algorithm is tested with various unconstrained, constrained, minimax and integer programming problems. The obtained results show, memetic approach of PSO outperforms the local and global versions of the standard PSO.

Wang et al. [21] propose a new firefly algorithm (NFA) with local search for numerical optimization. Their algorithm addresses an issue of firefly algorithm. If a firefly is brighter than the compared firefly, then this firefly can not begin any search process. In their proposed algorithm, brighter firefly can move based on local search principles. Experiments are done with common benchmark functions. Results of new algorithm outperforms the standard and memetic firefly algorithm versions.

Huang et al. [22] propose a space search optimization algorithm (SSOA) with improved convergence method against to the problems that are caused from purely random search mechanism. An opposition-based space search ability of their algorithm provides new solutions to the problem of trapped in local optima and it prevents premature convergences. In experiments CEC 2008 benchmark problems are used. Results show that proposed space search mechanism can effectively reduce an important difficulty encountered in evolutionary algorithms.

Li et al. [16] implement a heuristic space search mechanism that is used in particle swarm optimization. Genetic algorithm (GA) divides search space into smaller spaces. Therefore, the risk of trapping in a local minima are reduced and eliminated. Improved PSO algorithm can convergence to the global optima and find global optima more precisely. Eleven common benchmark test functions are used to assess the proposed approach. The obtained results show that this new algorithm can optimize all functions with high success.

A variance of PSO algorithm called LHNPSO has been proposed by Yang et al. [23]. In this new method, nonlinear functions are used to calculate inertia weight that varies upon iteration by the help of the other cognitive, social parameters and acceleration coefficients. Suggested method is controlled with a group of functions consisting of well-known evaluation test functions. Proposed algorithm proved that, the first particles that are produced with the Halton, have obtained the minimum discrepancy and efficiently searched the search space. The results indicate that this updated PSO method rapidly converges and gives more accurate results compared to other three PSO, LPSO and LPSO-TVAC methods.

D'Andreagiovanni [24] has presented a genetic algorithm to solve large examples of the Power, Frequency and Modulation Assignment Problem involved in the design of wireless networks. He shows how a genetic algorithm can be used effectively in order to better explore the field of solutions of precise algorithm solutions for wireless network design problems. D'Andreagiovanni et al. [25] have suggested an advanced ant-colony algorithm which exploits appropriate linear relaxations to direct a probabilistic valorization of variables, with the aid of an extensive variable neighborhood search to design wireless body sensor networks. Obtained results on realistic examples show that improved ant-colony algorithm performs better than an efficient solver, providing quick solutions associated with improved optimization gaps. In another study, D'Andreagiovanni et al. [26] have proposed hybrid primal heuristic method which combines a randomized fixing strategy inspired by ant colony optimization algorithm and an extensive neighborhood search. They show how an approximation algorithm and linear relaxations can be integrated with an ant colony-like algorithm to get a very effective and fast algorithm for multi-period network design. D'Andreagiovanni et al. [27] have also proposed to solve a Connected Facility Location Problem arising in the design of telecommunication access networks by a bio-inspired approach in which they integrate the ant colony-like variable fixing, which is guided by linear relaxations, with MILP heuristic that utilizes a detailed neighborhood search. Computational results and experiments show that proposed heuristic can provide better solutions than a state-of-the-art solver.

Gambardella et al. [28] have demonstrated how an integration of ant colony algorithms with solution improvement phases following some general simple yet effective rules can lead to very high improvements in the quality of final solutions for solving a telecommunication network design problem. Two operations are suggested to increase the performance of ant colony algorithms: the first one, considers the constructive phase and the second, the combining the constructive phase and the local search mechanism. These operations cause remarkable speedup in standard algorithm.

Tanweer et al. [9] have introduced self-regulating particle swarm optimization (SRPSO). In their paper, the best particle uses its own direction to update next position. Rest particles use historical personal best position and self-perception of global direction to update next positions. Thus, the best particle is used for better exploration and rest particles are used for intelligently exploitation of the solution space. The SRPSO algorithm has been tested with CEC 2005 problems. The results are compared with other successful PSO variants and provide better solutions. Disadvantage of the SRPSO is that only the best particle is used for better exploration. The method proposed in our paper can employ all particles that have better fitness values than previous global best particle.

Zhang et al. [29] have developed Bayesian PSO method. They used a probability intensity function based on Bayes principles for parameter setting. In this way, it is demonstrated that proposed method achieved success against regular PSO algorithm. Disadvantage of this study is that Bayesian approach frequently faces local optima problem. Especially, it is difficult to get successful results in complex multiple minima problems.

Ngo et al. [4] have proposed the Extraordinariness Particle Swarm Optimizer (EPSO). They state that in the standard PSO, the velocity of a particle is calculated based on global best and particle best position values. Therefore, trapping in local minima and fast premature convergences can occur. To avoid this, a new particle motion is introduced that is different from the standard PSO. In EPSO, particles can walk towards the best or the worst particles. Disadvantage of this study is that determining velocity precisely can already be a problem in convergence success.

3. Particle swarm optimization algorithm

Particle swarm algorithm is a swarm and population-based optimization technique that is inspired by social behaviors of bird and fish swarms. Particle swarm optimization (PSO) has been proposed by Kennedy and Eberhart [8,30]. Over the years, it becomes one of the most important algorithms and attracts increasing attention, since, as a new swarm intelligence-based algorithm, it makes it possible to solve complex optimization problems [8,31]. Special information transfer mechanism among particles of PSO facilitates solving different problems with good performance at low computational cost [31]. Simple implementation, minimum mathematical processing and good optimization ability are other advantages of this algorithm [9,31]. Vassiliadis and Dounias [32] review: it is a nature-inspired optimization approach that can be considered as the most frequently and commonly used one among the other nature-inspired based algorithms according to the related publications [31] and also it is one of the most preferred algorithms for different benchmark and engineering optimization problems, because it needs less memory requirements and has an easy implementation [9]. PSO is found to be robust and fast against to nonlinear, non-differentiable, and multi-modal optimization problems [33].

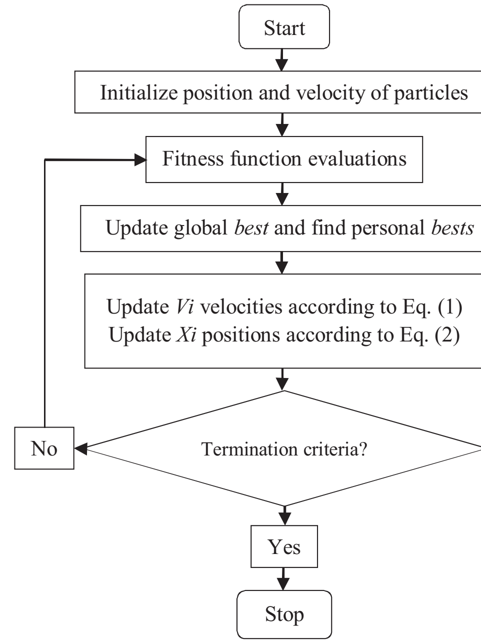


Fig. 1. Standard Particle Swarm Optimization algorithm flowchart.

However, there are some disadvantages of PSO. When implementing variance of this algorithm, it is necessary to get out of the local optima and avoid premature convergence. If the variable size of a problem increases, the optimization problem becomes more complex, thus, the probability of finding global optimum decreases [31]. Shi and Eberhart [33] state that due to PSO algorithm is still in its infancy, a lot of work and research are needed.

In Fig. 1, flowchart of standard particle swarm optimization algorithm is given. In PSO algorithm, every particle demonstrates a possible solution [34]. Particles come together to form a swarm. Social and cognitive information are shared among the particles by algorithm which seeks to find a reasonable value [35]. Each particle has a position (X_i) and velocity (V_i) value. Position and velocity are updated according to Eqs. (1) and (2), respectively. At the start, the initial population are initialized with randomly position and velocity values in the specific range. At the end, the optimization occurs when the particles reach their final positions. In case of zero velocity initialization, if a particle has a better neighbor than itself then it may quit from search space during the first iteration [36]. Fitness function varies depending on the study [37]. According to Eqs. (1) and (2); position of *i.* particle is X_i and velocity is V_i. *t* indicates current, *t+1* indicates next iteration values of the algorithm. Particle cognitive best value is *pbest* and group, social global best value is symbolized by *gbest* in the equations [38].

$$V_i(t+1) = wV_i(t) + c_1 r_1 (pbest_i(t) - X_i(t)) + c_2 r_2 (gbest(t) - X_i(t)) \quad (1)$$

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

An improved variant of PSO algorithm that employs inertia weight (w) parameter is developed by [33], but standard PSO does not contain this parameter [8]. r_1 and r_2 are random numbers in range of [0,1]. c_1 and c_2 values are acceleration coefficients that determine the size of acceleration applied on particle cognitive and group, social value [39]. A recent work by Harrison et al. [40] is on the inefficiency of existing approaches to dynamically adjusting PSO control parameters.

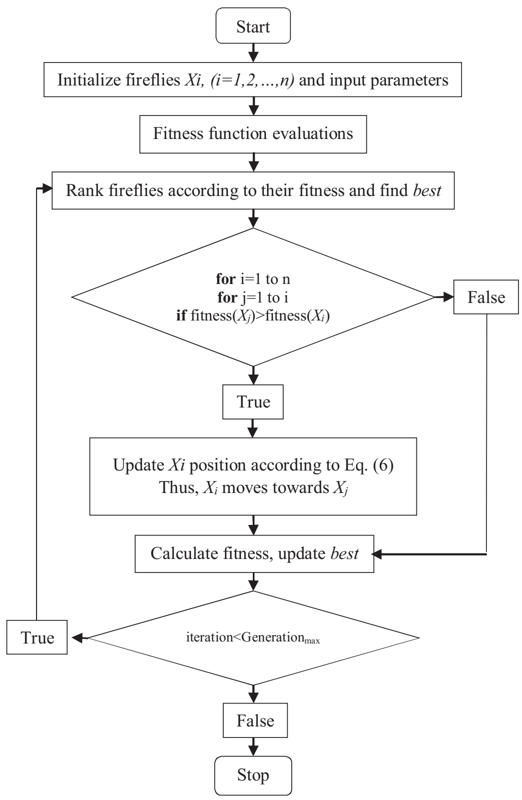


Fig. 2. Firefly algorithm flowchart.

4. Firefly optimization algorithm

Firefly algorithm was inspired by fireflies, a kind of insect that lives in the nature. Most fireflies generate rhythmic and short flashes. Type of flashes is often unique for a specific kind. Flashing light is generated by a bioluminescence process, and its real function is still a matter of debate among researchers. Fireflies use their chemical light attractiveness for communication, hunting, and warning their enemies [11,41,42]. In Fig. 2, Firefly algorithm flowchart is given.

Inverse square law is used to calculate light intensity (I) at a specific distance (r) from a light source. Therefore, light intensity decreases opposed of distance increases. Additionally, the air absorbs light and its intensity decreases and becomes weaker as distance increases. For these reasons, most fireflies can be seen from a few hundred meters and it is usually satisfactory distance for fireflies to communicate. Thus, flashing light can be formulated as the objective function to be optimized, so it provides a new population-based nature-inspired Firefly optimization algorithm (FA) [41].

According to inverse square law, a light intensity ($I(r)$) at r distance from a light source (l_s) is can be calculated by Eq. (3).

$$I(r) = l_s/r^2 \tag{3}$$

In an environment, light is absorbed with a constant light absorption coefficient (γ) $\in [0, \infty)$. Thus, equation can be formed in Gaussian by Eq. (4)

$$B(r) = B_0 e^{-\gamma r^2} \tag{4}$$

$B(r)$ is attractiveness of a firefly at r distance and B_0 is attractiveness when $r = 0$.

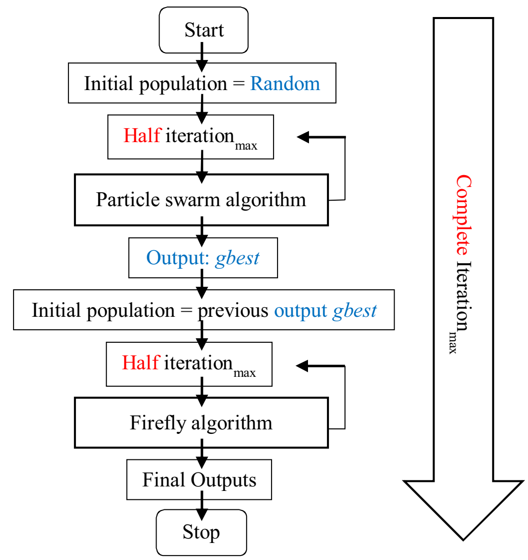


Fig. 3. HPSOFF algorithm summary flowchart.

Assuming that, i ve j are two fireflies and their positions are $X_i(x_i, y_i)$ and, $X_j(x_j, y_j)$ respectively. Distance (r_{ij}) between two fireflies is calculated based on Euclidean by Eq. (5)

$$r_{ij} = \|X_i - X_j\| = \sqrt{\left(x_i - x_j\right)^2 - \left(y_i - y_j\right)^2} \tag{5}$$

Thus, less brighter firefly i's new position (X_i) and movement towards more attractive firefly j is calculated by Eq. (6)

$$X_i = X_i + B_0 e^{-\gamma r_{ij}^2}\left(X_j - X_i\right) + a \in_i \tag{6}$$

In Eq. (6), $\in_i$ is vector of random variables and Randomization parameter (α) $\in$ [0, 1] [11,41,42].

5. Other recent related works

5.1. Hybrid particle swarm optimization and firefly (HPSOFF) algorithm

Arunachalam et al. [43] have proposed a hybrid particle swarm optimization and firefly algorithm (HPSOFF). In their paper, they propose a new approach, which has conflicting economic and emission objectives, to Combined Economic and Emission Dispatch (CEED) problem using a Hybrid Particle Swarm Optimization and Firefly (HPSOFF) algorithm that is shown in Fig. 3. In standard FA algorithm, initial random population is updated in each iteration until the last iteration. Thus, final result of optimization depends on the quality of the first initial random population. Their proposed hybrid method uses initial population of FA from PSO optimal result. Half of the maximum number of function evaluations (*MaxFES*) is used by following each other algorithms.

