Applied Soft Computing Journal 78 (2019) 545–568

Contents lists available at ScienceDirect

Applied Soft Computing Journal

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

A new meta-heuristic optimizer: Pathfinder algorithm

Hamza Yapici [a,*], Nurettin Cetinkaya [b]

[a] The Graduate Institute, Konya Technical University, Konya, Turkey
[b] Electrical & Electronics Engineering Department, Konya Technical University, Konya, Turkey

HIGHLIGHTS

- A new heuristic algorithm has been proposed.
- The method is a swarm-based algorithm and different in mathematical model.
- The proposed method has been tested on some test beds.
- The proposed method showed a superior performance to find global optima.
- Also, it has been applied to a real engineering problem and found good results.

ARTICLE INFO

Article history:
Received 5 October 2018
Received in revised form 7 February 2019
Accepted 5 March 2019
Available online 8 March 2019

Keywords:
Optimization
Optimization techniques
Metaheuristics
Multi-objective optimization
Pathfinder algorithm

ABSTRACT

This paper proposes a new meta-heuristic algorithm called Pathfinder Algorithm (PFA) to solve optimization problems with different structure. This method is inspired by collective movement of animal group and mimics the leadership hierarchy of swarms to find best food area or prey. The proposed method is tested on some optimization problems to show and confirm the performance on test beds. It can be observed on benchmark test functions that PFA is able to converge global optimum and avoid the local optima effectively. Also, PFA is designed for multi-objective problems (MOPFA). The results obtained show that it can approximate to true Pareto optimal solutions. The proposed PFA and MPFA are implemented to some design problems and a multi-objective engineering problem which is time consuming and computationally expensive. The results of final case study verify the superiority of the algorithms proposed in solving challenging real-world problems with unknown search spaces. Furthermore, the method provides very competitive solutions compared to well-known meta-heuristics in literature, such as particle swarm optimization, artificial bee colony, firefly and grey wolf optimizer.

1. Introduction

The swarm based meta-heuristic optimization algorithms have an important role to optimize the modern engineering problems. This is due to their flexibility, derivation-free mechanisms and local optima avoidance. First of all, it can be mentioned that a heuristic method is simple. Generally, these methods have been based on simple concepts of physical phenomena in nature. Also, this helps the researchers to implement easily meta-heuristic methods to their problems. Furthermore, heuristic methods can be modified for different areas without any major modifications in their structures. This makes them flexible. Moreover, these methods are concerned only with the inputs and outputs of a problem. So that, these methods become derivative-free. Also, in contrast of gradient-based methods, meta-heuristics optimize a problem stochastically, say, the process starts with random initial solution or solutions and they improve these solutions using the random operators during the process. This allows them to be able to avoid local optimums. According to this ability, the meta-heuristics can be implemented to many research areas.

The meta-heuristic methods can be taken into account in three classes: evolutionary based [1], physical-based [2] and swarm intelligence based [3]. Evolutionary based methods start with random population and then evaluate this initial population using one or several operators such as crossover, mutation and selection during the optimization process. In particular, these methods do not care with the information of previous population. The second class is physical-based, which is inspired by physical rules in universe. The search agents in these methods explore the search space in accordance with specific rules of physic. The last class is, generally speaking, based on behaviors of swarm of animals in nature. The methods in this class use the collective movement intelligence of animals. These methods can save the information

* Corresponding author.
E-mail address: hyapici@konya.edu.tr (H. Yapici).

https://doi.org/10.1016/j.asoc.2019.03.012
1568-4946/

about optimization problem over the process. On the other side, they have less operators to be adjusted, and thus, they are easy adapted with minor revisions to different areas.

Independently of the differences of algorithms, a general characteristic is that meta-heuristic algorithms divide the optimization process as exploration and exploitation [4]. In the exploration phase, an algorithm performs the searching a possible solution area, where the algorithm needs stochastic operators for the global search and also randomly search abilities. Conversely, in the part of the exploitation, the ability of the algorithm is shaped around local search capability in the promising areas obtained in the exploration [5].

Additionally, a meta-heuristic algorithm is not suitable for solving all optimization problems. Precisely at this time it is worth to mentioning the NFL (No Free Lunch) theorem [6]. This theorem has proved that a meta-heuristic algorithm can give very promising solutions on some optimization problems, but cannot show good performance on all optimization problems. For this reason, the NFL theory causes new meta-heuristic algorithms to be proposed or existing algorithms to be improved. Therefore, in this study, it is the motivation of this study to improve a new meta-heuristic algorithm inspired by the hunting behavior of animals leaded by a leader individual. Despite the use of leadership hierarchy in the literature, there is no a simple model of searching a feeding area or a hunt and directing a swarm by an individual or a pathfinder. For this reason, in this study, a new meta-heuristic algorithm, called Pathfinder (PFA), inspired by the behavior of searching a hunt or feeding area with the leadership of an individual in the animal herds, is presented. In contrast to the algorithms in the literature, in the proposed method there is a leader and the other members follow it. However, the motion of all particles is not orderly, all of them move randomly. Also, the proposed method is completely different in terms of mathematical model and inspiration.

1.1. Review of literature

We mentioned in previous section, the most important aspect of heuristic algorithms is that they are inspired by evolutionary, physical systems, swarm intelligence. Evolutionary based methods mimic the evolution concepts. It can be mentioned that the Genetic Algorithm (GA) [7] is well-known method in literature. This algorithm based on theory of Darwin for evolution. GAs have some operators to evaluate its initial population generated randomly which are crossover, mutation and selection. Some of the other famous methods are Differential Evolution (DE) [8] and Evolutionary Programming (EP) [9].

The other branch of algorithms is physics-based. They are based on physical systems. These methods use the physical rules of gravitational force, ray casting inertia force and etc. Some of them are big-bang big-crunch (BBBC) [10], gravitational search algorithm (GSA) [11], charged system search (CSS) [12], central force optimization (CFO) [13], artificial chemical reaction optimization algorithm (ACROA) [14], black hole (BH) [15] algorithm, ray optimization algorithm (RO) [16], small-world optimization algorithm (SWOA) [17], galaxy-based search algorithm (GBSA) [18], and curved space optimization (CSO) [19].

The other class is swarm intelligence-based methods. These methods use the concepts of behavior animals. So that, particle swarm optimization (PSO) [20] and artificial bee colony algorithm (ABC) [21] have been inspired from the behavior of the fish and bird schooling and the food foraging behavior of honey bees in nature, respectively. Another one is ACO [22], inspired by the behavior of ants which is based on finding the shortest path from the nest to the food. Since the development of heuristic algorithms, the interest on this area has been increased. Some recently developed algorithms are firefly algorithm (FA) [23]. The krill herd algorithm (KH) [24]. The bat algorithm (BA) [25], the cuckoo search (CS) [26], artificial algae algorithm (AAA) [27], tree seed algorithm (TSA) [28], the grey wolf optimizer algorithm (GWO), presented in [29], the social spider algorithm (SSA) [30], the moth-flame algorithm (MFA) [31], the salp swarm optimizer (SSO) [32], the whale optimization algorithm (WOA) [33], the dolphin echolocation algorithm (DEA) [34], cat swarm optimizer (CSO) [35] and lion optimization algorithm (LOA) [36].

The paper is organized as follows. Section 2 presents the inspiration and model of Pathfinder algorithm (PFA). Versions of single-objective and multi-objective PFA are introduced in this section. Two metric experiments have been carried out in Section 3 on variety of classical benchmark functions, composite functions and multi-objective functions. In Section 4, several design problems have been handled. Section 5 contains an experiment of challenging problem of multi-objective optimization problem. Finally, Section 6 concludes the study together with some recommends.

2. Pathfinder Algorithm (PFA)

2.1. Inspiration

The searching, exploiting and hunting abilities of animal swarms have always been a focus of interest for many scientists. All behaviors in a swarm are carried out on the basis of common action of all individuals. Along with that, an individual leads the swarm and this individual directs many acts. Additionally, this individual takes away the herds to targets such as pasture, water and feeding area. The leader may vary depending on the ability to achieve the target [37,38].

The animals living as a group often decide the movement through the social hierarchy among the members. These animals might need to make a decision either with leader individual or without leader individual. However, the leadership is temporal and few individuals may have knowledge of food location, hunting area, route or etc. [39,40]. In [39], authors proposed a simple model to demonstrate some informed members can lead the whole herd. In [41], a self-propelled particles-based model has been developed that includes leaders and followers. Also, in some other previous works [42–45], the collective movement of herds has been modeled. The authors considered the updating the direction of movement with terms of alignment and attraction/repulsion forces in 2-dimensional (2D) space. Therefore, due to the models given in [40,41], we proposed the mathematical model that includes interaction between members and leader in a swarm.

2.2. Mathematical model

In the proposed model, each member has a position in 2D, 3D or d-dimensional spaces. If a member in swarm is located in the most promising area in any time, then it will be selected as leader. In addition, it is supposed that the all candidate solutions of a problem are the position vector of individuals, thus, the individuals in the herd can walk in 2D, 3D and d-dimensional space. Note that we called the leader of swarm as pathfinder. To look for prey or feeding area and for following the pathfinder the model given below is proposed.

$$x\left(t+\Delta t\right)=x^{0}\left(t\right)\cdot n+f_i+f_p+\varepsilon \tag{2.1}$$

where t is time, x is the position vector, n is the unit vector without any angle, f_i is a pairwise interaction with neighbors x_i and x_j, f_p is the global force which depends on global optimum or position of pathfinder and ε is vector of vibration. On the other hand, the position of pathfinder is updated according to the following equation:

$$x_p\left(t+\Delta t\right)=x_p\left(t\right)+\Delta x+A \tag{2.2}$$

where, x_p is the position vector of pathfinder, Δx is the distance taken by pathfinder to move from one point to another and A is the vector of fluctuation rate.

Essentially, the model of collective swarm movement mentioned above cannot directly applied to solve optimization problems. Moreover, some modifications are required to make it applicable. The key purpose of optimization problem is to find the optimum. First of all, we modified Eqs. (2.1) and (2.2) to Eqs. (2.3) and (2.4) for applying our approach. The first modification is given below:

$$x_i^{K+1}=x_i^K+R_1\cdot\left(x_j^K-x_i^K\right)+R_2\cdot\left(x_p^K-x_i^K\right)+\varepsilon,\quad i\geq 2 \tag{2.3}$$

where, K represents the current iteration, x_i is the position vector of ith member, x_j is the position vector of ith member, R_1 and R_2 are the random vectors. R_1 is equal to αr_1 and R_2 is equal to βr_2, where r_1 and r_2 are random variable uniformly generated in the range of [0,1], α is the coefficient for interaction which defines the magnitude of movement of any member together with its neighbor and β is the coefficient of attraction which sets the random distance for keeping the herd roughly with leader. Also, r_1 and r_2 provide a random movement. There are two significant situations when $\alpha\rightarrow 0$, $\beta\rightarrow 0$ and $\alpha\rightarrow\infty$, $\beta\rightarrow\infty$. For the first case, each individual will move randomly without any interaction in the search space. On the other hand, in the second case, each individual will not be in interaction, this means that the individuals may not move any direction and follow the leader or may not able to change their positions. During the many executions of the proposed method, it has been observed that when $\alpha<1$ and $\beta<1$, it can be difficult for follower members to change their positions and to approach the leader member. On the other hand, when $\alpha\gg 1$ and $\beta\gg 1$ (i.e., 3, 4, 7, 10, 100, etc.) the location change distance of the follower members can be increased and the follower members can locate at far positions from the leader. Thus, in the both explained cases, the follower members cannot find the promising solutions. Ideally, α and β should be around 1. Furthermore, α and β can be constant (i.e., 1.1, 1.5, 1.8, etc.). In this case, position changes of follower members are more similar to position changes of particles in PSO. In this study, α and β are randomly selected in the range of [1,2] over the course of iterations. The final term is for vibration, where ε is generated in each iteration using Eq. (2.5).

The second modification is given below:

$$x_p^{K+1}=x_p^K+2r_3\cdot\left(x_p^K-x_p^{K-1}\right)+A \tag{2.4}$$

where, r_3 is a random vector uniformly generated in the range of [0,1], A is generated in each iteration using Eq. (2.6).

$$\varepsilon=\left(1-\frac{K}{K_{max}}\right).u_1.D_{ij},\quad \mathrm{D}_{ij}=\left\|x_i-x_j\right\| \tag{2.5}$$

$$A=u_2.e^{\frac{-2K}{K_{max}}} \tag{2.6}$$

where, u_1 and u_2 are random vectors range in [−1,1], D_{ij} is the distance between two members and K_{max} is the maximum number of iterations. When second terms in Eqs. (2.3) and (2.4) and third term in Eq. (2.3) are zero, A and ε can provide random movement (walk) for all members. Therefore, A and ε, for ensuring multi-directional and random movement, should be range in proper values. Also, to provide random movement these terms should be included a random number generator. Moreover, they perform rapid change in early iterations and then these changes slowdown in later iterations, and thus, this case facilitates the searching in exploration and exploitation phases. By u_1 and u_2 variables are set in the range [−1,1], members can also move to their previous positions. However, if $u_1<-1$ and $u_2<-1$ or $u_1>+1$ and $u_2>+1$, then the members can change their positions with big steps, therefore, members can move away from possible solutions.

In the PFA, the pathfinder tries to find the best food area/hunt. The best food area/hunt can be assumed the global optimum. In any iteration, the location of pathfinder is assigned as current optimum in the current iteration, so the other members move towards it. However, the main problem is to find the global optimum of optimization problems, because of its uncertainty. Therefore, we assumed that the best solution detected so far is the global optimum and accepted as the food area/hunt to be exploited by the herd.

The proposed method starts with the initialization of positions of herd members randomly. Then, the fitness of each individuals is calculated and the position of individual with the best fitness is selected as pathfinder to be followed. This member of herd moves in the search space using Eq. (2.4) and the vector of fluctuation rate A is generated simultaneously with Eq. (2.6) in each iteration. All steps of process are iteratively and the proposed method ends with reaching the maximum number of iterations.

To understand the pathfinder movement, one-dimensional position vector changing is demonstrated in Fig. 1. It can be seen that pathfinder in the current location of x can take up to the next distance covered. Pathfinder can arrive to expected location by adjusting A and random vector r_3 which causes the pathfinder move to any location between x' and x''. This allows the PFA to explore search space globally and underlines the exploration phase. In Fig. 2, it can be seen that initial position starting with random uniform distribution at time T, illustrated left side, pathfinder moves around the promising area and other individuals move randomly towards it. Note that the red dot represents the first leader. Then, in the right side of Fig. 2, after Δt_1 time the blue dot has become the leader. Furthermore, the location of red dot shown in middle has been changed to new location in right side, where the new position is obtained using Eq. (2.4) which modified for 2D space. In Fig. 3, it is also observed that the movement in 3D space is same to movement in 2D space. The distribution of colored members around the food source marked as '*' illustrates the effective search of proposed method. Further, the reasonable of changing location of members shows the capable for exploration and exploitation ability. This can be due to the search ability of pathfinder. Similarly, this model can be adopted and applied for high dimensional search space. Note that the pseudo code is given in Fig. 4. According to this figure, PFA starts with the initial random population and it then computes the fitness value. The position of pathfinder is a current optimum. In iterative process, it determines α and β and updates the position of pathfinder. If new position is better than old it should be updated. Then positions of follower members are updated with considering the bounds. PFA calculates new fitness of each member and when any member has better position than pathfinder, it is assigned as new pathfinder. After this step, PFA updates final population using "*if then*" rules and updates the vectors of A and ε. Finally, PFA checks the end criterion to stop the execute the iterative process. In the literature, there are different criteria proposed as end criterion including a fixed number of generations, the number of iterations, a located string with a certain value, and no change in the average fitness after some generations. Note that, in this study, maximum number of iterations have been used for end criterion to be able to compare under the same conditions.

On the other hand, the convergence of A and ε to around 0 emphasize the transition between exploration and exploitation. Other variables that support the exploration are α and β. These variables provide randomization of movement to get close to

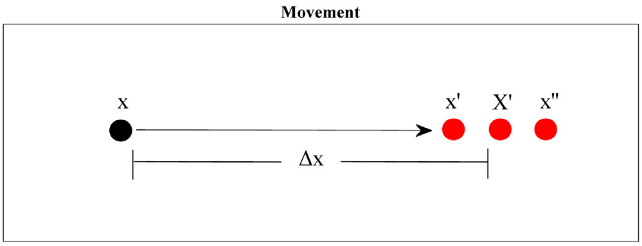

Fig. 1. Position update of Pathfinder in 1D.

the leader in order to stochastically find the hunt or feeding area. Also, α and β support the individuals to behave randomly throughout optimization process and have to be a random value between 1 and 2 in order to make exploration not only in initial iterations but also in all iterations. So, individuals in swarm have the potential to move towards global optimum. To understand how the proposed model and method apply in optimization problems, some remarks are given below:

1. The position vector of each member corresponds to the promising area of search space. In this context, the vector giving the best fitness is selected as pathfinder. It also means that PFA assigns the best location as the location of pathfinder, so, the fitness value in each iteration never get lost.
2. The positions of whole population are updated with respect to the pathfinder. In the course of iterations, the leader can be changed by the fitness value. Moreover, the whole members can explore and exploit the hunt or food source.
3. The position of each individual is updated with respect to the other members and pathfinder. So, they move towards the pathfinder and get close to their neighbor.
4. Random movement of all members can be cause to easily stagnate in local optima, but the changing of vibration vector ε and to get close to the pathfinder make it possible to avoid this situation.
5. The convergence of A and ε to 0 allow the exploration at first, then provide the exploitation.

All remarks mention above show that the PFA will be able to solve optimization problems. Owing to its adaptive structure, the proposed algorithm can be capable to avoid local optima and can effectively obtain the best solution in optimization. So, the proposed model can be implemented to both single-objective and multi-objective problems.

2.3. Differences of PFA

Many swarm-based algorithms have been proposed in literature. Among these intelligence algorithms, PSO, ACO, ABC and FA are well-known and widely used algorithms. Essentially, PFA is proposed for continuous optimization and may be classified as swarm-based methods. However, PFA has some differences.

PSO has been modeled based on coordinated motions of animal flock. In PSO, particles have a velocity and try to find global best position. Also, all particles have a local position and they change their current position according to their velocity. On the

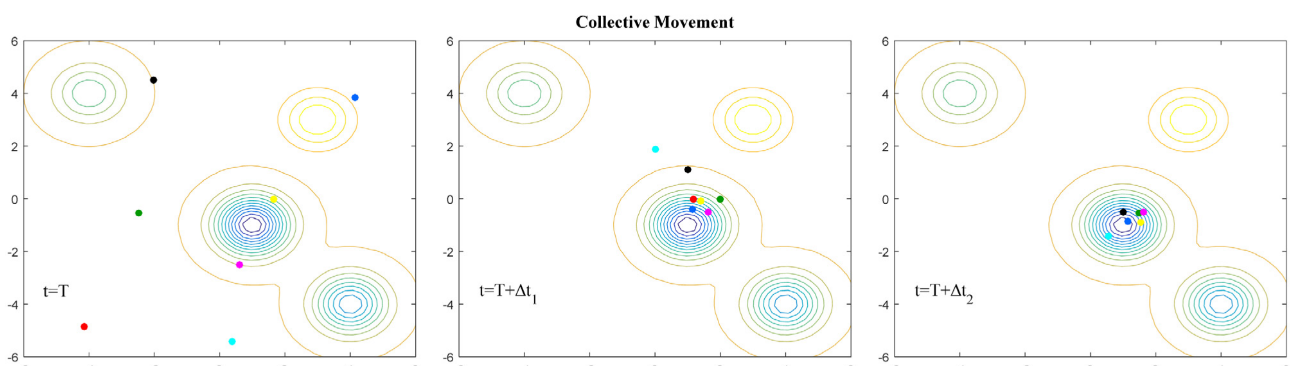

Fig. 2. Movement of 7 individuals in 2D. In time t=T, the individuals marked as yellow dot is the leader of herd. After Δt_1 time, the member marked as dark blue dot shown in middle figure has been leader. Then, after Δt_2 time, the member marked with dark blue has stayed as leader and other members followed this member.

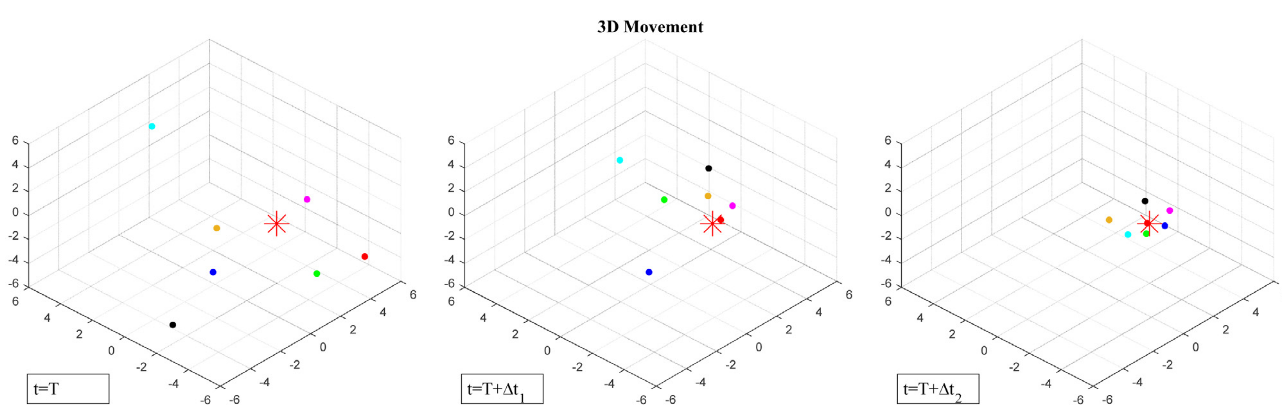

Fig. 3. Movement of 7 individuals in 3D. In time t=T, the individuals marked as pink dot is the leader of herd. After Δt_1 time, the member marked as red dot shown in middle figure has become leader. Then, after Δt_2 time, the member marked with red has stayed as leader and other members get close him.

```
Load PFA parameter
Initialize the population
Calculate the fitness of initial population
Find the pathfinder
while K < maximum number of iterations
    α and β = random number in [1,2]
    update the position of pathfinder using Equation (2.4) and check the bound
    if new pathfinder is better than old
        update pathfinder
    end
    for i=2 to maximum number of populations
        update positions of members using Equation (2.3) and check the bound
    end
    calculate new fitness of members
    find the best fitness
    if best fitness < fitness of pathfinder
        pathfinder = best member
        fitness = best fitness
    end
    for i=2 to maximum number of populations
        if new fitness of member (i) < fitness of member (i)
            update members
        end
    end
    generate new A and ε
end
```

Fig. 4. Pseudo code of PFA.

other hand, the members have no any velocity in PFA. Furthermore, PFA has a leader and members of swarm follow this leader according to their neighbor position and movement of leader. Members do not only follow the leader, but also explore the search space. This case cause to different behaviors. In PFA, the next positions are determined by interactions between members. Additionally, leader in PFA acts separately from the swarm. This case is a distinctive character of proposed method from many other algorithms.

There are some obviously differences between PFA and ACO. ACO uses ant behavior to find the optimum. Ants track the pheromone trace which gives the positive feedback for later path. Additionally, there are no hierarchic structure between ants and all ants move collectively. Conversely, in PFA, a leader leads the whole swarm, and individuals do not need feedback for moving forward, but they keep close to their neighbor using the interaction between them. Additionally, despite of ACO is commonly used to compute combinatorial problems like TSP (Travel Salesman Problem), in recent years some ACO-based algorithms [46] are used for solving continuous problems.

In ABC and Glow-Worm Swarm Optimization (GSO) [47], individuals in different types do the different jobs. ABC classifies the population into three types that search the local optima and global optima differently. According to this method, searching space, finding the promising solutions and global searching are performed by these types of individuals. GSO uses behavior of fireflies. Each member chooses a random member and move towards it in the context of its brightness, thus, whole population divided into subgroups. In contrast of these methods, all members in PFA are equal and follow the best individuals. Moreover, the leader and all other members are able to explore and exploit the food source or hunt.

Another method same as GSO is FA that uses brightness of fireflies. In reformulated movement of firefly swarm, each individual move in the direction of maximum light intensity. A special case of this method can be mentioned as accelerated PSO by the specific parameter of this algorithm. Also, the whole population can be divided into subgroups as in the GSO. However, the swarm of PFA is not divided into subgroups and members only follow one member which has the best fitness. Meantime, the distances between neighbor individuals can be decreased in the course of iterations. Meantime, each individual is able to get close to another one.

There are also two other swarm-based method proposed to solve optimization problem and they have some special features. GWO and SSO are modeled according to leader behavior of animal group. In both GWO and SSO, special models are pointed out which make orderly movement towards the next position for each individual. This case is due to their special parameters. Also, the individuals can update their position around the prey or food source. Furthermore, in order to mathematical model in GWO, there is a hierarchical structure among the individuals. Moreover, in SSO, salps group follow a leader salp which updates its position around the food source and the follower salps orderly move towards it according to the mathematical model of SSO algorithm. In contrast, expect the leader, members in PFA is not divided in hierarchical structures and all can do random movement and random search to find global optima because of their parameters.

As information mentioned above, movement of follower members in PFA may be a special case of particles movement in PSO or FA. However, complete behavior of swarm is quite different from these methods. On the other hand, swarm-based algorithms use different types of parameters that manipulate the optimization process to find global optima. The features of parameters used in PFA may allow to find promising solution during the optimization process and easy adapting to unimodal, multimodal and also multi-objective problems. In addition, its searching strategy and special behavior model can contribute the performance of PFA.

2.4. Adjusting parameters

To define proper parameters of an algorithm for optimization problems may take a long time. The definition test of parameters may give the high performance at the high computational problems. In fact, evaluation time of multi-objective and real-world problems may be longer than classical benchmark functions that causes to be impractical for adjusting parameters. Researchers proposed some models to replace the trial-and-error parameter settings and also specified into three groups [48]. First model is Fixed Parameter Model which chooses a combination before the numerical analysis or simulation and uses experimental or theoretical information of parameters. These adjusted parameters are being constant during the process. Second model is Deterministic Parameter Model which utilizes some rules to set the parameters during the process. Final model is Adaptive Parameter Model which adjust the parameters using adaptive learning mechanisms throughout the process [49–52].

In this study, we preferred the second model to test the performance of proposed model. In PFA, four important and adjustable parameters have been used to guide the behavior of movement in search space, namely, fluctuation rate A, vibration vector ε and coefficients α and β. The adjusting strategy of these parameters mentioned in Section 2.2 make the PFA adaptable for optimization problems. In different optimization problems, such as unimodal, multimodal or single-objective optimization problems, the same adjusting scheme has been utilized over the testing of proposed method. When considering the results for optimization problems, it can be seen that the proposed method obtains very competitive results.

2.5. PFA for multi-objective optimization

2.5.1. Multi-objective optimization

Multi-objective problems (MOPs) contain more than one objective function and the objective function constitute a multidimensional space. All of them are optimized simultaneously. Basic concept of MOP is [53,54]:

To minimize,

$$F\left(\vec{x}\right)=\left\{f_1\left(\vec{x}\right),\ f_2\left(\vec{x}\right),\ f_3\left(\vec{x}\right),\ \ldots,\ f_k\left(\vec{x}\right)\right\}$$

Subject to,

$$\mathrm{g}_i\left(\vec{x}\right)\geq 0,\ \ i=1,2,3,\ldots,n \tag{2.7}$$

and

$$\mathrm{h}_i\left(\vec{x}\right)=0,\ \ i=1,2,3,\ldots,m$$

where $\vec{x}=[x_1,\ x_2,\ x_3,\ldots,x_k]^T$ is the vector of decision variables, k is the number of objective functions, n is the number of inequality constraints and m is the number of equality constraints.

These problems do not allow to use of relational operators, because there are many criteria to compare solutions. On the other hand, due to the nature of MOPs different other operators are needed and some definition about these operators given below [32,53,54].

The main operator for comparing two solutions is Pareto dominance.

Definition 1 (*Pareto Dominance*)**.** A vector $\overrightarrow{u}=(u_1,\ u_2,\ u_3,\ldots,u_k)$ dominates $\overrightarrow{v}=(v_1,\ v_2,\ v_3,\ldots,v_k)$ if and only if $\overrightarrow{u}$ is partially less than denoted as $\overrightarrow{v}$ denoted as $\overrightarrow{u}\preccurlyeq\overrightarrow{v}$, $u_i\leq v_i,\ \forall i\in\{1,2,3,\ldots k\}$.

If Pareto dominance does not provide for two solutions, Pareto optimal or non-dominated solutions are used. $\vec{x}^*$ is Pareto optimal if there is no suitable vector that reduces some criterion without causing a simultaneous increase in at least one other criterion. The Pareto optimality is defined below:

Definition 2 (*Pareto Optimality*)**.**: If $\forall\vec{x}\in$ S, S is the promising area in search space, a vector $\vec{x}^*\in$ S is Pareto optimal. $\forall i\in\{1,2,3,\ldots k\}$, $f_i\left(\vec{x}\right)=f_i\left(\vec{x}^*\right)$ or at least for one i $f_i\left(\vec{x}\right)>f_i\left(\vec{x}^*\right)$.

In contrast of single-objective optimization problems, MOPs has a set of different solutions called Pareto optimal solutions, also, in objective space called Pareto front. These two sets are defined below.

Definition 3 (*Pareto Optimality Set*)**.**: For a given $F\left(\vec{x}\right)$, the Pareto optimal set P^* is pointed out as $P^*:=\{x\in \mathrm{S}|\ \mathrm{x}\text{ is Pareto optimal}\}$

Definition 4 (*Pareto Front*)**.**: For a given $F\left(\vec{x}\right)$, Pareto front $P\mathcal{F}^*$ is defined as $P\mathcal{F}^*:=\left\{\overrightarrow{u}=f=(f_1\left(x\right),\ f_2\left(x\right),\ f_3\left(x\right),\ \ldots,\ f_k\left(x\right)),\ x\in P^*\right\}$

Computational process of MOPs can be very easy with definitions mentioned above.

2.5.2. Multi-objective PFA (MOPFA)

The proposed method is capable to find the hunt or food source and saves it as global optima. However, it is not sufficient to solve multi-objective optimization problems due to the information given below.

* Global best is one solution obtained by PFA, and PFA is not capable to store multiple best solutions. Also, the updating of individuals performs to find hunt or food source and PFA tries to find the best location in each iteration, but there is not only one solution in multi-objective optimizations. Eventually, PFA cannot obtain multiple best locations.

Instead of single solution in single-objective problems, MOPs include different solutions so-called Pareto optimal set. In order to implement PFA for computing MOPs, it has to be modified properly. However, the number of variables of Pareto optimal set should be defined, then, Pareto front generated should be minimized with respect to global Pareto front. The distribution of solution should be dispersed, so this case can give a uniformly distribution vector as possible. As considered the population-based structure of PFA, it is inevitable that several non-dominated solutions can be generated in a run. So, same as other methods in [32,53,54], some main troubles have to be considered; in order to choose dominated solutions among nondominated solutions member/(s) should be selected, more than one solution needs to be handled and stored, these solutions should also be well diffused throughout the Pareto front and to avoid convergence to a single solution the diversity in herd should be preserved.

At this stage, the PFA is equipped with an external supply box as same as in [53,54] and has more than one pathfinder. Each pathfinder is preserved in supply box and only one can be selected for movements of members and best members are selected as pathfinders. The best non-dominated solutions obtained so far is stored in supply box. During the computation, non-dominated solutions in this store are used as pathfinders and complete box is the final output of the algorithm.

The supply box has a limited size. The pathfinders are compared with all store, then, if a pathfinder dominates a solution, this solution has to be replaced with pathfinder. If there are more than one dominated solution, then all should be removed. Thus, it can be guaranteed that best non-dominated solutions obtained so far are stored in the box. When the size of box is less than the number of pathfinders, pathfinders are random selected and then the same number of non-dominated solutions as the box size can be stored.