5.2. Hybrid firefly and particle (FFPSO) optimization algorithm

Kora et al., [44] have proposed hybrid firefly and particle swarm optimization (FFPSO) algorithm for detection of Bundle Branch Block (BBB). In this paper, for detection of Bundle Branch Block, hybrid firefly and particle Swarm Optimization (FFPSO) technique is used to help Levenberg Marquardt Neural Network (LMNN) classifier. FFPSO is used for feature optimization of ECG (BBB and Normal) patterns. Optimized FFPSO features are given as input to LMNN classifier. They stated that, in local search of FA, small distances between fireflies cause to random walk and delay in convergence. So, in FFPSO method, local search is performed with modified calculation of mixed PSO characteristics as shown in Eqs. (7), (8) and (9).

$$r_{px} = \sqrt{\sum_{j=1}^{d} \left(pbest_{i,j} - x_{i,j}\right)^2} \tag{7}$$

$$r_{gx} = \sqrt{\sum_{j=1}^{d} \left(gbest - x_{i,j}\right)^2} \tag{8}$$

$$X_i(t+1) = wX_i(t) + c_1 e^{-r_{px}^2}(pbest_i - X_i(t)) + c_2 e^{-r_{gx}^2}(gbest - X_i(t)) + a \in_i \tag{9}$$

6. Proposed hybrid firefly and particle swarm optimization (HFPSO) algorithm

Achieving a reliable success in evaluating a limited number of functions is the main goal of the proposed HFPSO algorithm. Therefore, speed of convergence is important in the early stage of iterations. Particle swarm optimization algorithm has faster convergence ability rather than some other algorithms in some problems [45,46]. In local search stage of PSO, this fast convergence ability decreases and slows down especially when searching in the solution space close to a global optimal solution. Balance between exploration and exploitation in PSO can be efficiently controlled by using control parameters of PSO: inertia weight (w), acceleration coefficients (c_1, c_2). Velocity of a particle (V) can be calculated by the use of these control parameters according to Eq. (1). Thus, velocities are used for calculation of the next particles' new positions [33]. Amount of particle velocity value is significant in local search, but calculating the proper velocity can be a challenging task during local search, as improper values can cause particles to oscillate around optimal solution like a pendulum. These oscillation or swing movements cause some delays in the whole optimization task. Due to this problem, velocity can be ignored in exploitation stages. In the other hand, firefly algorithm does not have a velocity (V) characteristic. Yang [11] points out the superiority of firefly algorithm over PSO and genetic algorithms, and generally, global optimum is easily achieved in experimental problems in firefly algorithm paper. Also, there are no parameters in firefly algorithm to use previous best position of each firefly. Therefore, fireflies move regardless of their previous best positions [44].

In this paper, an optimization algorithm that combines search ability of firefly and particle swarm optimization algorithms has been proposed. Thus, a balance between exploration and exploitation is aimed to establish and it benefits strengths of both algorithms [44,47]. Fireflies have no velocity (V) and personal best position ($pbest$) memories in comparison to particles. In the proposed hybrid combination of two algorithms, PSO is generally used in the global search, because it provides fast convergence in exploration. Additionally, FA is generally used in local search, because it provides fine-tuning in exploitation. Dynamically adjusted inertia weight studies that consider improvements on previous personal best, have succeed [48,49]. In Fig. 4, flowchart of proposed hybrid of firefly and particle swarm optimization (HFPSO) algorithm is shown. First of all, input parameters that are used by both algorithms in the following steps are inserted. Next, uniform particle vectors are randomly prepared in the pre-defined search and velocity ranges. Global ($gbest$) and personal best ($pbest$) particles are calculated and assigned. In the following comparing stage, it is compared that if particle has an improvement in its fitness value in the last iteration according to Eq. (11). Then current position is saved in a temp variable (X_{i_temp}) and new position and velocity are calculated according to Eqs. (12) and (13).

$$w = w_i - \left(\left(w_i - w_f\right)/iteration_{max}\right) \times iteration \tag{10}$$

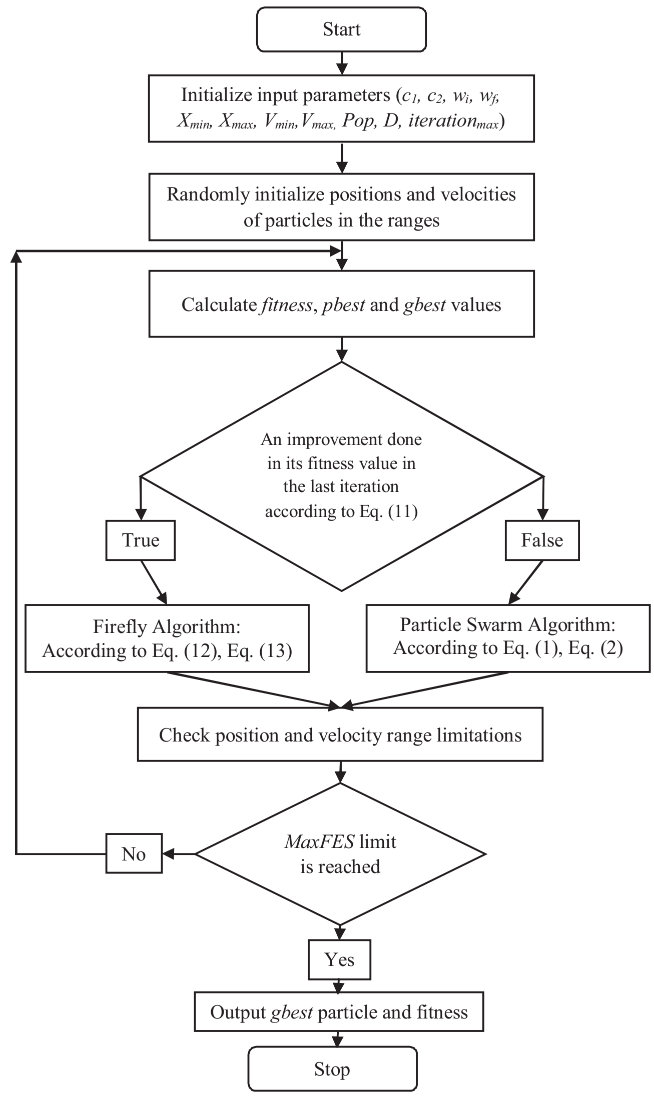


Fig. 4. Flowchart of proposed HFPSO algorithm.

$$f(i,t) = \begin{cases} true, \text{ if } fitness\left(particle_i^t\right) \le gbest^{t-1} \\ false, \text{ if } fitness\left(particle_i^t\right) > gbest^{t-1} \end{cases} \tag{11}$$

$$X_i(t+1) = X_i(t) + B_0 e^{-\gamma r_{ij}^2}\left(X_i(t) - gbest^{t-1}\right) + a \in_i \tag{12}$$

$$V_i(t+1) = X_i(t+1) - X_{i\_temp} \tag{13}$$

Thus, if a particle has a better or equal fitness value than previous global best, it is assumed that local search starts and particle is handled by an imitative FA, otherwise particle will be handled by PSO and PSO continues its standard processes with this particle according to Eqs. (1) and (2). In the following comparing stage, fitness function evaluations and range limitations are checked for all particles and fireflies. If maximum iteration limit is reached, hybrid algorithm will be terminated and *gbest* and its *fitness* value will be given as output of proposed hybrid algorithm.

In Algorithm 1, pseudo code of the proposed HFPSO optimization algorithm is given. Maximum number of fitness function evaluation (*MaxFES*) is used. In evolutionary computing, *MaxFES* is a popular termination criterion that is allowed the maximum calculation of objective functions [1]. Inertia weight (w) parameter helps to balance between exploration and exploitation in PSO [33]. A linear decreasing inertia weight is employed and calculated according to Eq. (10) [50]. Maximum and minimum velocities of a particle ($V_{\min}, V_{\max}$) are employed to limit next distance in a direction. They are randomly prepared at the beginning of the proposed algorithm in velocity range.

Algorithm 1 Proposed HFPSO optimization algorithm

Input: c_1, c_2 : Acceleration coefficients.
w : Inertia weight.
w_i, w_f : Initial and final values of linear decreasing inertia weight.
random : Random number generate function.
fitness : Fitness value calculate function.
pbest, gbest : Personal and global best positions.
X : Current positions of the particles.
X_{min} : Minimum search range limit.
X_{max} : Maximum search range limit.
V : Current velocities of the particles.
V_{min} : Minimum velocity limit.
V_{max} : Maximum velocity limit.
Pop : Size of the swarm.
D : Dimension of a particle.
t : Number of the current iteration.
$iteration_{max}$: Number of the maximum iteration.
Output: *gbest* particle and its fitness value.
Algorithm:
Initialize and assign input parameters (c_1, c_2, w_i, w_f, X_{min}, X_{max}, V_{min}, V_{max}, *Pop, D,* $iteration_{max}$)
Randomly initialize position *X[Pop][D]* matrix in the search range (X_{min}, X_{max})
Randomly initialize velocity *V[Pop][D]* matrix in the velocity range ($V_{\min}, V_{\max}$)
Calculate fitness, *pbest* and *gbest* values
while Maximum Fitness Evaluations (*MaxFES*=$iteration_{max}$ *x Pop*) limit is not reached **do**
  for i:= 1 to *Pop* **do**
    if particle has an improvement in its fitness value in the last iteration according to Eq. (11) **then**
      Save current particle position($X_i(t)$) in a temp variable (X_{i_temp})
      Update particle position according to Eq. (12)
      Update particle velocity according to Eq. (13)
    else
      Update inertia weight (w) according to Eq. (10)
      Update particle velocity according to Eq. (1)
      Update particle position according to Eq. (2)
    end if
    Check position and velocity range limitations
    Calculate new particle fitness function value
      if new particle fitness function value smaller than *pbest* **then**
        assign new *pbest*
      end if
      if new particle fitness function value smaller than *gbest* **then**
        assign new *gbest*
      end if
  end for
end while

7. Experiments and discussion

7.1. Experimental setup

Computationally expensive numerical CEC 2015, CEC 2017 and realistic engineering and mechanical design problems are used in experiments as benchmark test sets. Basic PSO, FA and other recent hybrid-related FFPSO and HPSOFF algorithms are compared with the proposed HFPSO algorithm. All optimized input parameters are set same for all hybrid algorithms in experiments and they are optimized and obtained from previous original papers [51]. Size of the swarm (*Pop*) and dimension of a particle (*D*) are set equal in algorithms. Thus, maximum iteration size is provided the equal in experiments. In FA algorithms, $a=0.2$, $B_0=2$, $\gamma=1$ and Euclidean distance $\left(r_{ij}\right)$ are normalized with the maximum distance [52]. In PSO algorithms acceleration coefficients are used with $c_1, c_2=1.49445$. Maximum velocity ($V_{\max}$) = 0.1 × Search range ($X_{\max}-X_{\min}$), where Minimum velocity ($V_{\min}$) = $-V_{\max}$. Inertia weight (w), $w_i=0.9$, $w_f=0.5$ are used according to Eq. (10) [50,53]. The linearly decreasing inertia weight is the most proper inertia weight strategy that has easy implementation and it is the most efficient way [9,54].

All results were obtained from 20 independent runs, and for each run, maximum number of function evaluations (*MaxFES*) was used from original competition of CEC problems paper [1], that was 500 for 10 dimensional problems (10D) and 1500 for 30 dimensional problems (30D). All evaluation criteria are strictly taken into consideration [1], experiments were done on a desktop computer which has Intel Core i7-4770HQ @ 2.20 GHz processor, 16 GB memory, Windows10 OS configuration, and experiment software, all algorithms were coded by Matlab2015a software.

7.2. CEC 2015 and CEC 2017 computationally expensive optimization test problems

CEC 2015 benchmark problems [1] are highly competitive and require efficient optimization algorithms to provide fast solutions with accuracy in limited allocated budgets. CEC 2015 benchmark functions is the collection of 15 challenging expensive budget optimization problems, including highly complex composite and hybrid problems, too [55]. CEC 2017 benchmark problems [19] are the recent collection of 30 functions. Both sets of benchmark

Table 1
Summary of CEC 2015 expensive benchmark problems.

CEC 2015			
Type	No.	Description	Fi*
Unimodal functions	1	Rotated Bent Cigar Function	100
	2	Rotated Discus Function	200
Simple Multimodal Functions	3	Shifted and Rotated Weierstrass Function	300
	4	Shifted and Rotated Schwefel's Function	400
	5	Shifted and Rotated Katsuura Function	500
	6	Shifted and Rotated HappyCat Function	600
	7	Shifted and Rotated HGBat Function	700
	8	Shifted and Rotated Expanded Griewank's plus Rosenbrock's Function	800
	9	Shifted and Rotated Expanded Scaffer's F6 Function	900
Hybrid functions	10	Hybrid Function 1 (N = 3)	1000
	11	Hybrid Function 2 (N = 4)	1100
	12	Hybrid Function 3 (N = 5)	1200
Composition Functions	13	Composition Function 1 (N = 5)	1300
	14	Composition Function 2 (N = 3)	1400
	15	Composition Function 3 (N = 5)	1500

Table 2
Summary of CEC 2017 expensive benchmark problems.