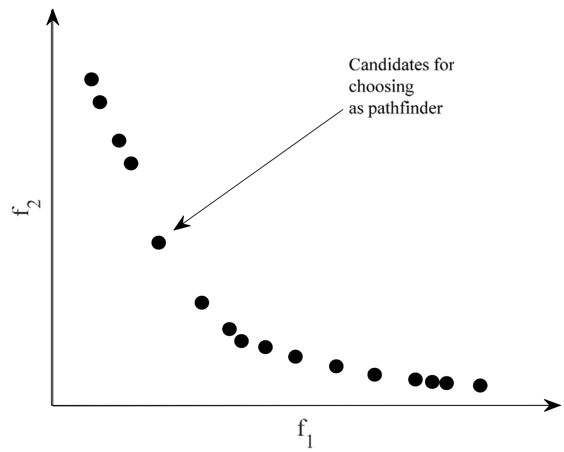

Fig. 5. Selecting of pathfinder in supply box.

The pathfinder is selected randomly from the box. But a proper method is that selects it between the less crowded neighborhood. Each non-dominated set is sorted within its own stacking distance. This distance is given as $\vec{d} = \left(f^{max} - f^{min}\right)/box\ size$ where f^{max} and f^{min} are maximum and minimum values of each objective respectively. The supply box with one solution is the best one. After arranging pathfinders according to the number of neighboring solutions, they are compared with the random operator r_c. The non-dominated solution with the low rank number is selected, also, this means that the solution in less crowded region has highest probability for being pathfinder. In Fig. 5, non-dominated solution pointed out with arrow is the best candidate to be selected as pathfinder and has the higher probability than others. Pseudo code of MOPFA is given in Fig. 6.

In Fig. 6, it can be seen that MOPFA starts with random initial population. The proposed method finds the objective values for each member, then assigns members to pathfinders with respect to best non-dominated solutions. All pathfinders are added to supply box and if the box is full, some of them are eliminated according to their worst value. It then defines a random number to select a pathfinder for movement of swarm. The next case is to update position of all agents using Eqs. (2.3) and (2.4) with checking the bounds, respectively. Finally, new pathfinders are chosen according to the best non-dominant solutions and supply box is updated. All above steps are iterated until stop criterion satisfied.

2.6. The effectiveness of PFA

The proposed method saves the best position obtained so far as position of pathfinder and it never get lost. The pathfinder is capable to explore and exploit the hunt or food source. Other members follow the pathfinder and interact with their neighbor, so they can explore and exploit the target in search space. The random movement of followers can prevent the PFA from stagnating in local optima. Therefore, PFA can be able to solve optimization problems. The parameters of PFA allow it to find effective solutions and to get accurate estimations. The remarks mentioned above can provide the proposed method to outperform other algorithms. But, according to the NFL theorem, it is not guaranteed to solve all optimization problems.

2.7. Time complexity

The computational complexity of the proposed method is $O\left(t\left(n * N + (F) * N\right)\right)$, where t is the number of iterations, n is the number of dimensions, N is the population size of swarm, and F is the cost of objective.

3. Results

The numerical analysis has been included testing of benchmark functions, statistical analysis of multi-objective functions, implementing to design problems and implementing of engineering application. In the next subsection, we discussed these problems respectively. Also, all methods have been coded in MATLAB 2016 software, and PC with Intel i7 CPU, 8 GB RAM and Nvidia GTX950 hardware has been used for all simulations. Note that, all methods have been simulated in same conditions (same population size and number of iterations).

3.1. Testing on benchmark functions

In this stage, the proposed method has been tested on 27 benchmark functions. The first 17 benchmark functions are unimodal and multimodal classical benchmark functions taken from [55,56]. In spite of their simplicity, these test functions have been chosen to compare with some well-known algorithms in the literature. These are given Tables 1 and 2. On the other hand, other last ten benchmark functions are composite functions handled in CEC2017 [57]. The composite test functions are listed in Table 3. In Tables 1–2, D represents dimension, minimum is the optimal fitness of functions and feasible bound is boundary of test functions. Additionally, the composite functions given Table 3 are shifted and rotated of combined functions, where N is the number of functions used for combining, σ is used to control each function, *bias* defines which optimum is global and λ is used for adjusting the heights of functions.

To prove the theoretical claims mentioned in previous, some of comparisons have been carried out. Note that for comparing the results obtained by proposed method, PSO, ABC, FA, CS, TSA, SSO and GWO have been used. Moreover, PSO, Teaching Learning Based Optimization (TLBO) [58] and first and second ranked methods in the CEC2017 competition, Effective Butterfly Optimizer using Covariance Matrix Adapted Retreat phase (EBO with CMAR9 [59] and Success-History based Adaptive Differential Evolution with Linear decrease in population size LSHADE-cnEpSin [60], have been deal with for composite test functions.

The test functions can be considered in three groups: unimodal, multimodal and composite functions [32,61,62]. It should be noted that the details of composite functions can be seen in [59]. The PFA has been run 30 times on each classical function and only one 50 times for composite functions. The results for classical functions are reported as minimum value, maximum value, mean value, average and standard deviation of the optimum solutions obtained in 30 runs, also, for composite functions, reported as function error value ($f - F$, where f is the solution obtained in each run and F is the global optimum) which presented in best, worst, mean, median and standard variance of solutions obtained in 50 runs. In the second case, the computational analysis is performed with 10 dimensions, 30 dimensions and 50 dimensions, respectively.

To see efficiency, capability and superiority of a method, it should be handled the qualitative and quantitative metrics. In the following subsections, the qualitative and quantitative metrics is detailed, results are reported and details are explained.

3.1.1. Qualitative results

In the single-objective optimization, one of the qualitative metrics is the convergence curve commonly used by researchers. Also, some other metrics can be used to show the performance of an algorithm. Several test samples with different characteristics can be needed to demonstrate the capabilities of an algorithm. Hence, an algorithm should exhibit different behavior according

```
Load MOPFA parameter
Initialize the population
Calculate the fitness of initial population
Find the pathfinders
Assign best members to pathfinders for non-dominated solutions in supply box
while K < maximum number of iterations
    K=K+1
    α and β = random number range in [1,2]
    r_c=random number
    For i = 1 to box size
        Assign a random index to pathfinder (i)
        Find minimum index < r_c
    End
    pathfinder = members in box with minimum index
    update the position of pathfinder using Equation (2.4) and check the bound
    if new pathfinder is better than old
        update pathfinder
    end
    for i=2 to maximum number of populations
        update positions of members using Equation (2.3) and check the bound
    end
    calculate new fitness of members
    generate new A and ε
    Find the new pathfinders
    Update to non-dominated solutions in supply box
end
```

Fig. 6. Pseudo code of MOPFA.

to design of test functions. Meanwhile, algorithms may be observed in terms of their performance and clearly compared with others. The next case is to define an appropriate test bed for PFA for generating the qualitative solutions.

In general terms, PFA is tested on three groups of benchmark functions; unimodal, multimodal and composite functions. There is only one optimum point in unimodal test functions and they have no local optimum point. These functions are proper for convergence test and exploitation analysis. Other two types of functions, multimodal and composite functions, have many optimum points which are proper for testing of avoidance of local optima and observing the exploration phases. However, composite test functions are very complex and challenging which are time-consuming and computationally expensive. Additionally, ten of them are used in this study for seeing competitivity of proposed method.

Search history of agents, convergence curve and the evaluation of mean value of fitness are given for qualitative metrics. Fig. 7 shows these metrics. Figures of history of agents point out the position of individuals over the process, hereby, we can see how sampled of the positions by whole individuals. Although we can observe the movement of herd, we cannot see the order of exploration and exploitation phases. In contrast of infrequent distribution of sampled positions in unimodal functions, in multimodal and composite functions, more points can be sampled in the non-promising points because of their difficulty. It can be seen that, in Fig. 7, individuals of PFA can be able to steer for promising areas. This motion supports the local optima avoidance and exploration capabilities of proposed method. The movement towards the global point in all test functions can support the exploitation phase and convergence.

The next results show the evaluation of mean value of fitness, in another saying mean fitness history, in Fig. 7. The results of search history do not show if exploration and exploitation phases are useful in the context of increasing the efficiency of the initial population and obtaining the true global point, which are the main goal of algorithms [28,32]. To see and prove that issues, mean value of the results obtained in each iteration and the global optimum obtained so far are recorded for the final output. The curves of mean fitness are inclined to decrease in optimization processes. Because benchmark test functions are problems of minimization, these curves verify that PFA is capable for evolving and then improving the population during the iterations. The exploration phase affects the fitness of each individual due to random dispersion and movement around search space, so individuals can deviate from its target. Even so, the average value of fitness decreases as well during the optimization due to the exploitation capability.

To state the approximation to the global point, the convergence curves verify how well PFA evaluate and improve its population. In Fig. 7, it can be proved that PFA improves the fitness for converging to global optima over the course of iterations. This status is not generally coherent. Especially, in multimodal and composite test functions, sometimes PFA performs no improvements in some iterations due to the carrying out exploration phase, which occasionally results in individuals located in non-promising solutions. The different behavior can be emerged when optimizing the different problems.

The remarks given above prove that PFA is capable for regularly explore and exploit the search space. Also, PFA can converge to best promising solutions effectively and is able to find a set of quality solutions for a problem. Note that the functions of Rosenbrock, Sum Square, Schwefel, composite function 1 and composite function 2 have been employed for Fig. 7.

3.1.1.1. Exploration and exploitation. As seen in Table 4, the proposed method has achieved competitive results. It can be mentioned that the unimodal functions are proper to test an algorithm for exploitation phase. PFA outperforms other methods in testing of f_1, f_3 and f_5. Hence, it can be mentioned that PFA has a superior performance to exploit the best location.

Table 1
Unimodal benchmark functions.

Function name	Function	Minimum	D	Feasible bounds
Rosenbrock	$f_1 = \sum_{i=1}^{D-1} \left\{100\,(x_{i+1} - x_i)^2 + (x_i - 1)^2\right\}$	0	20	$[-30, 30]^D$
Sum Squares	$f_2 = \sum_{i=1}^{D} i{x_i}^2$	0	30	$[-10, 10]^D$
Step 2	$f_3 = \sum_{i=1}^{D} (\lfloor x_i + 0.5 \rfloor)^2$	0	30	$[-100, 100]^D$
Schwefel 2.22	$f_4 = \sum_{i=1}^{D} \lvert x_i\rvert - \prod_{i=1}^{D} \lvert x_i\rvert$	0	30	$[-10, 10]^D$
Schwefel 1.2	$f_5 = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} x_j\right)^2$	0	30	$[-100, 100]^D$
Chung Reynolds	$f_6 = \left(\sum_{i=1}^{D} {x_i}^2\right)^2$	0	30	$[-100, 100]^D$

Table 2
Multimodal benchmark functions.

Function name	Function	Minimum	D	Feasible bounds
Goldstein Price	$f_7 = \left\{1 + (x_1 + x_2 + 1)^2 \left(19 - 14x_1 + 3{x_1}^2 - 14x_2 + 6x_1x_2 + 3{x_2}^2\right)\right\} \times \left\{30 + (2x_1 - 3x_2)^2 \left(18 - 32x_1 + 12{x_1}^2 + 48x_2 - 36x_1x_2 + 27{x_2}^2\right)\right\}$	3	2	$[-2, 2]^D$
Branin RCOS	$f_8 = \left(x_2 - \frac{5.1}{4\pi^2}{x_1}^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	0.398	2	$[-5, 5]^D$
Six-hump Camel	$f_9 = 4{x_1}^2 - 2.1{x_1}^4 + \frac{1}{3}{x_1}^6 + x_1x_2 - 4{x_2}^2 + 4{x_2}^4$	−1.0316	2	$[-5, 5]^D$
Hartman 3	$f_{10} = -\sum_{i=1}^{4} c_i \exp\left[-\sum_{j}^{3} a_{ij}\left(x_j - p_{ij}\right)^2\right]$	−3.8628	3	$[0, 1]^D$
Shekel 5	$f_{11} = -\sum_{i=1}^{5} \left[(x - a_i)\,(x - a_i)^T + c_i\right]^{-1}$	−10.1532	4	$[0, 10]^D$
Shekel 7	$f_{12} = -\sum_{i=1}^{7} \left[(x - a_i)\,(x - a_i)^T + c_i\right]^{-1}$	−10.4028	4	$[0, 10]^D$
Trid 6	$f_{13} = \sum_{i=1}^{D} (x_i - 1)^2 - \sum_{i}^{D} x_i x_{i-1}$	−50	6	$\left[-6^2, 6^2\right]^D$
Griewank	$f_{14} = \frac{1}{4000}\sum_{i=1}^{D} {x_i}^2 - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	0	20	$[-600, 600]^D$
Ackley	$f_{15} = -20\exp\left\{-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} {x_i}^2}\right\} - \exp\left\{\frac{1}{D}\sum_{i=1}^{D} \cos(2\pi x_i)\right\} + 20 + e$	0	30	$[-32, 32]^D$
Schwefel	$f_{16} = D * 418.9829 + \sum_{i=1}^{D} -x_i \sin\left(\sqrt{\lvert x_i\rvert}\right)$	0	30	$[-500, 500]^D$
Zakharov	$f_{17} = \sum_{i=1}^{D} {x_i}^2 + \left(\frac{1}{2}\sum_{i=1}^{D} ix_i\right)^2 + \left(\frac{1}{2}\sum_{i=1}^{D} ix_i\right)^4$	0	30	$[-5, 10]^D$

However, multimodal and composite functions have many local optima. Therefore, they may be suitable for exploration phase to show ability of PFA. As results given in Tables 4–6, PFA is able to obtain competitive results and it outperforms other algorithm in many functions, thus, it can be mentioned that it has superior capability in terms of exploration.

3.1.2. Quantitative results

Although results of qualitative metrics show the capability of exploration and exploitation, it is not enough to prove the performance of PFA. In this case, four quantitative metrics for unimodal and multimodal functions have been used to quantify the performance of proposed method. These are minimum value, maximum value, mean value and standard deviation of best results obtained in 30 runs. On the other hand, for composite functions mean value, median value and standard deviation of best solutions obtained in 50 runs has been used. Also, minimum and maximum value of all best solutions found in 50 runs has been considered. These metrics show the best and worst solutions obtained during all runs, stability of PFA and the performance in average.

For unimodal and multimodal benchmark functions, to ensure an equivalent comparison, population size and the maximum number of iterations of algorithms employed, are equal to 30 and 1000, respectively. For each algorithm, the parameters handled in the latest version, presented in literature, have been used to support the best performance. However, for composite functions population size is set as 18*D and the stopping criteria of number of function evaluation is set as 10000*D, where D is dimensions of functions. All results are tabulated in Tables 4–6.

The results tabulated in Table 4 point out that PFA outperforms other methods in the half of tests of unimodal functions. As seen in Table 5, the proposed method outperforms other methods in over the half of tests of multimodal test functions. While PFA can benefit from the exploitation because of the single optimum in unimodal function, it can explore the search space efficiently in the tests of multimodal functions which have more than one local optima. In general, minimum and maximum values obtained by PFA show that it performs a better performance than the others. Also, mean and standard deviation values are an evidence that PFA's superiority is stable. When the results in Table 6 are examined, PFA has achieved very competitive results at these tests as well. Especially, PFA has acquired quite good results against CEC2017 competition champions in low dimensions. However, it must be noted that EBO, the champion of competition, and LSHADE-cnEpSin, the third of competition, are very robust and superior methods. In addition, composite functions are very challenging test beds. On the other hand, the results obtained by PFA verify that it carries out superior and steady performance during 50 runs. Indeed, the minimum values found by PFA are highly competitive results. Moreover, PFA has shown better performance than PSO and TLBO on more than one functions in 10D, 30D and 50D. Consequently, these findings point out that PFA can be capable for solving very challenging problems as well. It is noteworthy that, due to the nature of PFA, the exploitation ability tends to be better than exploration ability. It can be concluded in the results found in all tests that PFA has a low capability for abrupt changes in solutions. Nevertheless, it is not a concern due to the position updating approach utilized, since exploratory behavior of PFA is also good enough. Note that since the results in [58] are better than results obtained in numerical analysis performed by code using, the values of original study have been considered.

3.1.3. Scalability of PFA

In addition to tests mentioned above, to analyze the scalability of proposed algorithm, a set of simulations on some benchmark functions (f_2, f_6, f_{16} and f_{17}) with a varied number of dimensions have been performed. The simulations have been carried out with 10D, 50D and 100D. PSO, TSA, SSO and GWO have been used for comparison and results are listed in Table 7. The following remarks have been observed:

- The performance of PFA is verified. PFA has achieved all the results in 10D, 50D and 100D simulations proportionally as the previous tests and its performance is satisfactory with compared to other algorithms.
- In the previous tests, PFA obtained better results than all other methods in f_{16} function and also better than PSO, TSA and SSO in f_2, f_6 and f_{17} functions. In 10D test, it then achieved better results than PSO and SSO in f_2, f_6 and f_{17} and obtained better results than SSO and GWO in f_{16}.
- In 50D and 100D tests, it achieved better results than all methods in f_{16}, also, it finally obtained better results than PSO, SSO and TSA in f_2, f_6 functions.

To sum up, the results obtained in these simulations have confirmed the scalability of PFA. The performance of PFA is not significantly affected when solving problems with various parameters. This can be improved by increasing the number of functions evaluations.

3.1.4. Computational time of PFA

In this subsection, to show the time performance of PFA, f_1, f_3, f_{15} and f_{16} has been used. The simulations have been performed with a varied number of population sizes. For comparison, PSO, TSA, SSO and GWO have been used with population size of 30, 100, 300 and all results have been obtained in 10 runs and the number of maximum iterations has been set as 1000. Not that the results have been compared on min (minimum), mean (average of all runs) and max (maximum) times. The results are listed in Table 8, where time elapsed is given as second. It can be pointed out that the computational time elapsed with proposed method is better that some methods. But PSO is outperform all other methods and also our proposed method because of its nature and simple structure.

On the other side, the small values of time mean that a method is less complex. By the results it can be seen that PFA is less complex than TSA, SSO and GWO, and more complex than PSO.

3.2. The results of MOPFA

To test the performance of MOPFA, some of the multi-objective test functions in the literature are employed. Four challenging multi-optimization test beds presented in [63–65] have been used similarly to the single-objective tests. The mathematical model of all functions given in Table 9. The test functions called ZDT functions here has been represented as ZDT1, ZDT2, ZDT3 and ZDT4. In this subsection, two efficient and robust methods are handled for verifying the results; MOPSO and MOSSO. For a good observation, the search agent and maximum iteration have been set as 100 and 300, respectively, and the dimensions of functions has been adjusted to 30 as generally given in literature. For proving the performance for quantitative metric analysis in 30 runs, the *Generational Distance* and *Spacing* metrics have been presented which were proposed by Van Veldhuizen [65] (for detail see [54] and [65]) and results of this analysis listed in Table 10. In addition, the Pareto optimal front is demonstrated in Fig. 8a–d.

The results specify that MOPFA has superior capability and outperforms MOPSO and MOSSO on the majority of ZDT functions. When considering the Pareto optimal fronts shown in Fig. 8, the proposed method has acquired slightly better results than others. Despite the concave-shaped Pareto optimal front of ZDT1, MOPFA is able to efficiently converge to the true front. In contrast, ZDT2 has a convex shape, and therefore, the convergence is slightly lower. However, MOPFA is better than the others even if the low differences. Even though ZDT3 function has many separated regions, the performance of MOPFA on this test function is provided both in terms of its ability to converge and to cover the true Pareto optimal solution. In this step, from the results obtained and those on the previous ZDT tests, a similar pattern can be observed in the Pareto optimal fronts acquired, in which MOPFA indicates the better results. In these results, it may be seen that MOPFA can be capable for covering the separated zones. In final test of ZDT4 function with 3 objectives, the solutions illustrated in Fig. 8d show the similar result to previous test. The results indicate that MOPFA is able to approximate the true Pareto front in the final test as well. The results and findings of MOPFA prove that it is capable for solving multi-objective problems and can be used for real engineering problems.

4. Real engineering applications

In this section, PFA is implemented to several constrained engineering design problems: tension/compression spring [66], welded beam [67], pressure vessel [68] and corrugated beam [69].

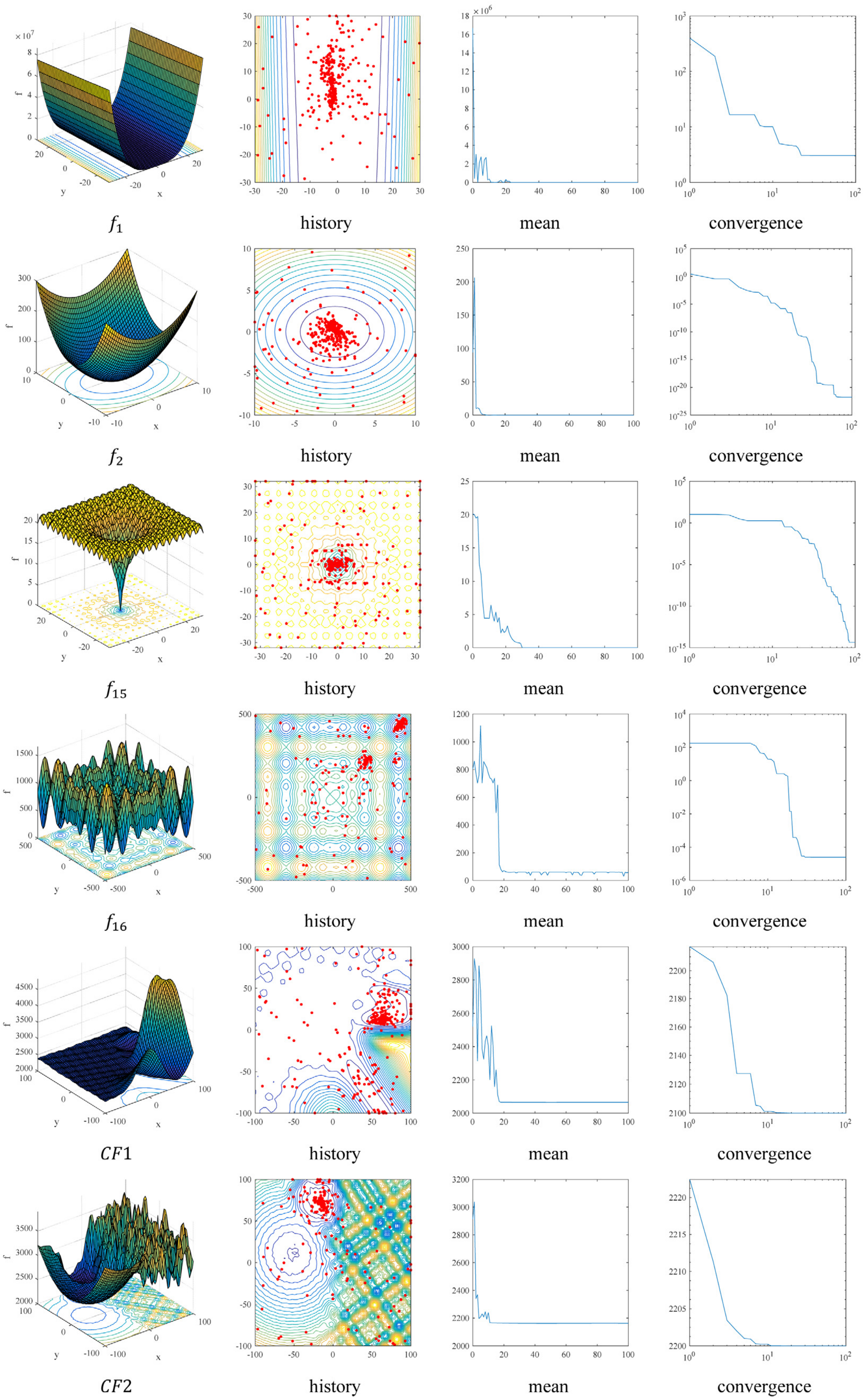

Fig. 7. Search history of PFA, evaluation of mean value of fitness and convergence curve.

Table 3
Composite functions.

Function		Bound	Global optimum
CF1	$f_{18}: \left\{ \begin{array}{c} f_1 = \text{Rotated and Shifted Rosenbrock's Function,} \\ f_2 = \text{Rotated and Shifted High Conditioned Elliptic Function,} \\ f_3 = \text{Rotated and Shifted Rastrigin's Function,} \end{array} \right\},$ $\sigma = [10, 20, 30]$ $\lambda = [1, 1e6, 1]$ $bias = [0, 100, 200]$	$[-100, 100]^D$	2100
CF2	$f_{19}: \left\{ \begin{array}{c} f_1 = \text{Rotated and Shifted Rastrigin's Function,} \\ f_2 = \text{Rotated and Shifted Griewank's Function,} \\ f_3 = \text{Rotated and Shifted Modified Schwefel's Function,} \end{array} \right\},$ $\sigma = [10, 20, 30]$ $\lambda = [1, 10, 1]$ $bias = [0, 100, 200]$	$[-100, 100]^D$	2200
CF3	$f_{20}: \left\{ \begin{array}{c} f_1 = \text{Rotated and Shifted Rosenbrock's Function,} \\ f_2 = \text{Rotated and Shifted Ackley's Function,} \\ f_3 = \text{Rotated and Shifted Modified Schwefel's Function,} \\ f_4 = \text{Rotated and Shifted Rastrigin's Function,} \end{array} \right\},$ $\sigma = [10, 20, 30, 40]$ $\lambda = [1, 10, 1, 1]$ $bias = [0, 100, 200, 300]$	$[-100, 100]^D$	2300
CF4	$f_{21}: \left\{ \begin{array}{c} f_1 = \text{Rotated and Shifted Ackley's Function,} \\ f_2 = \text{Rotated and Shifted High Conditioned Elliptic Function,} \\ f_3 = \text{Rotated and Shifted Griewank's Function,} \\ f_4 = \text{Rotated and Shifted Rastrigin's Function,} \end{array} \right\},$ $\sigma = [10, 20, 30, 40]$ $\lambda = [1, 1e6, 10, 1]$ $bias = [0, 100, 200, 300]$	$[-100, 100]^D$	2400
CF5	$f_{22}: \left\{ \begin{array}{c} f_1 = \text{Rotated and Shifted Rastirigin's Function,} \\ f_2 = \text{Rotated and Shifted HappyCat Function,} \\ f_3 = \text{Rotated and Shifted Ackley's Function,} \\ f_4 = \text{Rotated and Shifted Discus Function,} \\ f_5 = \text{Rotated and Shifted Rosenbrock's Function,} \end{array} \right\},$ $\sigma = [10, 20, 30, 40, 50]$ $\lambda = [10, 1, 10, 1e6, 1]$ $bias = [0, 100, 200, 300, 400]$	$[-100, 100]^D$	2500
CF6	$f_{23}: \left\{ \begin{array}{c} f_1 = \text{Rotated and Shifted Expanded Scaffer's Function,} \\ f_2 = \text{Rotated and Shifted Modified Schwefel's Function,} \\ f_3 = \text{Rotated and Shifted Griewank's Function,} \\ f_4 = \text{Rotated and Shifted Rosenborck's Function,} \\ f_5 = \text{Rotated and Shifted Rastrigin's Function,} \end{array} \right\},$ $\sigma = [10, 20, 20, 30, 40]$ $\lambda = [1e26, 10, 1e6, 10, 5e4]$ $bias = [0, 100, 200, 300, 400]$	$[-100, 100]^D$	2600
CF7	$f_{24}: \left\{ \begin{array}{c} f_1 = \text{Rotated and Shifted HGBat Function,} \\ f_2 = \text{Rotated and Shifted Rastrigin's Function,} \\ f_3 = \text{Rotated and Shifted Modified Schwefel's Function,} \\ f_4 = \text{Rotated and Shifted BentCigar Function,} \\ f_5 = \text{Rotated and Shifted High Conditioned Elliptic Function,} \\ f_6 = \text{Rotated and Shifted Expanded Scaffer's Function,} \end{array} \right\},$ $\sigma = [10, 20, 30, 40, 50, 60]$ $\lambda = [10, 10, 2.5, 1e26, 1e6, 5e4]$ $bias = [0, 100, 200, 300, 400, 500]$	$[-100, 100]^D$	2700
CF8	$f_{25}: \left\{ \begin{array}{c} f_1 = \text{Rotated and Shifted Ackley's Function,} \\ f_2 = \text{Rotated and Shifted Griewank's Function,} \\ f_3 = \text{Rotated and Shifted Discus Function,} \\ f_4 = \text{Rotated and Shifted Rosenbrock's Function,} \\ f_5 = \text{Rotated and Shifted HappyCat Function,} \\ f_6 = \text{Rotated and Shifted Expanded Scaffer's Function,} \end{array} \right\},$ $\sigma = [10, 20, 30, 40, 50, 60]$ $\lambda = [10, 10, 1e6, 1, 1, 5e4]$ $bias = [0, 100, 200, 300, 400, 500]$	$[-100, 100]^D$	2800
CF9	$f_{26}: \left\{ \begin{array}{c} f_1 = \text{Hybrid Function 5 given in CEC2017 Competition,} \\ f_2 = \text{Hybrid Function 8 given in CEC2017 Competition,} \\ f_3 = \text{Hybrid Function 9 given in CEC2017 Competition,} \end{array} \right\},$ $\sigma = [10, 30, 50]$ $\lambda = [1, 1, 1]$ $bias = [0, 100, 200]$	$[-100, 100]^D$	2900
CF10	$f_{27}: \left\{ \begin{array}{c} f_1 = \text{Hybrid Function 5 given in CEC2017 Competition,} \\ f_2 = \text{Hybrid Function 6 given in CEC2017 Competition,} \\ f_3 = \text{Hybrid Function 7 given in CEC2017 Competition,} \end{array} \right\},$ $\sigma = [10, 30, 50]$ $\lambda = [1, 1, 1]$ $bias = [0, 100, 200]$	$[-100, 100]^D$	3000

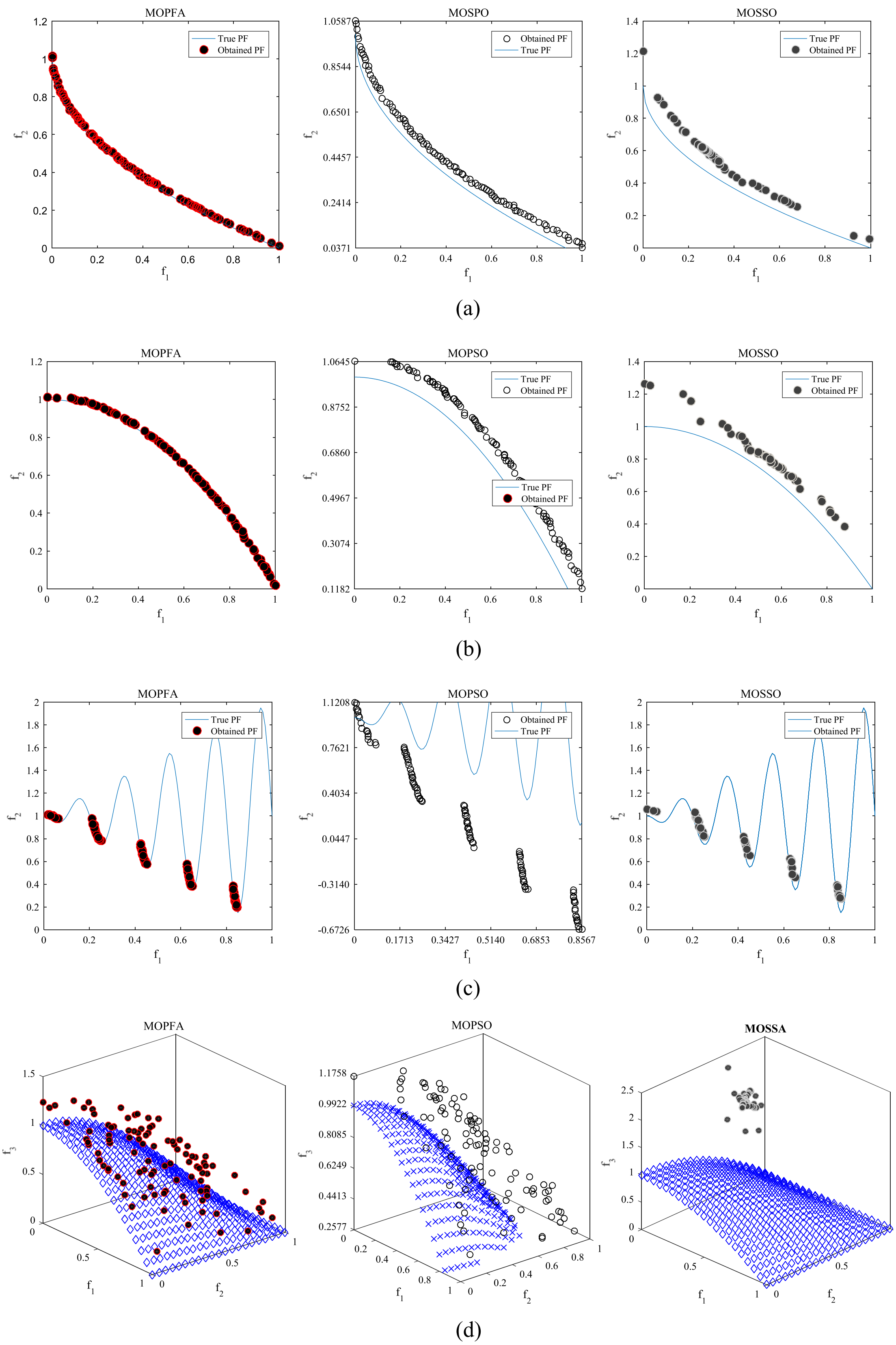

Fig. 8. The best optimal front achieved by MOPFA, MOPSO and MOSSO on multi-objective test functions: (a) test of ZDT1, (b) test of ZDT2, (c) test of ZDT3, (d) test of ZDT4 with 3 objectives.