CEC 2017			
Type	No.	Description	Fi*
Unimodal functions	1	Shifted and Rotated Bent Cigar Function	100
	2	Shifted and Rotated Sum of Different Power Function	200
	3	Shifted and Rotated Zakharov Function	300
Simple Multimodal Functions	4	Shifted and Rotated Rosenbrock's Function	400
	5	Shifted and Rotated Rastrigin's Function	500
	6	Shifted and Rotated Expanded Scaffer's F6 Function	600
	7	Shifted and Rotated Lunacek Bi_Rastrigin Function	700
	8	Shifted and Rotated Non-Continuous Rastrigin's Function	800
	9	Shifted and Rotated Levy Function	900
	10	Shifted and Rotated Schwefel's Function	1000
Hybrid functions	11	Hybrid Function 1 (N = 3)	1100
	12	Hybrid Function 2 (N = 3)	1200
	13	Hybrid Function 3 (N = 3)	1300
	14	Hybrid Function 4 (N = 4)	1400
	15	Hybrid Function 5 (N = 4)	1500
	16	Hybrid Function 6 (N = 4)	1600
	17	Hybrid Function 6 (N = 5)	1700
	18	Hybrid Function 6 (N = 5)	1800
	19	Hybrid Function 6 (N = 5)	1900
	20	Hybrid Function 6 (N = 6)	2000
Composition Functions	21	Composition Function 1 (N = 3)	2100
	22	Composition Function 2 (N = 3)	2200
	23	Composition Function 3 (N = 4)	2300
	24	Composition Function 4 (N = 4)	2400
	25	Composition Function 5 (N = 5)	2500
	26	Composition Function 6 (N = 5)	2600
	27	Composition Function 7 (N = 6)	2700
	28	Composition Function 8 (N = 6)	2800
	29	Composition Function 9 (N = 3)	2900
	30	Composition Function 10 (N = 3)	3000

problems consist of unimodal, simple multimodal, hybrid, and composition type of functions shown in Tables 1 and 2. 'Fi*' is the global optimum value and the search range of variable bounds $\in [-100,100]$.

A unimodal function has a non-separable and sensitive direction or smooth but narrow ridge. A multimodal function can be continuous everywhere yet differentiable nowhere or only a set of points, on these functions the global optimum point may be far away from a local optimum point and also there may be too much local optimum points. In real-world, subsets of a problem may have diverse characteristic. In hybrid functions, variables are randomly divided into different subsets and each calculation of these subsets is done by use of a different characteristic function (Modified Schwefel's, Rastrigin's, High Conditioned Elliptic etc.). Composition function combines properties of some sub-functions (Rosenbrock's, Bent Cigar, Discus etc.) better and provides durability around global or local optima [1].

In Table 3, 10 dimensional (10D) and 30 dimensional (30D) CEC 2015 problems' final mean optimized output values are shown. According to the table, the proposed HFPSO has better final mean values and outperforms other algorithms in all 30D problems which are unimodal, simple multimodal, hybrid, and composition function types. However, in F8, F9, F11, F13, F15 hybrid and composition 10D problems, HPSOFF algorithm slightly surpasses the proposed HFPSO which has lower chance to concrete for convergence with reduced and randomly divided variables into subsets on local search. In HPSOFF algorithm, as we know, firefly algorithm obtains its first initial population from PSO algorithm outputs. Additionally, FA is the best only in F2, and PSO is the best only in F10. When Table 3 analyzed, it can be seen that HFPSO can find the lowest mean minimization values in 8 and 15 out of 15 problems for

Table 3
CEC 2015 results summary.

	PSO		FA		FFPSO		HPSOFF		HFPSO	
Fn.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
10D										
1	2,4553E+08	1,3549E+08	4,3059E+08	2,8945E+08	1,6287E+10	4,9786E+09	4,8387E+07	3,4292E+07	***1,3768E+07***	6,6375E+06
2	3,8112E+04	1,5114E+04	***3,3304E+04***	9,7404E+03	1,4957E+08	4,6261E+08	3,8331E+04	1,2383E+04	3,8542E+04	1,5696E+04
3	3,0779E+02	1,3259E+00	3,0773E+02	1,2487E+00	3,1455E+02	1,6124E+00	3,0845E+02	1,5636E+00	***3,0671E+02***	1,4189E+00
4	2,2534E+03	3,5521E+02	1,5473E+03	3,2112E+02	3,1120E+03	2,6203E+02	1,7084E+03	3,0718E+02	***1,3159E+03***	3,9950E+02
5	5,0277E+02	6,3611E−01	5,0293E+02	5,9796E−01	5,0350E+02	9,3430E−01	5,0273E+02	8,2275E−01	***5,0250E+02***	5,7466E−01
6	6,0089E+02	2,8490E−01	6,0092E+02	5,8361E−01	6,0673E+02	1,3580E+00	6,0063E+02	1,4097E−01	***6,0054E+02***	1,4584E−01
7	7,0193E+02	1,8947E+00	7,0586E+02	5,8077E+00	8,0586E+02	3,3329E+01	7,0087E+02	9,3694E−01	***7,0060E+02***	2,5433E−01
8	8,1583E+02	2,7690E+01	8,6344E+02	1,7256E+02	2,7632E+05	2,7423E+05	***8,0740E+02***	2,8927E+00	8,0773E+02	4,0866E+00
9	9,0391E+02	3,2749E−01	9,0395E+02	2,3842E−01	9,0451E+02	1,7883E−01	***9,0388E+02***	2,3372E−01	9,0393E+02	2,6387E−01
10	***2,9540E+05***	1,9786E+05	5,3162E+05	6,5054E+05	5,1186E+07	7,7896E+07	3,5402E+05	2,9730E+05	3,3099E+05	3,3036E+05
11	1,1088E+03	2,9153E+00	1,1080E+03	2,4020E+00	1,2198E+03	6,7179E+01	***1,1067E+03***	1,9652E+00	1,1074E+03	2,6814E+00
12	1,4620E+03	1,1574E+02	1,3995E+03	9,1615E+01	2,1953E+03	4,3894E+02	1,4517E+03	9,5565E+01	***1,3983E+03***	1,0221E+02
13	1,6415E+03	1,9141E+01	1,6437E+03	2,9519E+01	3,0005E+03	9,8971E+02	***1,6333E+03***	2,5959E+01	1,6452E+03	2,8341E+01
14	1,6076E+03	4,5254E+00	1,6111E+03	3,5980E+00	1,6770E+03	4,2292E+01	1,6053E+03	5,0554E+00	***1,6021E+03***	5,8221E+00
15	1,9149E+03	1,4570E+02	1,9269E+03	7,4514E+01	2,1840E+03	1,0567E+02	***1,8365E+03***	1,8650E+02	1,9233E+03	1,0398E+02
30D										
1	3,9049E+09	1,6017E+09	2,8899E+10	5,8485E+09	9,3283E+10	1,2799E+10	4,7539E+09	1,3400E+09	***1,1795E+09***	6,2432E+08
2	9,9760E+04	2,5812E+04	1,3418E+05	2,6344E+04	6,9430E+06	1,4507E+07	9,7376E+04	2,0814E+04	***8,5653E+04***	2,0924E+04
3	3,3113E+02	2,8347E+00	3,3850E+02	1,9604E+00	3,4771E+02	1,9232E+00	3,3059E+02	2,6174E+00	***3,2638E+02***	4,1408E+00
4	7,7928E+03	5,2384E+02	8,0330E+03	4,2608E+02	9,6696E+03	4,0262E+02	6,8199E+03	8,5746E+02	***5,1202E+03***	6,7938E+02
5	5,0418E+02	7,2160E−01	5,0425E+02	5,1916E−01	5,0586E+02	1,1990E+00	5,0422E+02	6,1627E−01	***5,0410E+02***	8,1315E−01
6	6,0096E+02	1,8792E−01	6,0410E+02	2,2901E−01	6,0755E+02	6,6978E−01	6,0090E+02	1,3068E−01	***6,0076E+02***	9,5338E−02
7	7,0405E+02	2,0114E+00	7,6997E+02	1,2225E+01	8,9401E+02	2,6905E+01	7,0750E+02	4,2654E+00	***7,0074E+02***	3,3250E−01
8	4,0013E+03	3,3477E+03	2,3491E+06	1,4471E+06	1,6032E+08	1,1620E+08	1,7191E+04	3,3261E+04	***2,6354E+03***	4,4907E+03
9	9,1358E+02	2,3378E−01	9,1381E+02	2,8076E−01	9,1427E+02	1,8457E−01	9,1365E+02	3,1664E−01	***9,1337E+02***	2,2728E−01
10	7,5602E+06	3,1209E+06	2,9938E+07	1,4513E+07	3,9391E+08	1,6830E+08	1,1337E+07	6,3462E+06	***5,4690E+06***	3,3465E+06
11	1,1614E+03	4,1043E+01	1,2889E+03	3,7640E+01	2,1094E+03	4,6591E+02	1,1551E+03	2,8498E+01	***1,1336E+03***	2,2415E+01
12	2,2417E+03	1,8647E+02	2,9655E+03	2,7960E+02	4,9876E+05	6,1606E+05	2,0617E+03	2,0746E+02	***1,7752E+03***	1,4548E+02
13	1,7719E+03	2,6527E+01	1,9596E+03	8,1067E+01	3,6329E+03	7,7365E+02	1,7390E+03	2,3388E+01	***1,6866E+03***	1,6524E+01
14	1,6644E+03	1,8615E+01	1,7522E+03	2,6072E+01	2,1683E+03	1,4580E+02	1,6711E+03	2,8498E+01	***1,6469E+03***	9,7379E+00
15	2,5522E+03	1,7142E+02	2,8256E+03	5,5492E+01	3,9013E+03	4,4677E+02	2,6529E+03	1,6852E+02	***2,4467E+03***	2,1048E+02

10D and 30D, respectively. The best results are highlighted with italic and shown in the tables.

Table 4 orders the algorithms according to their final mean optimized output values. According to Table 3, the proposed HFPSO is best on all 30D problems and it has the first ranks, thus average rank value of HFPSO in 30D problems is 1,00. Thus, the final rank of proposed HFPSO is the first. According to their ranks, following second, third, fourth, and fifth algorithms are HPSOFF, PSO, FA, FFPSO, respectively. HPSOFF has second rank in 10D problems, but in 30D problems, PSO surpasses it and replaces with HPSOFF because of less success in hybrid and composition functions.

Convergence graphs versus function evaluations for 30D problems are shown in Fig. 5. Graphs are drawn by corresponding mean values of 20 independent run results. Mostly, HFPSO converges towards the better optima point with a faster rate of convergence than other algorithms. When graph of Fig. 5 is analyzed, it is seen that HFPSO generally tracks PSO convergence line and uses FA abilities. Therefore, exploration and exploitation are done in a combination and better diversity of particles are provided with this combination.

HFPSO is also compared with recent PSO variants and other optimization approaches. EPSO is a recent improved PSO variant paper [4]. EPSO is compared with standard PSO, other successful PSO variants and optimizers. It is showed in EPSO paper that EPSO outperforms the rest in CEC 2015 problems. ISRPSO [2] is an improved variant of SRPSO algorithm [9]. ISRPSO is also a participant of CEC 2015 competition. It is improved version of standard PSO [1] algorithm.

In Table 5, optimization results of Differential Evolution (DE), Evolutionary Strategy (ES), Covariance Matrix Adaptation Evolution Strategy (CMAES-S, CMAES-G), EPSO, ISRPSO are shown and all results are directly extracted from related papers [2,4]. In Table 5,

Table 4
Ranks of CEC 2015 results.

CEC2015	D	PSO	FA	FFPSO	HPSOFF	HFPSO
F1	10D	3	4	5	2	1
	30D	2	4	5	3	1
F2	10D	2	1	5	3	4
	30D	3	4	5	2	1
F3	10D	3	2	5	4	1
	30D	3	4	5	2	1
F4	10D	4	2	5	3	1
	30D	3	4	5	2	1
F5	10D	3	4	5	2	1
	30D	2	4	5	3	1
F6	10D	3	4	5	2	1
	30D	3	4	5	2	1
F7	10D	3	4	5	2	1
	30D	2	4	5	3	1
F8	10D	3	4	5	1	2
	30D	2	4	5	3	1
F9	10D	2	4	5	1	3
	30D	2	4	5	3	1
F10	10D	1	4	5	3	2
	30D	2	4	5	3	1
F11	10D	4	3	5	1	2
	30D	3	4	5	2	1
F12	10D	4	2	5	3	1
	30D	3	4	5	2	1
F13	10D	2	3	5	1	4
	30D	3	4	5	2	1
F14	10D	3	4	5	2	1
	30D	2	4	5	3	1
F15	10D	2	4	5	1	3
	30D	2	4	5	3	1
Avg. Rank	10D	**2,80**	**3,27**	**5,00**	**2,07**	***1,87***
	30D	**2,47**	**4,00**	**5,00**	**2,53**	***1,00***
Rank	10D	**3**	**4**	**5**	**2**	***1***
	30D	**2**	**4**	**5**	**3**	***1***