The structures and parameters of these design problems are illustrated in Fig. 9. The design problems mentioned have equality and inequality constraints, therefore, PFA must contain a constrained processing method to be capable for optimizing these problems.

Table 4
Results of unimodal functions.

Method		f_1	f_2	f_3	f_4	f_5	f_6
PSO	min	21.4120	0.0011	0.0001	20.0022	0	5.8956e−07
	max	9.0072e+4	2.4000e+2	1.0100e+4	1.1000e+2	2.4364e−06	1.0000e+8
	mean	2.4586e+4	9.7000e+2	2.6667e+3	59.3372	9.1958e−08	1.3333e+7
	std	4.0146e+4	5.6820e+2	4.4981e+3	19.2828	4.4668e−07	3.4574e+7
ABC	min	45.6699	0.5220	3.3764	3.9142	3.0154e−06	7.7944
	max	463.7563	1.0976	9.1417	63.0735	0.00998	131.4302
	mean	180.8207	0.7735	6.0348	29.0627	0.0013	34.5897
	std	101.7531	0.1433	1.6583	15.1728	0.0021	22.9010
FA	min	14.8901	4.7060e−05	7.0370e−05	0.0063	3.6188e−13	9.3158e−09
	max	1.8431e+3	0.1514	0.0002	0.1807	1.0659e−08	9.2230e−08
	mean	145.3708	0.0173	0.0001	0.0249	1.4404e−09	3.4158e−08
	std	336.1224	0.0363	4.9043e−05	0.0372	2.9103e−09	2.0357e−08
TSA	min	12.0228	9.9233e−21	4.8551e−11	5.8249e−15	5.8234e−07	6.3346e−38
	max	65.2165	4.6708e−19	4.4265e−10	3.1939e−14	0.0006	3.3809e−34
	mean	15.9661	1.4986e−19	1.4606e−10	1.6645e−14	9.7824e−05	1.9213e−35
	std	10.2698	1.0741e−19	9.9283e−11	7.4642e−15	0.0001	6.1891e−35
SSO	min	13.6187	2.8391e−05	7.6074e−09	0.0052	3.6206e−16	4.4794e−17
	max	481.5863	4.1118	1.8128e−08	3.5395	1.0664e−12	3.1282e−16
	mean	83.8358	0.6649	1.1816e−08	0.9831	2.0319e−13	1.6733e−16
	std	120.9319	0.9247	2.7841e−09	1.0158	3.0136e−13	6.8852e−17
GWO	min	15.3451	**6.7856e−62**	0.2496	**4.9228e−36**	4.9201e−08	**2.5206e−122**
	max	18.9693	**7.3070e−58**	1.4634	**1.2329e−33**	0.0003	**2.1366e−115**
	mean	16.2326	**2.8675e−59**	0.7742	**1.2166e−34**	2.9880e−05	**1.1002e−116**
	std	0.8757	**1.3285e−58**	0.3430	**2.2354e−34**	5.9408e−05	**4.0006e−116**
PFA	min	**8.4556**	3.8190e−27	**2.3215e−11**	2.2231e−16	**5.7422e−17**	2.3080e−52
	max	**15.9949**	2.9967e−24	**5.9741e−11**	2.5798e−13	**1.8171e−14**	1.5872e−44
	mean	**11.0791**	5.5674e−25	**3.7435e−11**	3.4831e−14	**1.8231e−15**	9.9813e−46
	std	**2.3153**	7.9092e−25	**9.8816e−12**	6.2094e−14	**3.6044e−15**	3.3585e−45

Table 5
Results of multimodal functions.

Method		f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}
PSO	min	3.0000	**0.3979**	−1.0316	−3.8628	−10.1532	−10.4029	−50.0000	3.1224e−09	0.0257	3.3375e+3	2.2345e+2
	max	3.0000	**0.3979**	−1.0316	−3.8549	−2.6304	−1.8375	−50.0000	0.0589	20.1613	5.9741e+3	1.0035e+3
	mean	3.0000	**0.3979**	−1.0316	−3.8612	−7.1218	−8.5826	−50.0000	0.0228	13.7475	4.4414e+3	5.4417e+2
	std	8.4502e−16	**0.0000**	6.6486e−16	0.0032	3.3664	2.9045	1.9952e−6	0.0171	7.8276	7.2466e+2	1.9519e+1
ABC	min	3.0000	**0.3979**	−1.0316	−3.8628	−10.1532	−10.4029	−50.0000	0.0194	2.2589	3.1019e+3	3.4015e+2
	max	3.0000	**0.3979**	−1.0316	−3.8549	−10.1531	−10.4029	−50.0000	0.3170	3.4969	4.7707e+3	1.3797e+3
	mean	3.0000	**0.3979**	−1.0316	−3.8612	−10.1532	−10.4029	−50.0000	0.1529	3.0235	3.9350e+3	9.0286e+2
	std	8.4098e−16	**0.0000**	6.6835e−16	3.0767e−14	2.1482e−05	1.1394e−15	1.8554e−07	0.0819	0.2623	5.5634e+2	2.3035e+2
FA	min	3.0000	0.3979	−1.0316	−3.8628	−10.1532	−10.4029	−50.0000	0.0001	0.0022	4.0103e+3	3.7119e−06
	max	3.0000	0.3979	−1.0316	−3.8549	−2.6304	−2.7519	−50.0000	0.1033	1.5017	6.6928e+3	1.4371e−05
	mean	3.0000	0.3979	−1.0316	−3.8612	−6.4110	−7.1213	−50.0000	0.0291	0.3981	5.3050e+3	8.4438e−06
	std	9.8033e−10	3.3926e−11	9.0377e−11	4.0584e−11	3.8062	3.8207	9.1361e−08	0.0262	0.5855	6.9556e+2	2.9273e−06
TSA	min	3.0000	**0.3979**	−1.0316	−3.8628	−10.1532	−10.4029	−50.0000	0.0000	9.2271e−11	2.7491e+3	17.1932
	max	3.0000	**0.3979**	−1.0316	−3.8628	−9.6203	−10.4029	−50.0000	0.0075	1.3828e−09	7.1750e+3	52.1575
	mean	3.0000	**0.3979**	−1.0316	−3.8628	−10.1354	−10.4029	−50.0000	0.0005	5.7348e−10	5.0006e+3	34.6392
	std	1.0720e−15	**0.0000**	6.7752e−16	3.0633e−15	0.0973	1.4378e−15	5.3302e−14	0.0018	3.0317e−10	1.0133e+3	7.6499
SSO	min	3.0000	0.3979	−1.0316	−3.8628	−10.1532	−10.4029	−50.0000	9.1765e−09	1.3404	3.3983e+3	0.0810
	max	3.0000	0.3979	−1.0316	−3.8628	−2.6304	−2.7519	−50.0000	0.0811	3.9825	6.1978e+3	6.1711
	mean	3.0000	0.3979	−1.0316	−3.8628	−7.5555	−8.2684	−50.0000	0.0193	2.4520	5.0230e+3	1.0778
	std	1.7228e−13	4.6494e−15	1.1645e−14	1.6106e−14	3.1248	3.3612	8.0061e−12	0.0194	0.6667	7.6343e+2	1.2963
GWO	min	3.0000	0.3979	−1.0316	−3.8628	−10.1532	−10.4029	−50.0000	0.0000	1.1546e−14	4.9642e+3	1.6069e−22
	max	3.0000	0.3979	−1.0316	−3.8549	−2.6304	−5.1287	−50.0000	0.0676	2.2204e−14	8.6688e+3	1.0280e−18
	mean	3.0000	0.3979	−1.0316	3.8616	−9.5652	−10.2267	−50.0000	0.0005	1.6164e−14	6.7395e+3	1.1433e−19
	std	2.5862e−10	2.3793e−07	6.8685e−09	0.0026	1.8316	1.8315	5.4968e−05	0.0139	2.9724e−15	8.0006e+2	2.1892e−19
PFA	min	**3.0000**	**0.3979**	**−1.0316**	**−3.8628**	−10.1532	−10.4029	−50.0000	**0.0000**	**1.1546e−14**	**1.7173e+3**	1.0485
	max	**3.0000**	**0.3979**	**−1.0316**	**−3.8628**	−10.1531	−10.4029	−50.0000	**0.0030**	**1.5099e−14**	**4.2456e+3**	63.5780
	mean	**3.0000**	**0.3979**	**−1.0316**	**−3.8628**	−10.1532	−10.4029	−50.0000	**0.0006**	**1.4862e−14**	**3.1549e+3**	11.5480
	std	**2.4952e−16**	**0.0000**	**6.1157e−16**	**1.5026e−15**	1.3452e−04	5.9834e−10	1.9802e−11	**0.0012**	**9.0135e−16**	**5.6274e+2**	12.9802

The fitness independent methods such as PSO and GA does not require the modify the mechanism to use any kind of constraint processing. Since the position update mechanisms of proposed method is operated with respect to each location of agents, the agents are no direct related with the fitness function. Thus, the simplest constraint processing method, penalty terms, where the agents are assigned to high values if they violate any of the constraints, can be applied effectively to overcome constraints in problems. In a nutshell, when members of PFA violate constraints, they are replaced with the new one in the next iteration. The penalty method utilized for design problems is explained in Section 4.5. To optimize the design problem, population size with 60 individuals are employed and the maximum number of iterations is set to 100.

4.1. Tension/compression spring design

The main objective is to minimize the weight of tension/compression spring. The design problem has several constraints: shear stress, surge frequency and minimum deflection. The variables of this design problem are diameter (d), mean coil diameter (D) and number of active coil (P), where $\vec{x} = [d, D, P]$. The model is given below:

$$\text{Minimize } f\left(\vec{x}\right) = (x_3 + 2)\, x_2 {x_1}^2,$$

$$\text{Subject to } G_1 = 1 - \frac{{x_2}^3 x_3}{71785 {x_1}^4} \le 0,$$

$$G_2 = \frac{4{x_2}^2 - x_1 x_2}{12566\left({x_1}^3 x_2 - {x_1}^4\right)} - \frac{1}{5108 {x_1}^2} \le 0 \qquad (4.1)$$

Table 6
Results of composite functions.

Method	D	Val	CF1	CF2	CF3	CF4	CF5	CF6	CF7	CF8	CF9	CF10
PSO	10	min	1.0294e+2	2.1014e+1	3.2514e+2	3.5656e+2	4.3427e+2	3.7051e+2	4.0322e+2	3.8241e+2	2.6938e+2	4.9817e+4
		max	2.4448e+2	1.1512e+2	3.3075e+2	3.6495e+2	4.5846e+2	4.0000e+2	4.0643e+2	6.4068e+2	3.2200e+2	2.9199e+5
		median	2.1433e+2	1.1248e+2	3.2774e+2	3.5951e+2	4.4117e+2	3.8407e+2	4.0359e+2	4.0568e+2	3.0300e+2	1.4400e+5
		mean	1.7592e+2	1.2094e+2	3.2839e+2	3.6076e+2	4.4457e+2	3.8430e+2	4.0436e+2	4.8866e+2	2.9723e+2	1.4508e+5
		std	5.9118e+1	2.4478e+1	2.3539e+0	3.4057e+0	1.0521e+1	1.2357e+1	1.4407e+0	1.2671e+2	2.3019e+1	9.3900e+4
	30	min	2.8201e+2	5.3116e+2	4.8361e+2	5.6642e+2	4.3791e+2	1.4070e+3	5.1842e+2	6.0631e+2	5.9939e+2	9.6876e+4
		max	5.0772e+2	5.7059e+3	6.6309e+2	8.1657e+2	1.5510e+3	4.2088e+3	6.5849e+2	3.7707e+3	1.7062e+3	2.2744e+7
		median	3.6078e+2	3.7841e+3	5.7137e+2	6.8491e+2	6.4920e+2	3.4727e+3	5.6612e+2	9.5212e+2	1.1250e+3	1.7454e+6
		mean	3.6184e+2	3.3382e+3	5.7692e+2	6.8702e+2	6.8995e+2	3.3216e+3	5.7493e+2	1.2064e+3	1.1100e+3	4.1910e+6
		std	4.1586e+1	1.5171e+3	4.6196e+1	4.9839e+1	1.8620e+2	6.2747e+2	3.6418e+1	6.6578e+2	2.4312e+2	6.0620e+6
	50	min	4.8920e+2	6.3125e+3	8.7170e+2	8.8974e+2	1.2425e+3	6.0509e+3	7.1529e+2	2.2932e+3	1.6252e+3	5.0908e+6
		max	7.6190e+2	1.0709e+4	1.1435e+3	1.3405e+3	6.1013e+3	9.2452e+3	1.3760e+3	8.0864e+3	3.8700e+3	6.2712e+8
		median	5.9692e+2	8.2787e+3	9.9489e+2	1.1090e+3	3.0719e+3	7.3863e+3	1.0393e+3	5.9932e+3	2.6486e+3	5.3572e+7
		mean	6.0148e+2	8.3257e+3	9.9900e+2	1.1213e+3	3.0482e+3	7.5103e+3	1.0347e+3	5.5403e+3	2.6766e+3	9.7530e+7
		std	6.4544e+1	9.7951e+2	7.2411e+1	9.5724e+1	1.1808e+3	8.0689e+2	1.6360e+2	1.5346e+3	5.6979e+2	1.2151e+8
TLBO	10	min	1.0000e+2	1.2000e+1	3.0000e+2	1.0000e+2	4.0000e+2	0.0000e+0	3.9000e+2	3.0000e+2	2.5000e+2	1.1000e+3
		max	2.2000e+2	1.0000e+2	3.2000e+2	3.4000e+2	4.5000e+2	3.6000e+2	4.0000e+2	9.3000e+2	3.0000e+2	1.3000e+6
		median	1.4000e+2	9.3000e+1	3.1000e+2	3.3000e+2	4.4000e+2	3.0000e+2	3.9000e+2	3.7000e+2	2.7000e+2	9.1000e+3
		mean	1.1000e+2	1.0000e+2	3.1000e+2	3.1000e+2	4.3000e+2	3.0000e+2	3.9000e+2	4.5000e+2	2.7000e+2	2.8000e+5
		std	5.2000e+1	2.3000e+1	3.8000e+0	6.9000e+1	2.2000e+1	4.6000e+1	3.3000e+0	1.6000e+2	1.4000e−1	4.9000e+5
	30	min	2.2000e+2	1.0000e+2	3.7000e+2	4.5000e+2	3.8000e+2	2.0000e+2	5.0000e+2	3.8000e+2	5.0000e+2	4.1000e+3
		max	2.6000e+2	1.1000e+2	4.3000e+2	5.1000e+2	4.4000e+2	2.2000e+3	5.9000e+2	4.8000e+2	8.0000e+2	1e7000e+5
		median	2.3000e+2	1.0000e+2	3.9000e+2	4.7000e+2	3.9000e+2	1.5000e+3	5.3000e+2	4.3000e+2	5.9000e+2	2.6000e+4
		mean	2.3000e+2	1.0000e+2	4.0000e+2	4.7000e+2	4.0000e+2	1.4000e+3	5.3000e+2	4.5000e+2	6.2000e+2	1.9000e+4
		std	1.2000e+1	1.9000e+0	1.6000e+1	1.6000e+1	1.8000e+1	4.7000e+2	2.1000e+1	2.7000e+1	9.1000e+1	2.8000e+4
	50	min	2.4000e+2	1.0000e+2	4.9000e+2	5.7000e+2	5.6000e+2	8.3000e+2	5.9000e+2	5.4000e+2	4.6000e+2	7.2000e+5
		max	3.2000e+2	1.3000e+4	6.5000e+2	7.7000e+2	6.8000e+2	5.0000e+3	1.4000e+3	7.1000e+2	1.5000e+3	2.0000e+6
		median	2.8000e+2	1.2000e+4	5.7000e+2	6.7000e+2	6.2000e+2	2.8000e+3	8.7000e+2	6.1000e+2	1.0000e+3	1.0000e+6
		mean	2.8000e+2	6.6000e+3	5.7000e+2	6.7000e+2	6.2000e+2	2.9000e+3	8.8000e+2	6.1000e+2	1.0000e+3	1.2000e+6
		std	1.5000e+1	6.4000e+3	3.4000e+1	4.7000e+1	2.7000e+1	6.0000e+2	1.8000e+2	4.2000e+1	2.5000e+2	3.1000e+5
LSHADE-cnEpSin	10	min	1.0000e+2	1.0000e+2	3.0003e+2	1.0000e+2	3.9800e+2	3.0000e+2	3.8849e+2	3.0000e+2	2.2715e+2	3.9449e+2
		max	2.0516e+2	1.0028e+2	3.0398e+2	3.3319e+2	4.4333e+2	3.0000e+2	3.8889e+2	5.9420e+2	2.3038e+2	5.2227e+5
		median	2.0174e+2	1.0000e+2	3.0307e+2	3.2910e+2	3.9800e+2	3.0000e+2	3.8852e+2	5.3785e+2	2.2751e+2	3.9451e+2
		mean	1.5770e+2	1.0058e+2	3.0265e+2	2.9753e+2	4.1613e+2	3.0000e+2	3.8862e+2	4.5438e+2	2.2804e+2	1.0478e+5
		std	5.1677e+1	4.0700e−2	1.5479e+0	7.9588e+1	2.4824e+1	2.9546e−5	2.3090e−1	1.4273e+2	1.3238e+0	2.3338e+5
	30	min	2.0708e+2	1.0000e+2	3.4360e+2	4.2345e+2	3.8666e+2	7.2253e+2	4.9185e+2	3.0000e+2	4.1976e+2	1.9413e+3
		max	2.1639e+2	1.0000e+2	3.6298e+2	4.3430e+2	3.8669e+2	1.0408e+3	5.1900e+2	4.1397e+2	4.5620e+2	2.0944e+3
		median	2.1252e+2	1.0000e+2	3.5489e+2	4.2802e+2	3.8667e+2	9.5877e+2	5.0422e+2	3.0000e+2	4.3550e+2	1.9705e+3
		mean	2.1213e+2	1.0000e+2	3.5471e+2	4.2824e+2	3.8668e+2	9.5299e+2	5.0426e+2	3.0426e+2	4.3481e+2	1.9876e+3
		std	2.5977e+0	8.6131e−14	3.8419e+1	2.8401e+0	7.2651e−3	5.0865e+1	5.6522e+0	2.1322e+1	7.6455e+0	4.8218e+1
	50	min	2.1758e+2	1.0000e+2	4.1719e+2	5.0290e+2	4.7731e+2	1.0118e+3	5.0001e+2	4.5486e+2	3.3086e+2	5.7782e+5
		max	2.3634e+2	4.1546e+3	4.5301e+2	5.2632e+2	4.9031e+2	1.3577e+3	5.7569e+2	5.0628e+2	3.6773e+2	7.7850e+5
		median	2.2917e+2	1.6379e+2	4.4421e+2	5.1451e+2	4.8018e+2	1.2098e+3	5.2535e+2	4.5646e+2	3.4875e+2	6.3389e+5
		mean	2.2776e+2	1.7342e+3	4.3988e+2	5.1493e+2	4.8082e+2	1.2031e+3	5.2718e+2	4.6023e+2	3.4912e+2	6.4455e+5
		std	5.6449e+0	1.7855e+3	7.8114e+0	6.5373e+0	2.7237e+0	1.0132e+2	1.3597e+1	1.3418e+1	8.7863e+0	6.2245e+4
EBO	10	min	1.0000e+2	1.0000e+2	3.0001e+2	1.0000e+2	3.9774e+2	2.0000e+2	3.8951e+2	3.0000e+2	2.2880e+2	3.9451e+2
		max	2.0258e+2	1.0000e+2	3.0410e+2	3.2954e+2	4.4577e+2	3.0000e+2	3.9436e+2	6.1182e+2	2.3300e+2	4.4265e+2
		median	1.0000e+2	1.0000e+2	3.0001e+2	1.0000e+2	3.9959e+2	3.0000e+2	3.9075e+2	3.0000e+2	2.3014e+2	4.0743e+2
		mean	1.2639e+2	1.0000e+2	3.0019e+2	1.4590e+2	4.1998e+2	2.8000e+2	3.9167e+2	3.1478e+2	2.3069e+2	4.1894e+2
		std	4.4976e+1	0.0000e+0	8.0148e−1	1.0265e+2	2.2774e+1	4.4721e+1	2.4449e+0	6.3394e+1	1.9500e+0	2.2285e+1
	30	min	1.0000e+2	1.0000e+2	3.4572e+2	1.0000e+2	3.8356e+2	2.0000e+2	4.9560e+2	3.0000e+2	3.8549e+2	1.9420e+3
		max	2.0695e+2	1.0000e+2	3.5875e+2	4.2844e+2	3.8678e+2	1.0434e+3	5.1852e+2	4.1397e+2	4.5986e+2	2.0966e+3
		median	2.0312e+2	1.0000e+2	3.5047e+2	4.2467e+2	3.8673e+2	3.0000e+2	5.0547e+2	3.0000e+2	4.2976e+2	1.9723e+3
		mean	1.9917e+2	1.0000e+2	3.5134e+2	4.1388e+2	3.8648e+2	5.6759e+2	5.0514e+2	3.0426e+2	4.3020e+2	1.9868e+3
		std	2.0516e+1	8.6131e−14	3.2936e+0	5.5358e+1	8.3936e−1	3.2436e+2	4.9999e+0	2.1322e+1	1.2214e+1	3.3683e+1
	50	min	2.0405e+2	1.0000e+2	4.2060e+2	5.0116e+2	4.6143e+2	3.0000e+2	5.0876e+2	4.5884e+2	3.2541e+2	5.7941e+5
		max	2.1911e+2	4.0668e+3	4.4763e+2	5.1468e+2	5.6257e+2	1.1884e+3	6.2036e+2	5.0769e+2	4.0191e+2	7.2144e+5
		median	2.1231e+2	1.0000e+2	4.3632e+2	5.0777e+2	4.8031e+2	6.5868e+2	5.2395e+2	4.5884e+2	3.6352e+2	5.9528e+5
		mean	2.1193e+2	4.6444e+2	4.3479e+2	5.0784e+2	4.8364e+2	7.0487e+2	5.2685e+2	4.6731e+2	3.6340e+2	6.1371e+5
		std	3.6302e+0	1.1084e+3	6.3934e+0	3.3508e+0	1.2574e+1	4.1049e+2	1.7524e+1	1.8302e+1	1.8615e+1	3.8397e+4

(*continued on next page*)

$$G_3 = 1 - \frac{140.45x_1}{{x_2}^2 x_3} \leq 0$$

$$G_4 = \frac{x_1 + x_2}{1.5} \leq 0$$

where $0.05 \leq x_1 \leq 2$, $0.25 \leq x_2 \leq 1.3$ and $2 \leq x_3 \leq 15$.

This design problem has been used as a benchmark problem for testing different heuristic algorithm such as GWO [29], GA [67], Mine Blast (MBA) [70], co-evolutionary PSO [71], DE [72] and harmony search algorithm (HSA) [73]. The comparison of results of these algorithms have been listed in Table 11. Note that the results handled in original papers have been considered for other methods. The proposed method has been obtained better results than the majority of other methods and acquired very competitive results compared to MBA.

Table 6 (*continued*).

Method	D	Val	CF1	CF2	CF3	CF4	CF5	CF6	CF7	CF8	CF9	CF10
PFA	10	min	1.0000e+2	1.1563e+1	3.0014e+2	1.0006e+2	3.9774e+2	3.0000e+2	3.8848e+2	3.0000e+2	2.4124e+2	2.8612e+3
		max	2.2019e+2	1.0311e+2	3.1846e+2	3.4380e+2	4.4602e+2	3.0000e+2	3.9570e+2	4.4498e+2	3.2367e+2	9.2812e+4
		median	1.5467e+2	1.0137e+2	3.0762e+2	1.6547e+2	4.0078e+2	3.0000e+2	3.9090e+2	3.0277e+2	2.8414e+2	1.7656e+4
		mean	1.5645e+2	9.3593e+1	3.0706e+2	2.1330e+2	4.1863e+2	3.0000e+2	3.9175e+2	3.4087e+2	2.8418e+2	6.2390e+4
		std	5.5756e+1	2.4340e+1	4.3861e+0	9.8189e+1	2.1735e+1	2.1902e−5	1.9096e+0	6.3004e+1	1.5359e+1	1.5105e+4
	30	min	2.0778e+2	1.0000e+2	3.7323e+2	5.2936e+2	3.8300e+2	2.0782e+2	4.8804e+2	4.0528e+2	5.6256e+2	2.4685e+3
		max	4.2379e+2	7.57753+3	5.2140e+2	5.8619e+2	4.3544e+2	2.6365e+3	5.1048e+2	4.7741e+2	1.0563e+2	1.5743e+5
		median	3.1604e+2	1.9984e+3	4.9308e+2	5.7169e+2	3.8810e+2	9.5600e+2	5.0277e+2	4.3275e+2	6.8022e+2	5.7236e+4
		mean	3.2029e+2	3.3880e+3	4.8791e+2	5.7049e+2	3.8989e+2	9.5505e+2	5.0281e+2	4.3839e+2	7.1405e+2	6.4199e+4
		std	5.5865e+1	3.3643e+3	2.6840e+1	1.2125e+1	7.3578e+0	6.2841e+2	2.5284e+0	2.5585e+1	1.1731e+2	3.0136e+4
	50	min	2.9169e+2	1.0000e+2	5.1701e+2	5.5598e+2	5.4014e+2	3.5236e+2	5.0000e+2	5.1385e+2	4.6940e+2	1.1016e+6
		max	6.5178e+2	1.4177e+4	7.7342e+2	8.3818e+2	6.7421e+2	4.1024e+3	6.8090e+2	6.5809e+2	1.4784e+3	1.8261e+6
		median	5.6687e+2	1.3255e+4	7.2031e+2	8.1295e+2	6.0900e+2	2.0229e+3	6.2269e+2	5.5260e+2	8.7184e+2	1.4303e+6
		mean	5.1824e+2	1.3002e+4	7.0909e+2	7.9178e+2	6.1075e+2	2.2147e+3	6.2270e+2	5.6326e+2	9.0843e+2	1.4301e+6
		std	1.0989e+1	1.9255e+3	4.9710e+1	6.0605e+1	2.1206e+1	6.3598e+2	3.8000e+1	3.0291e+1	2.5833e+2	1.5398e+5

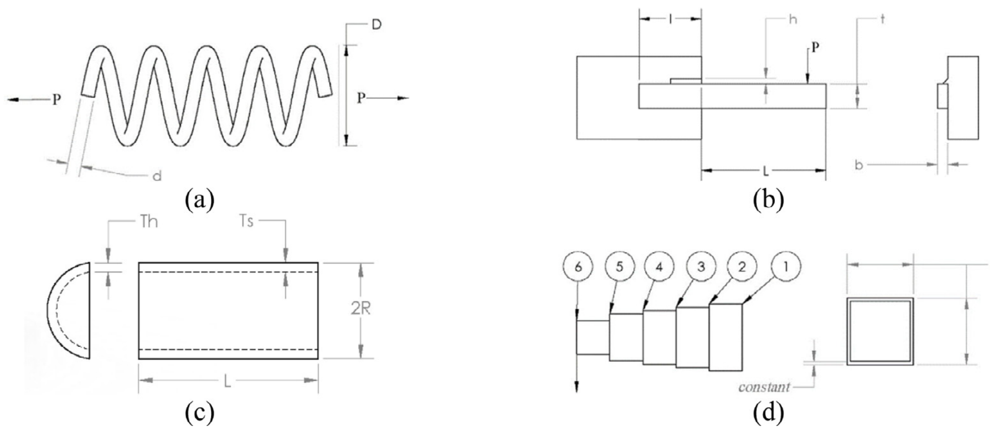

Fig. 9. The structure with parameters of design problems: (a) tension/compression spring, (b) welded beam, (c) pressure vessel, (d) cantilever beam.

4.2. Welded beam design

The main objective is to minimize the cost of fabrication of welded beam. This design problem has four variables: weld (h), length of bar (L), the height of bar (t) and the thickness of bar (b), where $\vec{x} = [h, L, t, b]$. The constraints of problems are shear stress, bending stress in the beam, buckling load on the bar, end deflection of beam and slide constraints. The model is given below:

$$\begin{aligned}
&\text{Minimize } f\left(\vec{x}\right) = 1.10471{x_1}^2x_2 + 0.04811x_3x_4\left(14 + x_2\right),\\
&\text{Subject to } G_1 = \tau\left(\vec{x}\right) - \tau_{\max} \le 0,\\
&\qquad G_2 = \sigma\left(\vec{x}\right) - \sigma_{\max} \le 0\\
&\qquad G_3 = \delta\left(\vec{x}\right) - \delta_{\max} \le 0\\
&\qquad G_4 = x_1 - x_4 \le 0\\
&\qquad G_5 = P - P\left(\vec{x}\right) \le 0\\
&\qquad G_6 = 0.125 - x_1 \le 0\\
&\qquad G_7 = 1.10471{x_1}^2x_2 + 0.04811x_3x_4\left(14 + x_2\right) - 5 \le 0
\end{aligned} \tag{4.2}$$

where $0.1 \le x_1 \le 2$, $0.1 \le x_2 \le 10$, $0.1 \le x_3 \le 10$ and $0.1 \le x_4 \le 2$. This problem has been solved with several methods in literature: GWO [29], MBA [70], GA [74,75], and HSA [76]. The results of this design problem have been given in Table 12. Note that the results handled in original papers have been considered for other methods. It can be seen that the proposed method outperforms the majority of algorithm and achieves the competitive results compared to MBA.

4.3. Pressure vessel design

The main objective of this problem is to minimize the cost of material, forming and welding of cylindrical vessel. There are four variables: thickness of the shell T_s, thickness of the head T_h, inner radius R and length of cylindrical section L, where $\vec{x} = [T_s, T_h, R, L]$. The model is given below:

$$\begin{aligned}
&\text{Minimize } f\left(\vec{x}\right) = 0.6224x_1x_3x_4 + 1.7781x_2{x_3}^2\\
&\qquad\qquad + 3.1661{x_1}^2x_4 + 19.84{x_1}^2x_3,\\
&\text{Subject to } G_1 = -x_1 + 0.0193x_3 \le 0,\\
&\qquad G_2 = -x_2 + 0.00954x_3 \le 0\\
&\qquad G_3 = -\pi{x_3}^2x_4 - \frac{4}{3}\pi{x_3}^3 + 1296000 \le 0\\
&\qquad G_4 = x_4 - 240 \le 0
\end{aligned} \tag{4.3}$$

where $0 \le x_1 \le 99$, $0 \le x_2 \le 99$, $10 \le x_3 \le 200$ and $10 \le x_4 \le 200$.