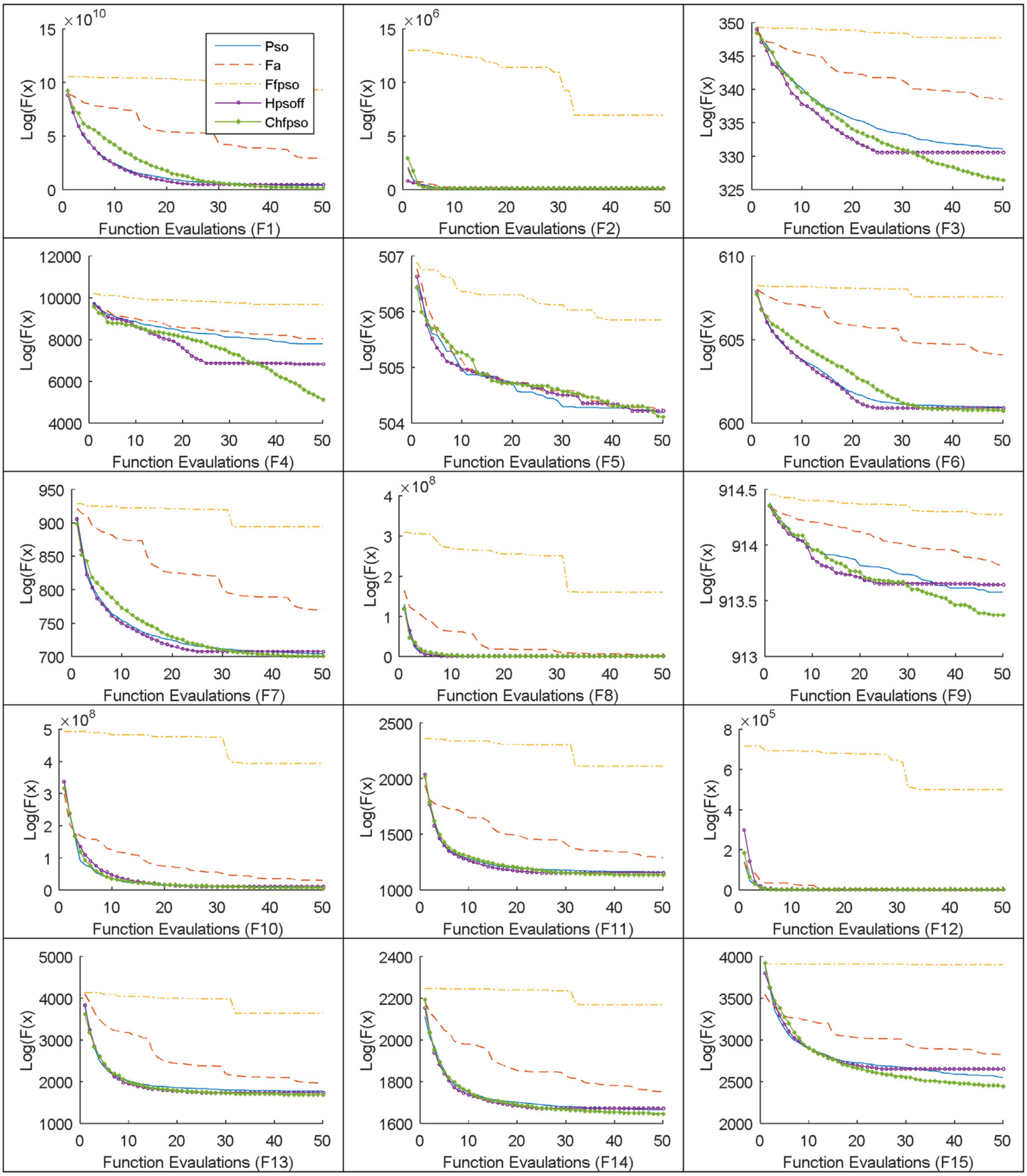


Fig. 5. CEC 2015 30D convergence graphs.

it is seen that HFPSO is better than other PSO variants and optimizers especially in hybrid and composition function categories. HFPSO can find the lowest mean minimization values in 8 out of 15 problems for 30D problems compared to other papers' different algorithm results.

In Table 6, CEC 2017 benchmarks set and 10 dimensional (10D) problems' final mean optimized output values are shown. According to the table, HFPSO can find the lowest mean minimization values in 14 out of 30 problems for 10D. Firefly algorithm (FA) slightly surpasses the proposed HFPSO in 11 problems out of 30 problems. Rotated and shifted CEC 2017 problems are more suitable for firefly than particle swarm. Firefly does not have velocity and some other parameters thus can escape local optima rather than PSO in CEC 2017 10D problems. CEC 2017 problem set contains 15 more problems than CEC 2015 problem set. Average rank of proposed HFPSO is 1,90 in CEC 2017 10D and this result is 1,87 in CEC 2015. Especially, the average rank is increased because of the hybrid function category results. Additionally, PSO has the best mean minimization output values in F9 and F28. HPSOFF is better in F1, F12, F15, F22, F25 functions than other algorithms. FFPSO has no success in these functions with limited allocated budgets. The best results are highlighted with italic and shown in the tables.

In Fig. 6, some CEC 2017 10D results of HFPSO are shown in box plots. Each box contains a central median value, outliers, and the 25th and 75th percentiles of the 20 independent run results. Box plots indicate that HFPSO results are stable and generally have low standard deviation values. Except some outliers, it can be inferred from F1, F12, F15, F18, F19 function box plot graphs that proposed HFPSO converges generally around the same points. F22, F23, F25 have short boxes indicating that the obtained results have low

Table 5
CEC 2015 30D HFPSO vs. other optimization algorithms.

CEC2015	DE	(μ + λ)-ES	CMAES-S	CMAES-G	EPSO	ISRPSO	HFPSO
F1	2,3911E + 10	3,5775E + 10	***6,8700E + 07***	1,1080E + 08	8,4866E + 09	7,1910E + 08	1,1795E + 09
F2	1,8254E + 05	1,6179E + 05	2,3630E + 05	2,9530E + 05	***6,3748E + 04***	7,6860E + 04	8,5653E + 04
F3	3,4190E + 02	3,4353E + 02	6,3390E + 02	6,5270E + 02	3,3800E + 02	***3,2569E + 02***	3,2638E + 02
F4	7,9627E + 03	7,0557E + 03	8,6730E + 03	1,2040E + 04	6,6946E + 03	5,8090E + 03	***5,1202E + 03***
F5	5,0431E + 02	5,0499E + 02	1,0010E + 03	1,0080E + 03	5,0430E + 02	5,0424E + 02	***5,0410E + 02***
F6	6,0365E + 02	6,0433E + 02	1,2010E + 03	1,2010E + 03	6,0276E + 02	***6,0064E + 02***	6,0076E + 02
F7	7,5438E + 02	7,8216E + 02	1,4010E + 03	1,4010E + 03	7,2189E + 02	***7,0057E + 02***	7,0074E + 02
F8	7,9963E + 05	7,3789E + 06	1,7670E + 03	2,3210E + 03	1,2746E + 05	***1,4262E + 03***	2,6354E + 03
F9	9,1394E + 02	9,1408E + 02	1,8270E + 03	1,8280E + 03	9,1372E + 02	9,1357E + 02	***9,1337E + 02***
F10	3,8759E + 07	9,5323E + 07	***3,6310E + 06***	1,4730E + 07	2,6363E + 07	6,8320E + 06	5,4690E + 06
F11	1,2870E + 03	1,4378E + 03	2,2460E + 03	2,2580E + 03	1,2288E + 03	1,1509E + 03	***1,1336E + 03***
F12	3,0110E + 03	3,8087E + 03	3,4540E + 03	4,0940E + 03	2,4432E + 03	1,9357E + 03	***1,7752E + 03***
F13	1,9613E + 03	2,2208E + 03	3,3840E + 03	3,4260E + 03	1,8839E + 03	1,6996E + 03	***1,6866E + 03***
F14	1,7479E + 03	1,8406E + 03	3,2660E + 03	3,3000E + 03	1,7016E + 03	1,6655E + 03	***1,6469E + 03***
F15	2,9304E + 03	2,9154E + 03	4,4270E + 03	4,8360E + 03	2,7488E + 03	2,4510E + 03	***2,4467E + 03***
Avg. Rank	**4,60**	**5,27**	**5,00**	**6,07**	**3,27**	**1,87**	***1,80***
Rank	**4**	**6**	**5**	**7**	**3**	**2**	***1***

Table 6
CEC 2017 10D results summary.

	PSO		FA		FFPSO		HPSOFF		HFPSO	
Fn.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
1	3,2165E + 08	2,3861E + 08	1,6258E + 09	1,2694E + 09	2,2812E + 10	7,5101E + 09	***1,7345E + 08***	2,8025E + 08	2,4541E + 08	9,8060E + 08
2	4,6348E + 08	7,5312E + 08	3,3699E + 11	1,4964E + 12	1,8124E + 16	3,4626E + 16	1,8320E + 08	3,2778E + 08	***1,4746E + 08***	4,9126E + 08
3	1,7908E + 04	4,9711E + 03	3,0383E + 04	2,0969E + 04	8,9010E + 07	1,8367E + 08	2,2767E + 04	1,0668E + 04	***1,1744E + 04***	5,9568E + 03
4	5,4787E + 02	3,2899E + 02	5,1021E + 02	5,6927E + 01	3,3173E + 03	1,6661E + 03	4,5895E + 02	4,3844E + 01	***4,4101E + 02***	4,5452E + 01
5	5,7248E + 02	1,6763E + 01	5,6142E + 02	1,8732E + 01	6,7964E + 02	2,4753E + 01	5,6661E + 02	1,7223E + 01	***5,5292E + 02***	1,8402E + 01
6	6,3032E + 02	1,5130E + 01	***6,1687E + 02***	5,2386E + 00	6,8447E + 02	1,7108E + 01	6,2494E + 02	9,6435E + 00	6,2194E + 02	1,3521E + 01
7	7,8146E + 02	1,1902E + 01	7,9202E + 02	2,2557E + 01	1,0146E + 03	6,1461E + 01	7,8107E + 02	1,5138E + 01	***7,6343E + 02***	1,7263E + 01
8	8,6055E + 02	9,0266E + 00	8,5822E + 02	1,8953E + 01	9,3140E + 02	2,0012E + 01	8,5221E + 02	1,3544E + 01	***8,3952E + 02***	1,4438E + 01
9	***1,1099E + 03***	2,2269E + 02	1,1843E + 03	1,7125E + 02	3,6784E + 03	8,9314E + 02	1,2238E + 03	2,7506E + 02	1,1775E + 03	3,0667E + 02
10	3,0080E + 03	2,9923E + 02	2,6285E + 03	3,2258E + 02	3,8614E + 03	2,8731E + 02	2,6270E + 03	3,7715E + 02	***2,2670E + 03***	3,7933E + 02
11	1,6759E + 03	9,9140E + 02	3,5406E + 03	4,9132E + 03	6,4221E + 04	1,6948E + 05	1,2625E + 03	1,2696E + 02	***1,1959E + 03***	5,2380E + 01
12	1,8611E + 07	8,8766E + 06	7,7087E + 06	6,2420E + 06	3,0343E + 09	1,5253E + 09	***2,8243E + 06***	3,0772E + 06	2,8998E + 06	4,1307E + 06
13	4,9528E + 05	3,9492E + 05	1,8233E + 04	2,0474E + 04	6,5693E + 08	6,4239E + 08	1,4573E + 04	7,4206E + 03	***1,1835E + 04***	7,6845E + 03
14	7,6428E + 03	7,0146E + 03	1,0411E + 04	9,2273E + 03	8,5002E + 05	1,0892E + 06	5,0414E + 03	3,4439E + 03	***4,6478E + 03***	4,1776E + 03
15	4,8266E + 04	4,6631E + 04	1,8770E + 04	1,3651E + 04	5,9980E + 07	9,8559E + 07	***1,3609E + 04***	1,1050E + 04	2,2845E + 04	2,3650E + 04
16	2,1201E + 03	1,9609E + 02	1,9873E + 03	1,5647E + 02	2,8816E + 03	3,1814E + 02	2,0516E + 03	1,5269E + 02	***1,9629E + 03***	1,5877E + 02
17	1,8650E + 03	5,9215E + 01	***1,8035E + 03***	5,1924E + 01	2,4780E + 03	2,7983E + 02	1,8217E + 03	4,6836E + 01	1,8248E + 03	8,3997E + 01
18	7,0533E + 05	8,2307E + 05	***2,1706E + 04***	1,6198E + 04	2,0293E + 09	1,8875E + 09	8,1675E + 04	1,9730E + 05	2,9732E + 04	1,7941E + 04
19	3,5868E + 04	2,5999E + 04	***1,4600E + 04***	1,2364E + 04	2,4345E + 07	4,1842E + 07	1,8030E + 04	9,0940E + 03	3,5177E + 04	3,8300E + 04
20	2,2668E + 03	1,1628E + 02	***2,1498E + 03***	5,6944E + 01	2,5686E + 03	1,6145E + 02	2,1834E + 03	9,3493E + 01	2,2023E + 03	1,0753E + 02
21	2,3553E + 03	3,7009E + 01	2,3532E + 03	1,0085E + 01	2,4638E + 03	3,5194E + 01	2,3488E + 03	4,6622E + 01	***2,3377E + 03***	4,7750E + 01
22	2,6684E + 03	7,7654E + 02	2,4297E + 03	1,9109E + 02	4,3663E + 03	6,4636E + 02	***2,4196E + 03***	2,0260E + 02	2,5975E + 03	5,8755E + 02
23	2,6906E + 03	3,4234E + 01	***2,6656E + 03***	1,2682E + 01	3,0293E + 03	2,0917E + 02	2,6783E + 03	2,3378E + 01	2,6757E + 03	2,8668E + 01
24	2,7972E + 03	9,4842E + 01	2,7778E + 03	6,4001E + 01	3,1865E + 03	1,5208E + 02	2,7337E + 03	1,1168E + 02	***2,7128E + 03***	1,4672E + 02
25	2,9648E + 03	1,5755E + 01	2,9868E + 03	4,3648E + 01	4,4005E + 03	5,4274E + 02	***2,9515E + 03***	1,8425E + 01	2,9581E + 03	5,0193E + 01
26	3,4545E + 03	6,3256E + 02	3,5102E + 03	4,3450E + 02	5,0797E + 03	4,2029E + 02	3,4471E + 03	5,3585E + 02	***3,2147E + 03***	3,4227E + 02
27	3,1529E + 03	3,9793E + 01	***3,1221E + 03***	1,9657E + 01	3,7507E + 03	2,6732E + 02	3,1356E + 03	3,3564E + 01	3,1538E + 03	3,9370E + 01
28	***3,3450E + 03***	1,0498E + 02	3,4768E + 03	1,8028E + 02	4,3603E + 03	2,8037E + 02	3,4064E + 03	9,4585E + 01	3,4422E + 03	1,0794E + 02
29	3,4299E + 03	8,8769E + 01	***3,3299E + 03***	6,6574E + 01	4,1052E + 03	3,2667E + 02	3,3721E + 03	1,0183E + 02	3,3864E + 03	9,3976E + 01
30	4,1456E + 06	4,2666E + 06	***2,6353E + 06***	2,6696E + 06	3,0366E + 08	2,4690E + 08	3,6005E + 06	2,9950E + 06	4,2723E + 06	3,7549E + 06
Avg. Rank	**3,43**		**2,57**		**5,00**		**2,10**		***1,90***	
Rank	**4**		**3**		**5**		**2**		***1***	

volatilities. Rest of the functions show some unbalanced results. For example, F9 has a long box, still its central line median value is low. On the other hand, F20 has also long result distributions, but its median is a bit higher.