This problem has been solved with several meta-heuristic algorithms in literature: GWO [29], MBA [70], PSO [71], CS [26], MFA [31] and Crow Search Algorithm (CSA) [77]. The proposed method finds better design with the minimum cost than all other methods. The results of this problem have been given in Table 13.

4.4. Cantilever beam design

Another design problem is cantilever beam design problem, which has five hollow blocks and five variables: the height of beams, where $\vec{x} = [h_1, h_2, h_3, h_4, h_5]$. There is one constraint.

Table 7
Results of scalability test.

Method	Function	D	min	max	median	mean	std
PSO	F2	10D	4.2447e−27	100	6.9590e−25	10	30.5128
	F6		1.0856e−51	5.3297e−42	1.7440e−46	2.1631e−43	9.7043e−43
	F16		0.0001	1905.6328	1072.0133	1000.5458	388.9656
	F17		9.3888e−14	125.6466	7.0259e−10	15.07944	28.8952
	F2	50D	904.7604	11000.8403	5156.0940	5558.6444	2397.6315
	F6		65.5658	1600318404.8049	101197903.2131	294226910.3145	385005265.5139
	F16		6569.4208	12063.9980	9014.7501	9158.0437	1492.7467
	F17		916.7116	1999.0203	1550.5670	1502.3231	229.6023
	F2	100D	17299.9433	63791.1913	35962.6663	36179.9417	9525.6147
	F6		669690327.0969	11724828758.4578	4095585307.8181	4697741974.9872	2758087556.3658
	F16		19009.9635	25108.7557	22467.8653	22472.6586	1620.9985
	F17		3372.7416	4939.4089	4168.5196	4131.0041	376.4128
SSO	F2	10D	2.2501e−11	1.0909e−10	4.5106e−11	4.7238e−11	2.0041e−11
	F6		3.0259e−20	1.1505e−18	4.1728e−19	4.4248e−19	2.8872e−19
	F16		593.7088	1837.5579	1287.6359	1294.6152	332.0040
	F17	50D	3.7676e−12	2.2696e−11	1.1180e−11	1.1531e−11	4.5692e−12
	F2		0.3133	20.6354	2.7269	5.0163	5.1424
	F6		2.7888e−15	1.9618e−14	5.6524e−15	7.2154e−15	4.2433e−15
	F16		6439.9193	10309.3594	8683.1765	8414.2187	962.4984
	F17		63.1450	243.3201	134.3034	147.1610	54.1609
	F2	100D	40.5798	205.0871	130.3773	122.0197	48.5192
	F6		0.2311	146.8957	6.1052	17.7239	30.0594
	F16		14048.0088	21356.9875	17696.4417	17788.6188	1452.1164
	F17		841.7501	1685.6509	1248.3687	1255.7153	183.7683
TSA	F2	10D	3.6789e−123	1.5624e−118	2.1515e−120	8.6482e−120	2.8334e−119
	F6		1.9834e−243	1.0347e−235	2.4930e−238	8.4495e−237	0
	F16		0.0001	239.4813	0.0001	7.9831	43.7230
	F17	50D	1.7749e−40	1.5164e−36	2.6517e−38	8.9009e−38	2.7423e−37
	F2		0.0021	0.0156	0.0042	0.0047	0.0025
	F6		0.00027	0.0036	0.0018	0.0018	0.0010
	F16		6325.0866	13657.8426	11804.6825	11495.3638	1605.2337
	F17		234.1410	404.7082	332.8844	329.9800	43.1813
	F2	100D	6091.2470	9559.3982	7320.1312	7408.2857	795.3381
	F6		395585550.6684	1433976302.5436	799515755.4946	829001189.4847	243558349.3479
	F16		24829.3153	32468.0851	29591.9266	29447.2670	2000.8044
	F17		1265.7398	1559.9426	1438.2049	1421.1660	78.3756
GWO	F2	10D	2.1787e−124	3.4187e−117	1.3604e−120	1.7713e−118	6.3784e−118
	F6		1.1897e−246	1.7383e−228	6.7837e−239	5.7950e−230	0
	F16		927.8545	2041.8953	1547.1672	1498.4311	301.7438
	F17		1.1614e−80	2.2596e−70	2.0853e−73	1.4717e−71	4.3924e−71
	F2	50D	2.3197e−46	1.5300e−43	1.5379e−44	2.3680e−44	3.2385e−44
	F6		7.5762e−91	3.9299e−84	1.2010e−87	1.6114e−85	7.1627e−85
	F16		9261.0519	16031.4348	11641.6530	11768.1598	1135.9048
	F17		2.0405e−11	1.1181e−06	1.0779e−08	1.1812e−07	2.7613e−07
	F2	100D	5.4963e−31	7.4435e−29	5.9069e−30	8.1263e−30	1.3260e−29
	F6		2.9215e−60	1.9934e−55	1.5445e−58	6.9860e−57	3.6333e−56
	F16		2.1787e−124	3.4187e−117	1.3604e−120	1.7713e−118	6.3784e−118
	F17		1.1897e−246	1.7383e−228	6.7837e−239	5.7950e−230	0
PFA	F2	10D	4.1003e−90	1.5714e−85	2.5361e−87	2.2649e−86	4.1741e−86
	F6		8.8189e−177	6.3580e−163	2.6666e−171	2.1199e−164	0
	F16		118.4384	830.5855	473.7534	449.5090	184.2821;
	F17		9.83491e−40	6.0344e−34	4.0979e−37	2.1768e−35	1.0991e−34
	F2	50D	1.2172e−12	2.7914e−10	1.4841e−11	4.3007e−11	7.1002e−11
	F6		8.0450e−23	2.4234e−19	3.5437e−21	2.6305e−20	6.2253e−20
	F16		5073.2337	7544.1814	6504.5721	6423.8391	663.2207
	F17		167.3491	625.7501	339.98124	343.1244	92.1694
	F2	100D	0.0078	0.3733	0.0337	0.0570	0.0732
	F6		0.0011	1.2095	0.0474	0.1872	0.3443
	F16		10597.4553	16710.3890	13743.3170	13610.2227	1361.9760
	F17		1457.8049	2420.9106	1910.5574	1911.1759	195.2386

Also, the problem is related with weight minimization. The model of this design problem can be seen below:

$$\text{Minimize } f\left(\vec{x}\right) = 0.0624(x_1 + x_2 + x_3 + x_4 + x_5),$$
$$\text{Subject to } G_1 = \frac{61}{{x_1}^3} + \frac{37}{{x_2}^3} + \frac{19}{{x_3}^3} + \frac{7}{{x_4}^3} + \frac{1}{{x_5}^3} - 1 \le 0 \qquad (4.4)$$

where $0 \le x_i \le 100$.

The optimal weight of this problem obtained by PFA and similar method, MFA [31], SSO [32], CS [26] and symbiotic organisms search (SOS) [78], are given in Table 14. Despite the low difference, PFA outperforms all other methods and obtains the minimum weight, when solving this design problem.

Table 8
Results of computational time test, elapsed times are given as second.

Method	Function	Population	min time	mean time	max time
PSO	F1	30	0.354763	0.3830645	0.404093
	F3		0.425314	0.4356947	0.44198
	F15		0.703374	0.7524818	0.818051
	F16		0.550972	0.5752564	0.610333
	F1	100	0.875722	0.9138127	0.944053
	F3		1.099001	1.1386516	1.281287
	F15		1.945798	1.9898769	2.024430
	F16		1.656690	1.7220414	1.979167
	F1	300	2.552744	2.6196844	2.718674
	F3		3.229883	3.3249967	3.511942
	F15		5.868082	6.0217450	6.342816
	F16		5.161146	5.3887669	5.791918
SSO	F1	30	0.569353	0.6208386	0.706434
	F3		0.617705	0.6444076	0.742604
	F15		0.908416	0.9251964	0.964837
	F16		0.922814	0.9964000	1.134920
	F1	100	1.752447	1.8031404	1.883593
	F3		2.161454	2.2117021	2.356463
	F15		3.101318	3.2080623	3.407128
	F16		2.925737	3.0254770	3.249718
	F1	300	6.667234	6.9100228	7.154857
	F3		8.341033	8.8798041	9.419435
	F15		11.405729	11.7004107	12.138642
	F16		19.418519	19.9439654	20.922074
TSA	F1	30	1.961643	2.1653930	2.594023
	F3		2.678633	2.8761022	3.160214
	F15		4.223213	4.4821965	4.647719
	F16		4.133652	4.3066874	4.500889
	F1	100	18.789608	19.3958446	20.203326
	F3		23.845537	24.6402969	26.587980
	F15		38.732107	40.5563091	41.735138
	F16		37.816996	40.9681401	43.810925
	F1	300	170.026532	172.6349406	176.940422
	F3		218.032598	220.4026755	222.302654
	F15		360.025461	362.9196066	365.552633
	F16		371.005698	372.5000928	374.789718
GWO	F1	30	0.509657	0.5422037	0.635866
	F3		0.617234	0.6454354	0.706790
	F15		0.869734	0.8808276	0.899637
	F16		1.114421	1.1486347	1.316858
	F1	100	1.448781	1.5272568	1.730125
	F3		1.884329	1.9432750	2.130417
	F15		2.782846	2.9150391	3.061054
	F16		3.561494	3.7042943	3.947562
	F1	300	4.181230	4.2702566	4.411311
	F3		5.444921	5.7220681	5.958180
	F15		8.095746	8.4691856	8.909274
	F16		10.832390	11.0548878	11.671932
PFA	F1	30	0.391311	0.4137877	0.502639
	F3		0.460660	0.4683687	0.478929
	F15		0.720375	0.7292489	0.739353
	F16		0.640765	0.6815396	0.751944
	F1	100	1.037421	1.0640429	1.148811
	F3		1.282541	1.3046697	1.377884
	F15		2.068879	2.1249761	2.291435
	F16		2.038891	2.1085900	2.447016
	F1	300	3.044431	3.0982861	3.247555
	F3		3.884427	3.9275214	4.007580
	F15		6.186482	6.2886223	6.495169
	F16		5.720642	5.7863550	5.967194

4.5. Constraint processing method

A commonly used idea is to determine a penalty function, so that the constrained problem is constructed as an unconstrained problem [79]. Now we defined:

$$F\left(x, m_i, v_j\right) = f\left(x\right) + \sum_{i}^{M} m_i {\varphi_i}^2 + \sum_{j}^{V} v_j {\omega_j}^2 \tag{4.5}$$

where m_i and v_j are the weights, φ_i is equality constraints and ω_j is inequality constraints. In this case, m_i and v_j should be at sufficient value, depending on the quality needed, where $m_i \gg 1$ and $v_j \geq 0$. When an equality constraint is provided, its effect to F is zero. On the other hand, if it is violated, it then penalized heavily. Similarly, it is valid for inequality constraints when they get tight and exact. For numerical implementation, an index function H is used to rewrite the above function as given

Table 9
Multi-objective functions.

Test	Function model
ZDT1	Minimize $f_1(x) = x_1$ Minimize $f_2(x) = g(x)\, h(f_1(x), g(x))$ where $G(x) = 1 + \frac{9}{N-1}\sum_{i=2}^{N} x_i$ $h(f_1(x), G(x)) = 1 - \sqrt{\frac{f_1(x)}{g(x)}}$ $0 \le x_i \le 1, \quad i = 1, 2, \ldots, 30$
ZDT2	Minimize $f_1(x) = x_1$ Minimize $f_2(x) = g(x)\, h(f_1(x), g(x))$ where $G(x) = 1 + \frac{9}{N-1}\sum_{i=2}^{N} x_i$ $h(f_1(x), G(x)) = 1 - \left(\frac{f_1(x)}{g(x)}\right)^2$ $0 \le x_i \le 1, \quad i = 1, 2, \ldots, 30$
ZDT3	Minimize $f_1(x) = x_1$ Minimize $f_2(x) = g(x)\, h(f_1(x), g(x)) + 1$ where $G(x) = 1 + \frac{9}{N-1}\sum_{i=2}^{N} x_i$ $h(f_1(x), G(x)) = 1 - \sqrt{\frac{f_1(x)}{g(x)}} - \left(\frac{f_1(x)}{g(x)}\right)\sin(10\pi f_1(x))$ $0 \le x_i \le 1, \quad i = 1, 2, \ldots, 30$
ZDT4	Minimize $f_1(x) = x_1$ Minimize $f_2(x) = x_2$ Minimize $f_3(x) = g(x)\, h(f_1(x), g(x)) h(f_2(x), g(x))$ where $G(x) = 1 + \frac{9}{N-1}\sum_{i=3}^{N} x_i$ $h(f_1(x), G(x)) = 1 - \sqrt{\frac{f_1(x)}{g(x)}}, \quad h(f_2(x), G(x)) = 1 - \sqrt{\frac{f_2(x)}{g(x)}}$ $0 \le x_i \le 1, \quad i = 1, 2, \ldots, 30$

below:

$$F\left(x, m_i, \upsilon_j\right) = f(x) + \sum_{i}^{M} m_i H_i(\varphi_i)\, {\varphi_i}^2 + \sum_{j}^{V} \upsilon_j H_j\left(\omega_j\right) {\omega_j}^2 \qquad (4.6)$$

If $\varphi_i \neq 0$ then $H_i(\varphi_i) = 1$, and if $\varphi_i = 0$ then $H_i(\varphi_i) = 0$. Similarly, if $\omega_j > 0$ then $H_j\left(\omega_j\right) = 1$, and if $\omega_j \le 0$ then $H_j\left(\omega_j\right) = 0$. Generally, in computational methods, m_i and υ_j are selected around 10^{15}. In this paper, we preferred m_i and υ_j as 10^{11} and 10^{15} respectively.

In summary, the results obtained on the design problems show that PFA demonstrates high performance and is capable in solving challenging problems. This is due to the parameters of PFA which allow it to find global optimum in high accuracy and to avoid local optima.

5. Optimal Placement and Sizing of Renewable Energy Sources (RESs)

In this section, the proposed method has been implemented to optimal placement and sizing problem with four objectives: power loss minimization, voltage deviation optimization, minimization of gas emission and cost minimization. Various algorithms have been applied to optimal placement and sizing of renewable energy sources (RESs) or distributed generations (DGs) in electrical power systems. Optimal placement and sizing of RESs is aimed to optimize the several objective functions such as power loss, cost optimization, emission minimization of power system, maximization of energy efficiency and optimization of voltage deviation subject to system operating constraints. Analytic methods, genetic algorithm (GA), PSO, honey bee mating optimization algorithm (HBMO), ant lion optimization algorithm (ALOA), CS, BBBC and ABC and its variation have been handled

Table 10
Results for multi-objective functions.