CEC 2017 problems with 30D results are shown in Table 7. According to the table, HFPSO can find the lowest mean minimization values in 28 out of 30 problems for 30D. Average rank of HFPSO, which is 1,00 in CEC 2015 30D, is 1,07 here. Only standard PSO and HPSOFF results are better than HFPSO in simple multimodal F9 and composition F22 functions, respectively. Number of particles is 10 and 30 on 10D and 30D problems respectively. Therefore, using proper particle size in the proposed HFPSO algorithm can affect CEC problem results.

In Fig. 7, mean runtime of algorithms versus the function numbers are shown in milliseconds. In all 10D, 30D CEC 2015 and CEC 2017 problems, HFPSO has the short runtimes as competitive with PSO. 30D problems' runtimes are about 3 times higher than 10D problem runtimes. Runtimes are comparatively increased against hybrid and composition functions, which are more complex than other functions. It is clear that, HPSOFF has about half of PSO and FA algorithm values. Thus, HPSOFF has third minimum mean runtime values in milliseconds, FFPSO is fourth, and the last rank of runtime belongs to FA. By examining the graph, it can be said that proposed HFPSO combination uses particle swarm algorithm mechanisms more and if it is necessary, a local search or particle fine tuning process starts, then firefly algorithm is used in that way.

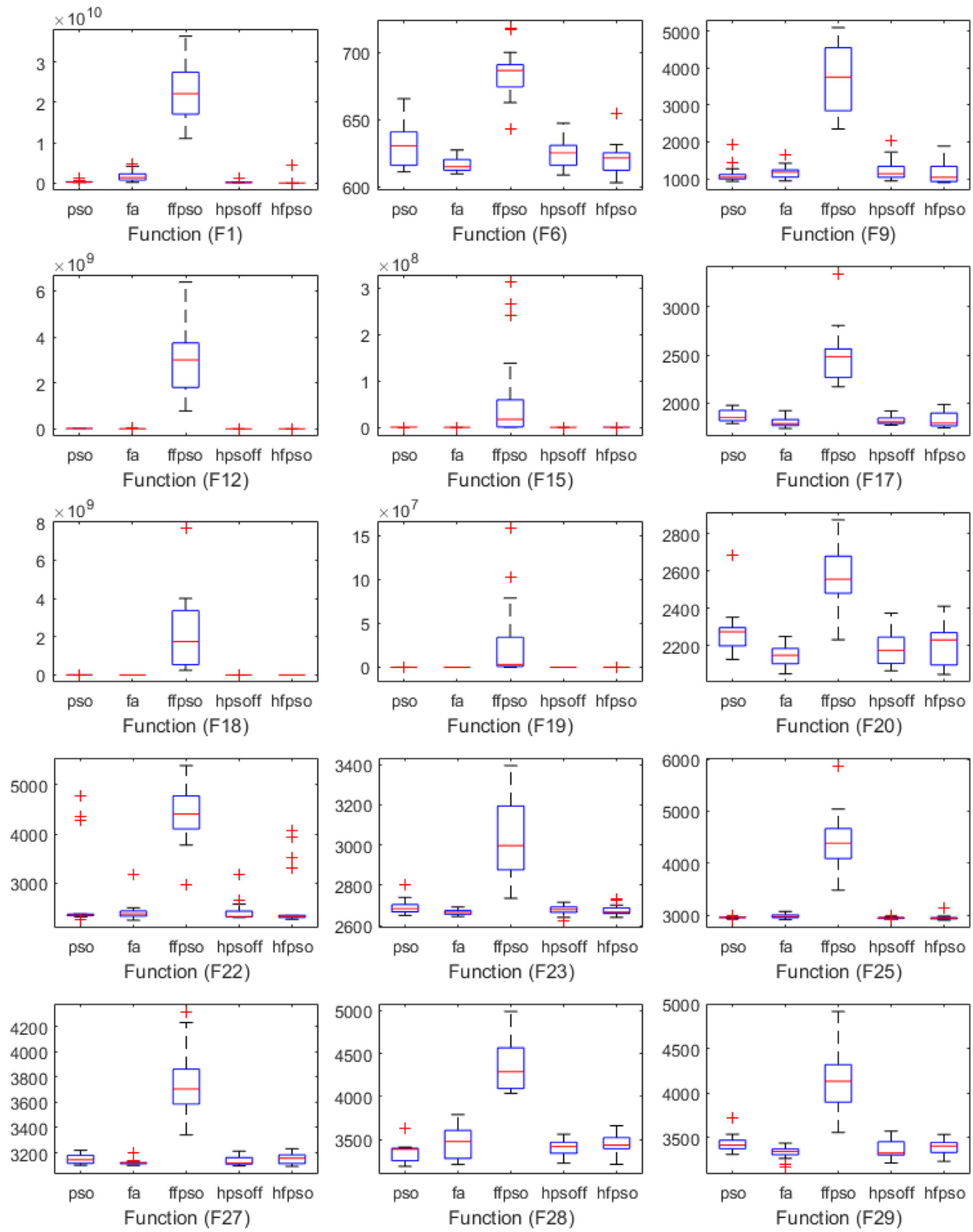


Fig. 6. Boxplot graphs.

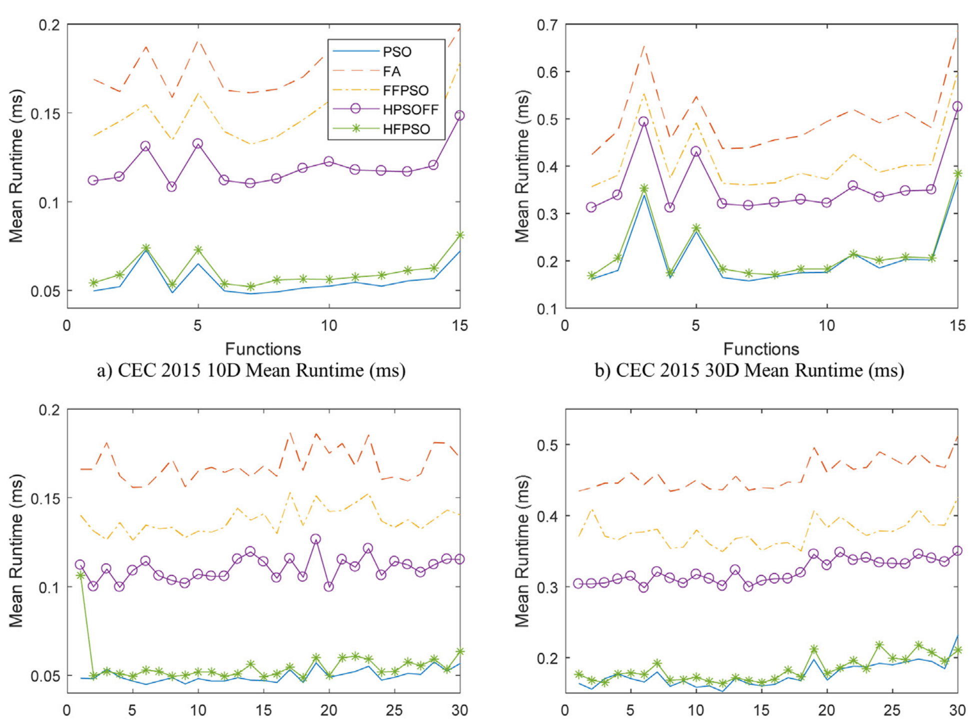
Fig. 8 shows the results of 10D, 30D, CEC 2015 and CEC 2017 the grand total mean fitness values for unimodal, simple multimodal, hybrid, and composition functions based on algorithms. The lower values indicate better minimization output values. According to the figure, in unimodal functions, HFPSO has a much better comparative ratio than the others. Besides, in summary, in simple multimodal, hybrid, and composition function categories, HFPSO outperforms PSO and FA algorithms and other hybrid approaches. In the graph, it can be seen that FFPSO algorithm is not suitable for unimodal functions.

Our proposed HFPSO algorithm combines ability and capability of particle swarm and firefly optimization algorithms. A balanced combination between local and global search ability is achieved. Therefore, the advantages of both algorithms are utilized in a proper hybridization.

The updated fast convergence ability and increased particle diversity in the hybrid FF and PSO (FFPSO) algorithm are obtained through a FA inserted into PSO to avoiding to get trapped in local minima. Difference between FA and FFPSO is: the light attraction of each particle is mutated by a PSO operator. During the calcula-

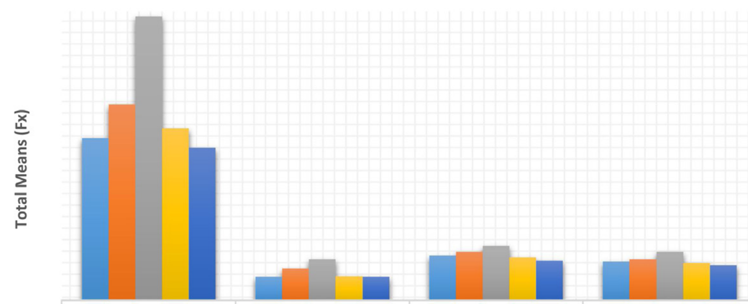
Table 7
CEC 2017 30D results summary.

	PSO		FA		FFPSO		HPSOFF		HFPSO	
Fn.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.	Mean	Std.
1	5,5573E+09	3,3893E+09	3,3061E+10	4,4031E+09	1,1348E+11	1,2767E+10	6,0544E+09	2,1716E+09	***1,4877E+09***	8,9773E+08
2	3,9099E+32	1,6788E+33	1,8379E+39	3,6984E+39	1,0418E+57	4,6498E+57	3,1775E+34	1,0461E+35	***4,2895E+30***	1,3632E+31
3	1,3257E+05	3,4533E+04	2,1556E+05	4,4961E+04	6,7723E+09	1,9830E+10	1,3635E+05	3,6125E+04	***1,3159E+05***	5,0828E+04
4	9,0860E+02	1,7332E+02	6,8240E+03	1,7720E+03	4,4924E+04	1,0769E+04	1,2583E+03	3,3792E+02	***6,8099E+02***	7,2977E+01
5	8,0387E+02	2,0464E+01	8,7976E+02	2,1895E+01	1,1284E+03	6,7520E+01	8,1363E+02	3,4695E+01	***7,4431E+02***	2,8310E+01
6	6,5628E+02	1,3442E+01	6,7297E+02	6,4935E+00	7,2989E+02	1,2755E+01	6,7252E+02	8,5569E+00	***6,5458E+02***	1,4862E+01
7	1,1273E+03	3,0590E+01	1,7985E+03	1,6329E+02	2,9927E+03	2,0330E+02	1,1241E+03	5,3149E+01	***1,0634E+03***	3,8217E+01
8	1,0857E+03	2,1442E+01	1,1591E+03	3,0228E+01	1,3608E+03	3,8288E+01	1,0756E+03	3,0109E+01	***1,0174E+03***	3,4938E+01
9	***6,6038E+03***	2,5645E+03	1,3053E+04	1,6887E+03	3,0274E+04	3,4821E+03	9,0601E+03	3,0384E+03	9,0430E+03	2,4238E+03
10	9,2920E+03	3,1888E+02	9,4664E+03	4,8533E+02	1,0715E+04	4,5678E+02	8,9099E+03	5,0987E+02	***7,4962E+03***	9,1080E+02
11	3,6404E+03	9,2311E+02	1,1963E+04	3,7161E+03	3,4403E+04	1,4283E+04	4,5469E+03	1,0526E+03	***2,2880E+03***	6,8090E+02
12	6,4666E+08	3,0524E+08	3,0340E+09	9,0341E+08	2,7727E+10	5,6036E+09	3,3040E+08	2,2407E+08	***5,9479E+07***	3,7560E+07
13	1,9336E+08	5,8827E+07	1,1545E+09	6,1763E+08	2,7922E+10	1,2029E+10	2,7671E+07	1,9963E+07	***5,0687E+06***	1,5852E+07
14	1,1117E+06	7,8687E+05	1,8933E+06	1,8783E+06	5,6035E+07	3,5343E+07	1,1120E+06	1,1345E+06	***6,6292E+05***	9,6993E+05
15	3,9118E+07	1,8225E+07	1,0725E+08	6,1741E+07	7,1411E+09	3,1801E+09	1,5676E+06	1,1456E+06	***7,3512E+04***	2,3482E+04
16	3,9514E+03	4,2128E+02	4,4164E+03	2,9240E+02	9,6352E+03	2,6750E+03	3,6383E+03	4,0419E+02	***3,1951E+03***	4,2601E+02
17	2,6556E+03	2,3445E+02	3,0044E+03	1,8143E+02	3,3347E+04	4,9223E+04	2,5254E+03	2,5071E+02	***2,2984E+03***	2,1441E+02
18	7,9043E+06	6,2705E+06	2,9281E+07	1,9063E+07	1,0554E+09	7,7432E+08	8,5589E+06	9,7364E+06	***2,5984E+06***	2,1944E+06
19	5,1700E+07	2,9419E+07	1,6613E+08	6,2094E+07	8,2628E+09	3,4065E+09	8,4341E+06	9,5782E+06	***5,9487E+05***	5,7691E+05
20	3,0177E+03	2,1693E+02	3,1126E+03	1,3585E+02	3,7975E+03	2,1617E+02	2,9083E+03	2,5178E+02	***2,7685E+03***	1,8760E+02
21	2,6009E+03	2,2044E+01	2,6405E+03	2,6874E+01	2,8880E+03	6,5735E+01	2,5982E+03	5,7252E+01	***2,5087E+03***	2,9204E+01
22	6,8425E+03	3,8923E+03	8,0003E+03	1,8365E+03	1,2117E+04	3,2675E+02	***5,7433E+03***	3,1589E+03	5,7966E+03	3,2553E+03
23	3,0819E+03	1,0238E+02	3,1542E+03	4,9848E+01	4,0749E+03	2,8961E+02	3,0969E+03	9,0742E+01	***2,9597E+03***	*7,4124E+01*
24	3,2139E+03	7,7916E+01	3,3056E+03	5,1836E+01	4,4529E+03	2,5580E+02	3,2719E+03	9,8133E+01	***3,1742E+03***	*1,0577E+02*
25	3,2145E+03	6,7870E+01	4,9480E+03	4,0432E+02	1,5958E+04	3,1469E+03	3,2320E+03	1,4716E+02	***3,0623E+03***	*7,0241E+01*
26	6,6461E+03	1,1813E+03	8,8041E+03	6,0012E+02	1,6163E+04	1,5021E+03	7,2386E+03	1,6475E+03	***5,6590E+03***	*1,5232E+03*
27	3,3573E+03	8,5636E+01	3,6059E+03	6,7467E+01	5,8099E+03	7,3079E+02	3,4360E+03	9,9726E+01	***3,3243E+03***	*6,1360E+01*
28	3,6388E+03	1,2870E+02	5,6940E+03	4,5852E+02	1,3535E+04	1,8714E+03	3,7938E+03	2,4230E+02	***3,3925E+03***	*5,6285E+01*
29	4,9240E+03	3,1984E+02	5,7051E+03	3,5852E+02	9,3924E+04	1,0965E+05	4,8406E+03	4,0875E+02	***4,4249E+03***	*2,5350E+02*
30	4,8514E+07	1,7476E+07	1,7163E+08	6,1288E+07	4,1425E+09	1,5608E+09	2,5749E+07	1,9168E+07	***4,6082E+06***	*2,7679E+06*
Avg. Rank	**2,43**		**4,00**		**5,00**		**2,50**		***1,07***	
Rank	**2**		**4**		**5**		**3**		***1***	



a) CEC 2015 10D Mean Runtime (ms)
b) CEC 2015 30D Mean Runtime (ms)
c) CEC 2017 10D Mean Runtime (ms)
d) CEC 2017 30D Mean Runtime (ms)

Fig. 7. Runtime comparison graphs.



	Unimodal	Simple Multimodal	Hybrid	Composition
PSO	3.91E+32	4.89E+04	9.68E+08	5.27E+07
FA	1.84E+39	2.41E+06	4.53E+09	1.74E+08
FFPSO	1.04E+57	1.61E+08	7.84E+10	4.45E+09
HPSOFF	3.18E+34	6.26E+04	3.92E+08	2.94E+07
HFPSO	4.29E+30	4.33E+04	7.73E+07	8.95E+06

Fig. 8. Categorical sum of function mean values.

tion of position vector, distance to personal best (*pbest*) and global best (*gbest*) are considered. Hence, local search is occurred via the modified attractiveness step and particles are randomly attracted global best [44]. The main goal of FFPSO is feature selection before training process of artificial neural network.

Hybrid PSO-FFA (HPSOFF) algorithm combines the fast calculation advantage of PSO with robustness advantage of FA in order to get an increased global search ability. PSO starts with a set of random initial population. Until a termination criterion is reached, the population is promoted and moves towards the best personal and global particles. At the end of iterations, the optimal solution is given as outputs. Afterwards, FA uses these outputs as input initial population and in every iteration, FA improves and promotes the population obtained from PSO [43]. Hence, optimal solution obtained from HPSOFF algorithm depends on both standard PSO and FA algorithms and the quality of initial solution. Main objective of HPSOFF is to provide a new and better solution approach for Combined Economic and Emission Dispatch (CEED) problem, which is formulated as multi-objective optimization problem.

7.3. Engineering and mechanical design benchmark problems

In order to see how the proposed algorithm performs in realistic optimization problems, three engineering and mechanical design benchmark problems are used. These problems are widely used in the literature and mathematical equations and descriptions can be attained by all researchers. In Tables 8–10, the results of the other approaches and experiment settings are presented: particle sizes and maximum iterations are directly obtained from related papers [4,56]. After 100 independent runs, the results are statistically given in the tables.

The first design problem is suggested by Kannan and Kramer and named as pressure vessel design problem [57]. Fitness function of pressure vessel design problem finds minimized total cost, including cost of material, welding and giving shapes [4]. Our proposed HFPSO method finds one of the minimum bests 5885.3328 result at (0.778169, 0.384649, 40.319619, 200) point in 100.000 function evaluations (FES). As it is seen in Table 8, the proposed HFPSO also has the competitive mean result in pressure vessel engineering design problem with 5,000 FES.

The second problem is welded beam design problem proposed by Coello [58]. This problem has been designed to find minimized cost subject to constraints on shear, end deflection of beam, tension tendency in beam, buckling load on bar, and also side constraints [4]. In Table 9, it is given that HFPSO can find one of the best results and also has competitive mean result in welded beam design problem against other algorithm.

The last problem is the tension and compression spring design problem proposed by Arora [59]. Fitness function aims to minimize a tension and compression spring weight subject to constraints some technical design parameters and variables [4]. As seen in Table 10, HFPSO can also be used to find this kind of engineering design problems. Under different function evaluations (FES), the results are stable and robust in tension and compression spring design problem in the behalf of mean and best results.

7.4. Statistical test

In order to indicate whether differences in performance between the algorithms are statistically significant or not, the Holm–Bonferroni procedure, see [60,61], is employed. The statistical procedure is adopted from [62,63]. For this purpose, related 5 algorithms and 2 problem sets containing 15 and 30 problems are used and the tests are performed. Two hypotheses are taken into consideration:

- Null hypothesis (H_0): There is no significant difference between performances of two algorithms. If the null hypothesis is accepted, $h = 0$ is donated.
- Alternative hypothesis (H_1): There is a significant difference between performances of two algorithms. Thus, null hypothesis is rejected and $h = 1$ is donated.

In detail, in Holm–Bonferroni procedure, a score (S_i) ($i = 1,\ldots,N_A$), N_A is the number of compared algorithms, $N_A = 5$ in our paper. In order to calculate score (S_i), a rank score R_i, ($i = 1,\ldots,N_A$) is assigned. For example, the algorithm which has the best performance is assigned a rank score of 5, next, to the second best, a rank score of 4 is given, thus scoring continues until the worst performance algorithm which is assigned a rank score of 1, the lowest. At final, scores (S_i) of each algorithm are get from problems by the mean of the rank scores (R_i). The z values of related algorithms are estimated by Eq. (14). Here, S_0 represents the score of proposed HFPSO algorithm used as a comparing reference score in the equation. The cumulative normal distribution (p) values are calculated with using of z values. Obtained P values are compared with the threshold: θ, δ is the level of confidence and used as 0.05 in our tests, $\theta = \delta/(N_A - 1)$. If "$p < \theta$" then the null hypothesis is rejected, alternative hypothesis

Table 8
Pressure vessel design problem.

Method	Worst	Mean	Best	Std.	FES
GA3	6308.4970	6293.8432	6288.7445	7.41E + 00	900,000
GA4	6469.3220	6177.2533	6059.9463	1.30E + 02	80,000
CPSO	6363.8041	6147.1332	6061.0777	8.64E + 01	240,000
HPSO	6288.6770	6099.9323	6059.7143	8.62E + 01	81,000
NM-PSO	5960.0557	5946.7901	5930.3137	9.16E + 00	80,000
G-QPSO	7544.4925	6440.3786	6059.7208	4.48E + 02	8000
QPSO	8017.2816	6440.3786	6059.7209	4.79E + 02	8000
PSO	14076.324	8756.6803	6693.7212	1.49E + 03	8000
CDE	6371.0455	6085.2303	6059.7340	4.30E + 01	204,800
UPSO	N/A	9032.5500	6544.2700	9.95E + 02	100,000
PSO-DE	N/A	6059.7140	6059.7140	N/A	42,100
ABC	N/A	6245.3080	6059.7140	2.05E + 02	30,000
(μ + λ)-ES	N/A	6379.9380	6059.7016	2.10E + 02	30,000
TLBO	N/A	6059.7143	6059.7143	N/A	10,000
MBA	6392.5062	6200.6477	5889.3216	1.60E + 02	70,650
EPSO	7315.6752	6254.1804	5885.3383	4.24E + 02	10,000
	6076.6205	***5920.8442***	***5885.3328***	5.21E + 01	100,000
HFPSO	6940.8249	6140.5905	5885.3329	1.97E + 02	8000
	6505.2363	6080.0462	***5885.3328***	1.34E + 02	100,000

Table 9
Welded beam design problem.

Method	Worst	Mean	Best	Std.	FES
GA3	1.785835	1.771973	1.748309	1.12E – 02	900,000
GA4	1.993408	1.792654	1.728226	7.47E – 02	80,000
CAEP	3.179709	1.971809	1.724852	4.43E – 01	50,020
CPSO	1.782143	1.748831	1.728024	1.29E – 02	240,000
HPSO	1.814295	1.749040	1.724852	4.01E – 02	81,000
PSO-DE	1.724852	***1.724852***	1.724852	6.70E – 16	66,600
NM-PSO	1.733393	1.726373	***1.724717***	3.50E – 03	80,000
MGA	1.995000	1.919000	1.824500	5.37E – 02	N/A
SC	6.399679	3.002588	2.385435	9.60E – 01	33,095
DE	1.824105	1.768158	1.733461	2.21E – 02	204,800
UPSO	N/A	2.837210	1.921990	6.83E – 01	100,000
CDE	N/A	1.768150	1.733460	N/A	240,000
(μ + λ)-ES	N/A	1.777692	1.724852	8.8E – 02	30,000
ABC	N/A	1.741913	1.724852	3.1E – 02	30,000
TLBO	N/A	1.728447	1.724852	N/A	10,000
MBA	1.724853	1.724853	1.724853	6.94E – 19	47,340
EPSO	1.747220	1.728219	1.724853	5.62E – 03	50,000
HFPSO	1.983895	1.743780	1.724852	5.60E – 02	10,000
	1.974449	1.727370	1.724852	2.50E – 02	80,000

Table 10
Tension and compression spring design problem.

Method	Worst	Mean	Best	Std.	FES
GA3	0.012822	0.012769	0.012705	3.94E – 05	900,000
GA4	0.012973	0.012742	0.012681	5.90E – 05	80,000
CAEP	0.015116	0.013568	0.012721	8.42E – 04	50,020
CPSO	0.012924	0.012730	0.012675	5.20E – 04	240,000
HPSO	0.012719	0.012707	0.012665	1.58E – 05	81,000
NM-PSO	0.012633	***0.012631***	***0.012630***	8.47E – 07	80,000
G-QPSO	0.017759	0.013524	0.012665	1.27E – 03	2000
QPSO	0.018127	0.013854	0.012669	1.34E – 03	2000
PSO	0.071802	0.019555	0.012857	1.16E – 02	2000
DE	0.012790	0.012703	0.012670	2.70E – 05	204,800
DELC	0.012666	0.012665	0.012665	1.30E – 07	20,000
DEDS	0.012738	0.012669	0.012665	1.30E – 05	24,000
HEAA	0.012665	0.012665	0.012665	1.40E – 09	24,000
PSO-DE	0.012665	0.012665	0.012665	1.20E – 08	24,950
SC	0.016717	0.012923	0.012669	5.90E – 04	25,167
UPSO	N/A	0.022940	0.013120	7.20E – 03	100,000
CDE	N/A	0.012703	0.012670	N/A	240,000
(μ + λ)-ES	N/A	0.013165	0.012689	3.90E – 04	30,000
ABC	N/A	0.012709	0.012665	1.28E – 02	30,000
TLBO	N/A	0.012666	0.012665	N/A	10,000
MBA	0.012900	0.012713	0.012665	6.30E – 05	7650
EPSO	0.016911	0.014056	0.012670	1.27E – 03	2000
	0.014218	0.013030	0.012669	3.64E – 04	10,000
HFPSO	0.016707	0.013213	0.012672	8.91E – 04	2000
	0.013808	0.012894	0.012665	2.30E – 04	80,000

Table 11
Holm–Bonferroni statistical comparison procedure for CEC 2015 10D problems.

Method	Score	z	p	Θ	h
FFPSO	1.0000e + 00	−5.4271e + 00	2.8640e − 08	1.2500e − 02	1 (rejected)
FA	2.7333e + 00	−2.4249e + 00	7.6569e − 03	1.6667e − 02	1 (rejected)
PSO	3.2000e + 00	−1.6166e + 00	5.2984e − 02	2.5000e − 02	0 (accepted)
HPSOFF	3.9333e + 00	−3.4641e − 01	3.6452e − 01	5.0000e − 02	0 (accepted)
HFPSO	4.1333e + 00	–	–	–	–

Table 12
Holm–Bonferroni statistical comparison procedure for CEC 2015 30D problems.

Method	Score	z	p	Θ	h
FFPSO	1.0000e + 00	−6.9282e + 00	2.1311e − 12	1.2500e − 02	1 (rejected)
FA	2.0000e + 00	−5.1962e + 00	1.0173e − 07	1.6667e − 02	1 (rejected)
HPSOFF	3.4667e + 00	−2.6558e + 00	3.9559e − 03	2.5000e − 02	1 (rejected)
PSO	3.5333e + 00	−2.5403e + 00	5.5372e − 03	5.0000e − 02	1 (rejected)
HFPSO	5.0000e + 00	–	–	–	–

Table 13
Holm–Bonferroni statistical comparison procedure for CEC 2017 10D problems.

Method	Score	z	p	Θ	h
FFPSO	1.0000e + 00	−7.5934e + 00	1.5579e − 14	1.2500e − 02	1 (rejected)
PSO	2.5667e + 00	−3.7559e + 00	8.6365e − 05	1.6667e − 02	1 (rejected)
FA	3.4333e + 00	−1.6330e + 00	5.1235e − 02	2.5000e − 02	0 (accepted)
HPSOFF	3.9000e + 00	−4.8990e − 01	3.1210e − 01	5.0000e − 02	0 (accepted)
HFPSO	4.1000e + 00	–	–	–	–

Table 14
Holm–Bonferroni statistical comparison procedure for CEC 2017 30D problems.

Method	Score	z	p	Θ	h
FFPSO	1.0000e + 00	−9.6347e + 00	2.8540e − 22	1.2500e − 02	1 (rejected)
FA	2.0000e + 00	−7.1852e + 00	3.3562e − 13	1.6667e − 02	1 (rejected)
HPSOFF	3.5000e + 00	−3.5109e + 00	2.2327e − 04	2.5000e − 02	1 (rejected)
PSO	3.5667e + 00	−3.3476e + 00	4.0752e − 04	5.0000e − 02	1 (rejected)
HFPSO	4.9333e + 00	–	–	–	–

is accepted and donated $h = 1$ is donated. Else, the null hypothesis is accepted and $h = 0$ is donated.

$$z_j = \frac{S_j - S_0}{\sqrt{\frac{N_A(N_A+1)}{6N_{TP}}}} \tag{14}$$

In Tables 11 and 12, CEC 2015 benchmarks 10D and 30D dimensional problems' Holm–Bonferroni statistical procedure results are given, respectively. In Table 11, there is a significant difference in performance between FFPSO, FA algorithms, and the proposed HFPSO algorithm. Additionally, in comparison with PSO and HPSOFF, the null hypothesis is accepted. Hence, there is no significant difference in performance between with these algorithms and the proposed HFPSO algorithm in CEC 2015 10D problems. In Table 12, CEC 2015 30D problems are statistically compared. According to table, performance between FFPSO, FA, HPSOFF, PSO and proposed HFPSO are significantly different. In the other hand, according to score column, the best method is HFPSO in both 10D and 30D dimensional problems.

In Tables 13 and 14, CEC 2017 benchmarks 10D and 30D dimensional problems' Holm–Bonferroni statistical procedure results are given, respectively. In Table 13, between FFPSO, PSO and the proposed HFPSO algorithm, there is a significant difference in performance. However, in comparison with FA and HPSOFF, the null hypothesis is accepted and there is no significant difference in performance between these algorithms and the proposed HFPSO algorithm in CEC 2017 10D problems. According to Table 14, statistically compared algorithm results of CEC 2017 30D problems are given. 'h' values indicate that, performance between FFPSO, FA, HPSOFF, PSO and proposed HFPSO are statistically significant different. Score columns indicate that, the best method is HFPSO in both 10D and 30D dimensional problems.

The Holm–Bonferroni statistical procedure indicates that, proposed HFPSO algorithm results are clearly better in 30D problems than other related algorithms compared to 10D problems. In 10D problems: statistical results of HPSOFF and PSO or FA algorithms show that there is no significant difference between performances of these algorithms. On the other hand, both in CEC 2015 and CEC 2017 30D problems proposed HFPSO algorithm statistically outperforms other algorithms.

7.5. Comparison of the memetic algorithms on high dimensional problems

In order to see comparison results of memetic algorithms and related algorithms, memetic PSO (MPSO) [20], new memetic FA (NFA) [21] with a local search strategy and a memetic metaheuristic called the shuffled frog-leaping algorithm (SFLA) are employed and compared with related algorithms by use of CEC 2017 100D high dimensional problems. The SFLA is a population-based cooperative search approach inspired by natural memetics. This algorithm mechanism swaps local and global search information. A meme is a unit of cultural evolution that can be transferred from one individual to another. Memeplex is called as parallel communities. The SFLA consists of a frog population partitioned into several memeplexes. Memes of the frogs are independently transferred into a memeplex. Thus, the local search is performed. In SFLA, the local

Table 15
The comparison of memetic algorithm results for CEC 2017 100D problems.

Fn.	PSO	FA	MPSO	NFA	SFLA	HFPSO	Rank					
1	6,2337E+10	4,3699E+11	5,6238E+10	1,8113E+11	4,5712E+11	***4,4471E+10***	3	5	2	4	6	1
2	1,3709E+141	3,4532E+180	3,7674E+139	9,9350E+162	2,0159E+178	***1,0529E+133***	3	6	2	4	5	1
3	5,3800E+05	7,9793E+05	5,5691E+05	5,6506E+05	8,3419E+05	***5,0192E+05***	2	5	3	4	6	1
4	8,6241E+03	1,5011E+05	7,3788E+03	3,6901E+04	1,5990E+05	***4,9001E+03***	3	5	2	4	6	1
5	1,8410E+03	2,6236E+03	1,8696E+03	2,0058E+03	2,6762E+03	***1,6574E+03***	2	5	3	4	6	1
6	6,8893E+02	7,3885E+02	6,8851E+02	7,0262E+02	7,4045E+02	***6,8268E+02***	3	5	2	4	6	1
7	2,9662E+03	1,0060E+04	3,0451E+03	4,0365E+03	1,0336E+04	***2,6554E+03***	2	5	3	4	6	1
8	2,2052E+03	2,9994E+03	2,1772E+03	2,4399E+03	3,1139E+03	***2,0160E+03***	3	5	2	4	6	1
9	7,0698E+04	1,5882E+05	7,3705E+04	7,6284E+04	1,6188E+05	***7,0223E+04***	2	5	3	4	6	1
10	3,2900E+04	3,3460E+04	3,2468E+04	2,9139E+04	3,3537E+04	***2,7614E+04***	4	5	3	2	6	1
11	1,5757E+05	3,5339E+05	1,5083E+05	2,1429E+05	3,5012E+05	***1,3354E+5***	3	6	2	4	5	1
12	1,5477E+10	1,8635E+11	1,2976E+10	6,9869E+10	2,2659E+11	***5,3806E+09***	3	5	2	4	6	1
13	2,3058E+09	4,2807E+10	1,5086E+09	8,3070E+09	4,9759E+10	***2,2492E+08***	3	5	2	4	6	1
14	2,2895E+07	1,8214E+08	2,0883E+07	2,0264E+07	1,9423E+08	***8,7443E+06***	4	5	3	2	6	1
15	6,2988E+08	1,7405E+10	4,9911E+08	1,7783E+09	2,3648E+10	***9,4576E+06***	3	5	2	4	6	1
16	1,1866E+04	2,1089E+04	1,1631E+04	1,4570E+04	2,4722E+04	***8,0251E+03***	3	5	2	4	6	1
17	8,9138E+03	1,8324E+06	9,3046E+03	4,0864E+04	4,6562E+06	***6,2298E+03***	2	5	3	4	6	1
18	3,4869E+07	3,3000E+08	3,1081E+07	3,0029E+07	3,5765E+08	***1,1296E+07***	4	5	3	2	6	1
19	7,4256E+08	1,8263E+10	5,0068E+08	1,3788E+09	2,3512E+10	***5,7930E+07***	3	5	2	4	6	1
20	7,8575E+03	8,4068E+03	7,8535E+03	6,7147E+03	8,4250E+03	***6,4196E+03***	4	5	3	2	6	1
21	3,8499E+03	4,6989E+03	3,8664E+03	4,3364E+03	4,8875E+03	***3,6000E+03***	2	5	3	4	6	1
22	3,5232E+04	3,6331E+04	3,5255E+04	3,1710E+04	3,6450E+04	***3,1114E+04***	3	5	4	2	6	1
23	4,7921E+03	6,1725E+03	4,7025E+03	5,4627E+03	6,8887E+03	***4,6198E+03***	3	5	2	4	6	1
24	5,7744E+03	9,7276E+03	***5,2218E+03***	7,4711E+03	1,1186E+04	5,3992E+03	3	5	1	4	6	2
25	9,1766E+03	7,4852E+04	8,5664E+03	1,8144E+04	8,0127E+04	***6,3591E+03***	3	5	2	4	6	1
26	2,3622E+04	5,7650E+04	***2,2984E+04***	4,0357E+04	6,5708E+04	2,3688E+04	2	5	1	4	6	3
27	4,3206E+03	1,0960E+04	4,1538E+03	6,3659E+03	1,2361E+04	***4,1534E+03***	3	5	2	4	6	1
28	1,1365E+04	5,0295E+04	1,0886E+04	2,3324E+04	5,2480E+04	***8,7851E+03***	3	5	2	4	6	1
29	1,2976E+04	5,5652E+05	1,3872E+04	2,3687E+04	8,3856E+05	***1,0962E+04***	2	5	3	4	6	1
30	1,8632E+09	2,9316E+10	1,6777E+09	5,6044E+09	4,0158E+10	***3,0746E+08***	3	5	2	4	6	1
	2,87	5,07	2,37	3,67	5,93	***1,10***						

search processes are adopted from a particle swarm-like algorithm [64].

In Table 15, memetic PSO, FA and the shuffled frog-leaping memetic algorithm results for CEC 2017 100D problems are given and compared with other related algorithms. As seen in table, in 100D high dimensional problems, proposed HFPSO outperforms the others which are memetic metaheuristic and other standard algorithms. Memetic PSO, FA algorithms show better performance over the standard PSO and FA algorithms as well.

8. Conclusions

In this paper, a hybrid algorithm combining PSO and FA is proposed. Advantages and strengths of PSO and FA are combined and disadvantages of these algorithms i.e. premature convergence and local optima are tried to be mitigated. Hybrid use of these algorithms helps to provide a balance between exploration and exploitation processes. The general results show that this hybridization provides good results over standard PSO and FA in the limited function evaluations of computationally expensive numerical problems. In experiments, CEC 2015 and CEC 2017 benchmark optimization problem sets and some well-known engineering design realistic problems are used. In experiments, 10 dimensional (10D), 30 dimensional (30D) and also 100 dimensional (100D) variables are employed in benchmark optimization problem sets. Proposed hybrid firefly and particle swarm optimization (HFPSO) algorithm is compared with FFPSO and HPSOFF that are other recent hybrid FA and PSO optimization algorithm approaches. Some recently published robust PSO variants: EPSO and ISRPSO, and other optimizers DE, (μ+λ)-ES, CMAES-S, CMAES-G results are extracted from related papers and compared with HFPSO as well. HFPSO has achieved sufficient success over the other approaches. Additionally, in order to see how proposed algorithm performs in realistic optimization problems, three engineering and mechanical design benchmark problems: pressure vessel, welded beam, and tension and compression spring are used, respectively. HFPSO obtains acceptable good engineering and mechanical cost results in comparison with the other algorithms' results that are extracted from previous published papers. In order to indicate whether differences between the algorithms in performance are statistically significant or not, the Holm–Bonferroni procedure is employed. For this purpose, 5 related algorithms and 2 problem sets containing 15 and 30 dimensional problems are used and the statistical tests are performed. Gathered statistical results prove that the proposed HFPSO algorithm is significantly different in performance against the other algorithms in 30D problems. At last, a memetic metaheuristic called the shuffled frog-leaping algorithm (SFLA) is compared with other related algorithms in CEC 2017 100D high dimensional problems. The general results show that the proposed HFPSO has much better performances than other benchmark algorithms. Especially, it is observed that the proposed method's results of unimodal and simple multimodal problems are clearly better than the others. Also, runtimes are measured and graphs of runtime comparison are given. According to runtime graphs, mean runtime results show that the proposed method has acceptable runtime statistics. It exhibits a competitive and good performance over standard PSO, FA, and other hybrid approaches.

References

[1] Q. Chen, B. Liu, Q. Zhang, J. Liang, Evaluation Criteria for CEC 2015 Special Session and Competition on Bound Constrained Single-Objective Computationally Expensive Numerical Optimization, Singapore, 2015 http://web.mysites.ntu.edu.sg/epnsugan/PublicSite/SharedDocuments/CEC-2015/Expensive Problems/CEC15-single objective optimization competition-expensive.pdf.

[2] M.R. Tanweer, S. Suresh, N. Sundararajan, Improved SRPSO algorithm for solving CEC 2015 computationally expensive numerical optimization problems, 2015 IEEE Congr. Evol. Comput. CEC 2015 – Proc. (2015), http://dx.doi.org/10.1109/CEC.2015.7257123.

[3] J.L. Rueda, I. Erlich, MVMO for bound constrained single-objective computationally expensive numerical optimization, 2015 IEEE Congr. Evol.

Comput. CEC 2015 – Proc. (2015) 1011–1017, http://dx.doi.org/10.1109/CEC.2015.7257000.
[4] T.T. Ngo, A. Sadollah, J.H. Kim, A cooperative particle swarm optimizer with stochastic movements for computationally expensive numerical optimization problems, J. Comput. Sci. 13 (2016) 68–82, http://dx.doi.org/10.1016/j.jocs.2016.01.004.
[5] P. Agarwal, S. Mehta, Nature-Inspired algorithms: state-of-art, problems and prospects, Int. J. Comput. Appl. 100 (2014) 14–21, http://dx.doi.org/10.5120/17593-8331.
[6] S.A. Uymaz, G. Tezel, E. Yel, Artificial algae algorithm (AAA) for nonlinear global optimization, Appl. Soft Comput. J. 31 (2015) 153–171, http://dx.doi.org/10.1016/j.asoc.2015.03.003.
[7] I. Fister, X.S. Yang, J. Brest, D. Fister, A brief review of nature-inspired algorithms for optimization: elektroteh, Vestnik/Electrotechnical Rev. 80 (2013) 116–122.
[8] J. Wen, H. Ma, X. Zhang, Optimization of the occlusion strategy in visual tracking, Tsinghua Sci. Technol. 21 (2016) 221–230, http://dx.doi.org/10.1109/ICNN.1995.488968.
[9] M.R. Tanweer, S. Suresh, N. Sundararajan, Self regulating particle swarm optimization algorithm, Inf. Sci. (Ny) 294 (2015) 182–202, http://dx.doi.org/10.1016/j.ins.2014.09.053.
[10] R. Thangaraj, M. Pant, A. Abraham, P. Bouvry, Particle swarm optimization: hybridization perspectives and experimental illustrations, Appl. Math. Comput. 217 (2011) 5208–5226, http://dx.doi.org/10.1016/j.amc.2010.12.053.
[11] X.S. Yang, Firefly algorithms for multimodal optimization, in: Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics). 5792 LNCS, 2009, pp. 169–178, http://dx.doi.org/10.1007/978-3-642-04944-6_14.
[12] I. Fister, X.S. Yang, J. Brest, A comprehensive review of firefly algorithms, Swarm Evol. Comput. 13 (2013) 34–46, http://dx.doi.org/10.1016/j.swevo.2013.06.001.
[13] S.K. Pal, C. Rai, A.P. Singh, Comparative study of firefly algorithm and particle swarm optimization for noisy non-linear optimization problems, Int. J. Intell. Syst. Appl. 4 (2012) 50–57, http://dx.doi.org/10.5815/ijisa.2012.10.06.
[14] C. Blum, A. Roli, M. Sampels, Hybrid Metaheuristics– An Emerging Approach to Optimization, Springer, 2008, http://dx.doi.org/10.1007/978-3-540-78295-7.
[15] I.H. Osman, G. Laporte, Metaheuristics A bibliography, Ann. Oper. Res. 63 (1996) 511–623, http://dx.doi.org/10.1007/BF02125421.
[16] H. Li, C. Nantasenamat, T. Monnor, C. Isarankura-Na-Ayudhya, V. Prachayasittikul, Genetic algorithm search space splicing particle swarm optimization as general-purpose optimizer, Chemom. Intell. Lab. Syst. 128 (2013) 153–159, http://dx.doi.org/10.1016/j.chemolab.2013.08.009.
[17] G.R. Raidl, J. Puchinger, Combining (Integer) linear programming techniques and metaheuristics for combinatorial optimization, Hybrid Metaheuristics 62 (2008) 31–62, http://dx.doi.org/10.1007/978-3-540-78295-7_2.
[18] C. Blum, C. Cotta, A.J. Fernàndez, J.E. Gallardo, M. Mastrolilli, Hybridizations of metaheuristics with branch & bound derivates, Stud. Comput. Intell. 114 (2008) 85–116, http://dx.doi.org/10.1007/978-3-540-78295-7_4.
[19] A. Wan, L. Jiang, C.S.S. Sangeeth, C.A. Nijhuis, Reversible Soft Top-Contacts to Yield Molecular Junctions with Precise and Reproducible Electrical Characteristics, 2014, http://dx.doi.org/10.1002/adfm.201304237, Singapore.
[20] Y.G. Petalas, K.E. Parsopoulos, M.N. Vrahatis, Memetic particle swarm optimization, Ann. Oper. Res. 156 (2007) 99–127, http://dx.doi.org/10.1007/s10479-007-0224-y.
[21] H. Wang, W. Wang, H. Sun, J. Zhao, H. Zhang, J. Liu, X. Zhou, A New Firefly Algorithm with Local Search for Numerical Optimization, Springer, Singapore, 2016, http://dx.doi.org/10.1007/978-981-10-0356-1_2.
[22] W. Huang, S.K. Oh, Z. Guo, W. Pedrycz, A space search optimization algorithm with accelerated convergence strategies, Appl. Soft Comput. J. 13 (2013) 4659–4675, http://dx.doi.org/10.1016/j.asoc.2013.06.005.
[23] C. Yang, W. Gao, N. Liu, C. Song, Low-discrepancy sequence initialized particle swarm optimization algorithm with high-order nonlinear time-varying inertia weight, Appl. Soft Comput. J. 29 (2015) 386–394, http://dx.doi.org/10.1016/j.asoc.2015.01.004.
[24] F. D'Andreagiovanni, On improving the capacity of solving large-scale wireless network design problems by genetic algorithms, in: EvoApplications 2011, Springer Lect. Notes Comput. Sci. Vol. 6625, Springer, Berlin, 2011, pp. 11–20, http://dx.doi.org/10.1007/978-3-642-20520-0_2.
[25] F. D'Andreagiovanni, A. Nardin, Towards the fast and robust optimal design of wireless body area networks, Appl. Soft Comput. J. 37 (2015) 971–982, http://dx.doi.org/10.1016/j.asoc.2015.04.037.
[26] F. D'Andreagiovanni, J. Krolikowski, J. Pulaj, A fast hybrid primal heuristic for multiband robust capacitated network design with multiple time periods, Appl. Soft Comput. J. 26 (2015) 497–507, http://dx.doi.org/10.1016/j.asoc.2014.10.016.
[27] F. D'Andreagiovanni, F. Mett, J. Pulaj, An (MI)LP-based primal heuristic for 3-architecture connected facility location in urban access network design, in: EvoApplications 2016, Lect. Notes Comput. Sci. LNCS 9597, Springer, 2016, pp. 283–298, http://dx.doi.org/10.1007/978-3-319-31204-0_19.
[28] L.M. Gambardella, R. Montemanni, D. Weyland, Coupling ant colony systems with strong local searches, Eur. J. Oper. Res. 220 (2012) 831–843, http://dx.doi.org/10.1016/j.ejor.2012.02.038.
[29] L. Zhang, Y. Tang, C. Hua, X. Guan, A new particle swarm optimization algorithm with adaptive inertia weight based on Bayesian techniques, Appl. Soft Comput. J. 28 (2015) 138–149, http://dx.doi.org/10.1016/j.asoc.2014.11.018.
[30] J. Kennedy, R. Eberhart, Particle swarm optimization, Proceedings of the IEEE international conference on neural networks 4 (1995) 1942–1948, http://dx.doi.org/10.1109/ICNN.1995.488968.
[31] Y. Shi, H. Liu, L. Gao, G. Zhang, Cellular particle swarm optimization, Inf. Sci. (Ny) 181 (2011) 4460–4493, http://dx.doi.org/10.1016/j.ins.2010.05.025.
[32] V. Vassiliadis, G. Dounias, Nature-inspired intelligence: a review of selected methods and applications, Int. J. Artif. Intell. Tools 18 (2009) 487–516, http://dx.doi.org/10.1142/S021821300900024X.
[33] Y. Shi, R. Eberhart, A modified particle swarm optimizer, 1998 IEEE Int. Conf. Evol. Comput. Proceedings. IEEE World Congr. Comput. Intell. (Cat. No.98TH8360) (1998) 69–73, http://dx.doi.org/10.1109/ICEC.1998.699146.
[34] Q. Yu, K. Wang, A hybrid point cloud alignment method combining particle swarm optimization and iterative closest point method, Adv. Manuf. 2 (2014) 32–38, http://dx.doi.org/10.1007/s40436-014-0059-0.
[35] T. Çavdar, PSO tuned ANFIS equalizer based on fuzzy C-means clustering algorithm, AEU Int. J. Electron. Commun. 70 (2016) 799–807, http://dx.doi.org/10.1016/j.aeue.2016.03.006.
[36] S. Helwig, R. Wanka, Theoretical Analysis of Initial Particle Swarm Behavior, Parallel Probl. Solving from Nature–PPSN X, 2008, pp. 889–898 http://link.springer.com/chapter/10.1007/978-3-540-87700-4_88.
[37] S. Gheisari, M.R. Meybodi, BNC-PSO structure learning of bayesian networks by particle swarm optimization, Inf. Sci. (Ny) 348 (2016) 272–289, http://dx.doi.org/10.1016/j.ins.2016.01.090.
[38] W.H. Lim, N.A. Mat Isa, An adaptive two-layer particle swarm optimization with elitist learning strategy, Inf. Sci. (Ny) 273 (2014) 49–72, http://dx.doi.org/10.1016/j.ins.2014.03.031.
[39] M.A. Arasomwan, A.O. Adewumi, On the performance of linear decreasing inertia weight particle swarm optimization for global optimization, Sci. World J. (2013) 2013, http://dx.doi.org/10.1155/2013/860289.
[40] K.R. Harrison, A.P. Engelbrecht, B.M. Ombuki-Berman, Inertia weight control strategies for particle swarm optimization: too much momentum, not enough analysis, Swarm Intell. 10 (2016) 267–305, http://dx.doi.org/10.1007/s11721-016-0128-z.
[41] X.S. Yang, Firefly algorithm, Stochastic test functions and design optimisation, Int. J. Bio-Inspired Comput. 2 (2) (2010) 78–84, http://dx.doi.org/10.1504/IJBIC.2010.032124.
[42] X.S. Yang, Genetic Algorithms, second ed., Luniver Press, 2014, http://dx.doi.org/10.1016/B978-0-12-416743-8.00005-1.
[43] S. Arunachalam, T. AgnesBhomila, M. Ramesh Babu, Hybrid particle swarm optimization algorithm and firefly algorithm based combined economic and emission dispatch including valve point effect, in: Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), Springer International Publishing, 2015, pp. 647–660, http://dx.doi.org/10.1007/978-3-319-20294-5_56.
[44] P. Kora, K.S. Rama Krishna, Hybrid firefly and particle swarm optimization algorithm for the detection of bundle branch block, Int. J. Cardiovasc. Acad. 2 (2016) 44–48, http://dx.doi.org/10.1016/j.ijcac.2015.12.001.
[45] J. Kennedy, R.C. Eberhart, Y. Shi, C. Jacob, J.R. Koza, F.H.B. Iii, D. Andre, M.A. Keane, Swarm Intelligence The Morgan Kaufmann Series in Evolutionary Computation, 2001 http://www.citeulike.org/user/pesimov/article/1084312.
[46] A. Sahu, S.K. Panigrahi, S. Pattnaik, Fast convergence particle swarm optimization for functions optimization, Procedia Technol. 4 (2012) 319–324, http://dx.doi.org/10.1016/j.protcy.2012.05.048.
[47] S.M. Abd-Elazim, E.S. Ali, A hybrid particle swarm optimization and bacterial foraging for optimal power system stabilizers design, Int. J. Electr. Power Energy Syst. 46 (2013) 334–341, http://dx.doi.org/10.1016/j.ijepes.2012.10.047.
[48] A. Nickabadi, M.M. Ebadzadeh, R. Safabakhsh, A novel particle swarm optimization algorithm with adaptive inertia weight, Appl. Soft Comput. J. 11 (2011) 3658–3670, http://dx.doi.org/10.1016/j.asoc.2011.01.037.
[49] M. Taherkhani, R. Safabakhsh, A novel stability-based adaptive inertia weight for particle swarm optimization, Appl. Soft Comput. J. 38 (2016) 281–295, http://dx.doi.org/10.1016/j.asoc.2015.10.004.
[50] G. J. Xin, Y. Hai Chen, A particle swarm optimizer with multi-stage linearly-decreasing inertia weight, in: Proc. 2009 Int. Jt. Conf. Comput. Sci. Optim., CSO 2009, 2009, pp. 505–508, http://dx.doi.org/10.1109/CSO.2009.420.
[51] Q. Dou, Z. Yu, Z. Shi, E. Yu, Y. Zheng, Cluster-degree analysis and velocity compensation strategy of PSO, in: Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), Springer, Berlin, Heidelberg, 2008, pp. 98–106, http://dx.doi.org/10.1007/978-3-540-92137-0_11.
[52] S.M.K. Heris, Implementation of Firefly Algorithm (FA) in MATLAB, 2015, http://yarpiz.com/259/ypea112-firefly-algorithm. (Accessed 16 December 2016).
[53] K.K. Bharti, P.K. Singh, Opposition chaotic fitness mutation based adaptive inertia weight BPSO for feature selection in text clustering, Appl. Soft Comput. J. 43 (2016) 20–34, http://dx.doi.org/10.1016/j.asoc.2016.01.019.
[54] W. Han, P. Yang, H. Ren, J. Sun, Comparison study of several kinds of inertia weights for PSO, in: Proc. 2010 IEEE Int. Conf. Prog. Informatics Comput., PIC 2010, 2010, pp. 280–284, http://dx.doi.org/10.1109/PIC.2010.5687447.
[55] M.R. Tanweer, R. Auditya, S. Suresh, N. Sundararajan, N. Srikanth, Directionally driven self-regulating particle swarm optimization algorithm, Swarm Evol. Comput. 28 (2016) 98–116, http://dx.doi.org/10.1016/j.swevo.2016.01.006.
[56] A. Sadollah, A. Bahreininejad, H. Eskandar, M. Hamdi, Mine blast algorithm: a new population based algorithm for solving constrained engineering

optimization problems, Appl. Soft Comput. J. 13 (2013) 2592–2612, http://dx.doi.org/10.1016/j.asoc.2012.11.026.
[57] B.K. Kannan, S.N. Kramer, An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design, J. Mech. Des. 116 (1994) 405–411, http://dx.doi.org/10.1115/1.2919393.
[58] C.A. Coello Coello, Use of a self-adaptive penalty approach for engineering optimization problems, Comput. Ind. 41 (2000) 113–127, http://dx.doi.org/10.1016/S0166-3615(99)00046-9.
[59] J.S. Arora, Introduction to Optimum Design, 2004, pp. X5000–X5009, http://dx.doi.org/10.1016/B978-0-12-064155-0.
[60] S. García, A. Fernández, J. Luengo, F. Herrera, A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability, Soft Comput. 13 (2009) 959–977, http://dx.doi.org/10.1007/s00500-008-0392-y.
[61] S. Holm, A simple sequentially rejective multiple test procedure a simple sequentially rejective multiple test procedure, Scand. J. Stat. Scand. J. Stat. 6 (1979) 65–70, http://dx.doi.org/10.2307/4615733.
[62] F. Neri, E. Mininno, G. Iacca, Compact particle swarm optimization, Inf. Sci. (Ny) 239 (2013) 96–121, http://dx.doi.org/10.1016/j.ins.2013.03.026.
[63] R. Cheng, Y. Jin, A social learning particle swarm optimization algorithm for scalable optimization, Inf. Sci. (Ny) 291 (2015) 43–60, http://dx.doi.org/10.1016/j.ins.2014.08.039.
[64] M. Eusuff, K. Lansey, F. Pasha, Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization, Eng. Optim. 38 (2006) 129–154, http://dx.doi.org/10.1080/03052150500384759.