Test			MOPSO	MOSSO	MOPFA
ZDT1	Spacing	min	0.0088	0.0101	0.0073
		max	0.0115	0.0137	0.0108
		mean	0.0101	0.0122	0.0085
		std	0.0010	0.0011	0.0013
	Gen. Dist.	min	0.0100	0.0290	0.0010
		max	0.0400	0.0460	0.0016
		mean	0.0385	0.0354	0.0014
		std	0.0040	0.0052	2.4199e−04
ZDT2	Spacing	min	0.0090	0.0114	0.0076
		max	0.0164	0.0178	0.0166
		mean	0.0107	0.0122	0.0107
		std	0.0018	0.0011	0.0026
	Gen. Dist.	min	0.0100	0.0258	6.1298e−05
		max	0.0395	0.0405	0.0015
		mean	0.0259	0.0301	0.0011
		std	0.0040	0.0058	4.5457e−04
ZDT3	Spacing	min	0.0075	0.0060	0.0043
		max	0.0090	0.0082	0.0062
		mean	0.0088	0.0071	0.0053
		std	2.5875e−04	6.8857e−04	5.5559e−04
	Gen. Dist.	min	0.0316	0.0388	0.0269
		max	0.0333	0.0433	0.0295
		mean	0.0324	0.0424	0.0285
		std	5.0025e−04	5.9855e−04	9.3023e−04
ZDT4	Spacing	min	0.0658	0.0702	0.0538
		max	0.0788	0.0855	0.0747
		mean	0.0701	0.0751	0.0612
		std	0.0065	0.0071	0.0060
	Gen. Dist.	min	0.0130	0.0158	0.0123
		max	0.0230	0.0258	0.0183
		mean	0.0162	0.0198	0.0148
		std	0.0033	0.0042	0.0021

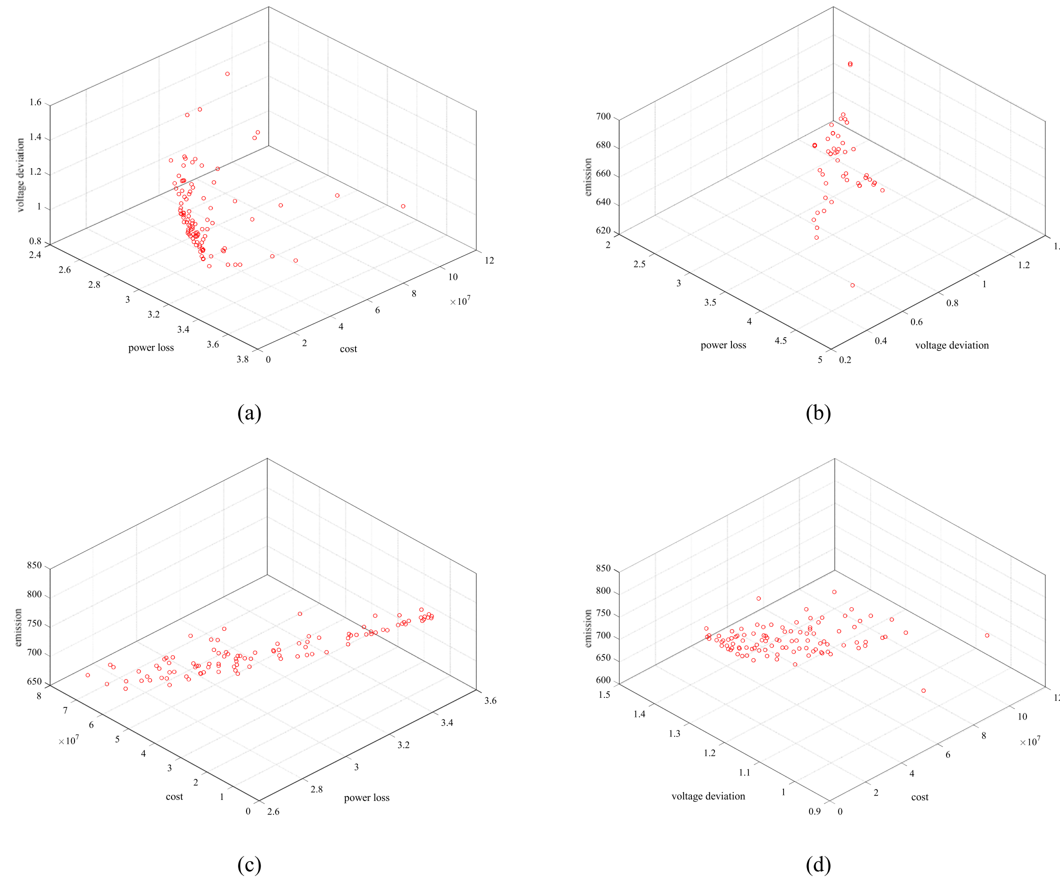

Fig. 10. The Pareto front obtained in tests of three objectives, (a) first case, it includes optimization of power loss, cost and voltage deviation, (b) second case, it includes optimization of power loss, voltage deviation and emission, (c) third case includes optimization of power loss, cost and emission, (d) fourth case includes voltage deviation, cost and emission.

Table 11
The comparison results of tension/compression design problem.

Method	d	D	P	Weight
GWO [12]	0.05169	0.356737	11.28885	0.012666
MBA [68]	0.051656	0.355940	11.344665	0.012665
c-PSO [69]	0.051728	0.357644	11.244543	0.0126747
GA [65]	0.051480	0.351661	11.632201	0.0127048
DE [70]	0.051609	0.354714	11.410831	0.0126702
HSA [71]	0.051154	0.349871	12.076432	0.0126706
Proposed method	0.05172695	0.3576296	11.235724	0.01266528

for solving optimal placement and sizing of RES problem in radial distribution systems and large-scale power systems [80–90]. Also, many other methods have been proposed for this problem in same or different type of test systems [91–107].

The proposed algorithm has been implemented to solve the optimal placement and sizing problem in IEEE 30-bus test system. The data for this test systems is detailed in [108]. In this case study, three hybrid PV (photovoltaic) - Wind power plant has been employed. In a hybrid station, half of the power is supplied from the PV units and the other half from the Wind units. The main goal is to determine the optimal size (MW) and location (bus no). In addition, we assumed that the sunshine duration is 12 h (PV units can generate power only from 6:00 am to 6:00 pm) for PV units, and wind units are turned by the wind at the average speed in 24 h (wind turbines can generate power all hour in each day).

It is worth mentioned that Each individual of the proposed algorithm is a vector in the search space as given in Eq. (5.1). The variables handled in this section are integer or discrete. However, the proposed method searches the promising solutions in the continuous space regardless of the types of variables same as the other algorithms. To handle these variables in the performing of evaluation of objective function, each individual is searched space continuously and it is then interrupted into the corresponding dimensions of these discrete variables.

$$x_i = \left[L^1,\ \ldots L^n,\ P^1,\ \ldots P^n\right],$$
$$i \in population\ size,\ \ n \in number\ of\ power\ plant \qquad (5.1)$$

where x_i is the vector of ith individual, n is the number of RES power plants, L is the location of RES installed and P is the output power of the power plant. It must be noted that the optimal power flow has been run in each iteration.

To find optimal Pareto front of this problem, population size is set to 100 and the proposed method is run over 100 iterations. The maximum box size is adjusted to 100 as well. Note that for this case study, MATPOWER software [109,110] is used to calculate optimal power flow for objectives. Because of the software used cannot illustrate the four objectives together, to show Pareto front, four cases have been performed. Each case includes three objectives and then the optimal front obtained by MOPFA is given in Fig. 10. First case indicates the power loss, voltage deviation and cost minimization, second case is carried out with power loss, voltage deviation and gas emission, third case includes the power loss, cost minimization and gas emission, and the final case utilizes the voltage deviation, cost minimization and gas emission.

The mathematical formulation of objective functions and constraints are given below [85]:

Minimize

Table 12
The comparison results of welded beam design problem.

Method	h	L	t	b	Cost
GWO [12]	0.205676	3.478377	9.03681	0.205778	1.72624
MBA [68]	0.205729	3.470493	9.036626	0.205729	1.724853
GA [72]	–	–	–	–	1.8245
GA [73]	0.2489	6.1730	8.1789	0.2533	2.4331
HSA [74]	0.2442	6.2231	8.2915	0.2443	2.3807
Proposed method	0.2057295	3.470495	9.036624	0.2057297	1.7248530

Table 13
The comparison results of pressure vessel design problem.

Method	T_s	T_h	R	L	Cost
GWO [12]	0.812500	0.434500	42.089181	176.758731	6051.5639
MBA [68]	0.7802	0.3856	40.4292	198.4964	5889.3216
PSO [69]	0.8125	0.4375	42.09127	176.7465	6061.0777
CS [10]	0.8125	0.4375	42.09845	176.636596	6059.71434
MFA [15]	0.8125	0.4375	42.098445	176.636596	6059.7143
CSA [75]	0.812500	0.434500	42.098445	176.636599	6059.7144
Proposed method	0.7781684	0.3846489	40.31964	199.9999	5885.3351

Table 14
The comparison results of cantilever beam design problem.

Method	h_1	h_2	h_3	h_4	h_5	Weight
MFA [15]	5.98487	5.3167269	4.49733	3.5136165	2.161620	1.3399881
SSO [16]	6.0151345	5.3093047	4.4950067	3.501426	2.1527879	1.33995639
CS [10]	6.0089	5.3049	4.5023	3.5077	2.1504	1.33999
SOS [76]	6.01878	5.30344	4.49587	3.49896	2.15564	1.33996
Proposed method	6.0154633	5.30902227	4.494631457	3.50178505	2.152757831	1.33995638

Table 15
Economic specifications (in 2016).

Technology type	Mean installed cost ($/kW)	Installed cost Std. Dev. (+/−$/kW)	Fixed O&M ($/kW-yr)	Fixed O&M Std. Dev. (+/−$/kW-yr)	Lifetime (yr)	Lifetime Std. Dev. (yr)	Fuel and/or water cost ($/kWh)	Fuel and/or water Std. Dev. ($/kWh)
PV <10 kW	$3,897	$889	$21	$20	33	11	–	–
PV 10–100 kW	$3,463	$947	$19	$18	33	11	–	–
PV 100–1,000 kW	$2,493	$774	$19	$15	33	11	–	–
PV 1–10 MW	$2,025	$694	$16	$9	33	9	–	–
Wind <10 kW	$7,645	$2,431	$40	$34	14	9	–	–
Wind 10–100 kW	$6,118	$2,101	$35	$12	19	5	–	–
Wind 0.1–1 MW	$3,751	$1,376	$31	$10	16	0	–	–
Wind 1–10 MW	$2,346	$770	$33	$16	20	7	–	–

Table 16
The coefficient of thermal units of 30-bus test systems.

Generator index	α	β	γ	ξ	λ
1	0.06490	−0.05554	0.04091	2e−4	2.8570
2	0.05638	−0.06047	0.02543	5e−4	3.3330
3	0.04586	−0.05094	0.04258	1e−6	8.0000
4	0.03380	−0.03550	0.05326	2e−3	2.0000
5	0.04586	−0.05094	0.04258	1e−6	8.0000
6	0.05151	−0.05555	0.06131	1e−5	6.6670

$$F_1(\vec{x}) = \min\left(\sum_{i=1}^{NB}\left(|I_i|^2 R_i\right)\right),$$

$$F_2(\vec{x}) = \min\left(\sum_{i=1}^{NB}\frac{|V_n - V_i|}{V_n}\right),$$

$$F_3(\vec{x}) = \min\left(E^{pv} + E^{w} + E^{hybrid} + \sum E^{thermal}\right),$$

$$E = \sum_j \alpha_j P_j^2 + \beta_j P_j + \gamma_j + \xi_j \exp\left(\lambda_j P_j\right)$$

$$F_2(\vec{x}) = \min\left(\sum_{k=1}^{Nk} C_k(P_k) + Cs\right),$$

$$C(P) = a + b \times p, \tag{5.2}$$

$$a = \frac{Capital\ Cost\ (\$/\mathrm{kW}) \times Capacity\ (\mathrm{kW}) \times Ri}{Life\ time\ \times 365 \times 24 \times L}$$

$$b = Fuel\ Cost\ (\$/\mathrm{kWh}) \times OM\ (\ /\mathrm{kWh})$$

Subject to

$$V_{imin} \le V_i \le V_{imax}$$

$$S < S_{max}$$

$$\sum_{k=1}^{nres} kW_k^{RES} \le \mu P_{load}$$

where, i is the index of bus, NB is the number of bus, I is the current of the line, R is the resistance of the line, V_n is the nominal voltage, V_i is the effective voltage of bus, j is the index power plants, P is the output power of each power plant, α, β, γ, ξ and λ are the emission coefficients, E is total emission of each power plant, Ri is the annual interest rate, L is the load factor, OM is the operation and maintenance cost and Cs is sum of power rating of station and $C(P)$ total installation cost of power plants. The specification of economic parameters [111] and coefficients of emission are tabulated in Tables 15 and 16, respectively. Note that RES units do not directly emit any gases. For this reason, coefficients of emission are considered for only thermal units.

Table 17
The results of optimal placement and sizing problem for 30-bus test system.

Method	F_1(MW)	F_2($)	F_3	F_4(ton/h)	Bus/Power of hybrid plant (MW)	Bus/Power (Wind) (MW)	Bus/ Distributed generation (MW)
ABC [110]	8.716	–	–	–	–	24,26/4.5	–
Fuzzy-PSO [111]	–	–	0.1231	905.31	–	–	–
N-R [112]	13.610	–	–	–	–	–	11/35
Fuzzy-GA [113]	7.158	–	–	–	–	14,15,23/6 8,21/10 26,29/2 30/9	–
MOPFA	3.1225	4.4142e+07	1.2726	698.1176	2/0.5291 28/19.7391 26/0.8410	–	–

This problem has been solved with several methods for 30-bus test system: ABC [112], fuzzy-PSO [113], Newton–Raphson based approach [114] and fuzzy-GA [115]. Moreover, the results acquired in this problem are listed in Table 17. This results and findings again point out the superiority of the proposed method. Despite many objectives, MOPFA has been achieved best results with minimum value. This is again due to the adaptive parameters of the proposed method. It has to be noted that this problem is solved in one run.

In addition, in Fig. 10, optimal Pareto front solutions are well distributed between both objective functions. The true Pareto front of this problem is unknown; therefore, it is not possible to say about how close to true Pareto front. The results of this study case also point out that the model of the proposed method can be very efficient to find optimal solutions in challenging problems with unknown search space.

6. Conclusions

This study proposed a new swarm-based meta-heuristic method to solve optimization problems. The proposed method mimicked the collective movements of swarms with using the hierarchy between leader and other members of swarm. Two separated mathematical formulation were used for position updating of leader and other members. The proposed method simulated in different test beds. The simulations in 2D and 3D space proved that the model presented can be capable for searching around optimal solutions. The model then implemented to single-objective and multi-objective problems. The best solutions so far were handled as the optimum to be followed by the swarm. Also, the parameters of PFA performed the exploration and exploitation and facilitated the transition between both phases with using integrated adaptive model. These parameters were then easy adapted to multi-objective test beds. For multi-objective optimization, a supply box, like archive of MOPSO and repository of MOSSO, were integrated for storing the non-dominated solutions.

To show and prove the performance and efficient, the proposed method was applied some tests. PFA was tested on some benchmark functions including unimodal, multimodal and composite functions. In particular, in experiment on CEC2017, it obtained challenging results, despite of two very effective methods. It was then observed that PFA is able to explore the promising solutions, provide the abrupt changing in the initial iterations, and exploit the best one over the course of iterations. All results were compared with well-known methods. PFA outperformed the majority of these methods in a statistically expressive manner. So, it may be concluded that PFA is capable to find the global optimum in challenging problems.

Then, the proposed model was designed for multi-objective problems and tested on several functions. The results obtained in test problems were compared with MOPSO and MOSSO. As per results, it can be seen that MOPFA can close to the true optimal Pareto front.

Moreover, despite the fact that four objectives, MOPFA acquired very effective results in real engineering problem. According to results obtained in this problem, it can also achieve effective results in that problems with unknown spaces.

As in all studies, there are some limitations of this study. According to NFL theorem, an algorithm cannot show a superior performance for all optimization problems. The proposed algorithm in this study provides a superior performance in some optimization problems. When the dimension of a problem is extremely increased, the performance of this method decreases. Furthermore, when the population size in a problem is increased, the computational time of this algorithm is extended. Besides, if the number of iterations is increased, finding of new promising solutions will be difficult because of fluctuation rate A and vibration vector ε converging to 0.

To sum up, it can be pointed out that the proposed method is noteworthy among the methods in literature and it can also be implemented to other problems in different areas. In addition, investigating the effects of different approaches and operators on the performance the proposed algorithm, such as mutation operator and binary version, is recommended. So that, these can provide the precious contributions to both PFA and its multi-objective version.

References

[1] T. Back, Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms, Oxford University press, 1996.
[2] B. Webster, P.J. Bernhard, A local search optimization algorithm based on natural principles of gravitation, 2003.
[3] G. Beni, J. Wang, Swarm Intelligence in Cellular Robotic Systems. Robots and Biological Systems: Towards a New Bionics, Springer, 1993, pp. 703–712.
[4] S. Mirjalili, S.Z.M. Hashim, A new hybrid PSOGSA algorithm for function optimization, in: Computer and Information Application, ICCIA, 2010 International Conference on, IEEE, 2010, pp. 374–377.
[5] L. Lin, M. Gen, Auto-tuning strategy for evolutionary algorithms: balancing between exploration and exploitation, Soft Comput. 13 (2009) 157–168.
[6] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Trans. Evol. Comput.. 1 (1997) 67–82.
[7] D.E. Goldberg, Holland J.H., Genetic algorithms and machine learning, Mach. Learn. 3 (1988) 95–99.
[8] R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Optim. 11 (1997) 341–359.
[9] X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster, IEEE Trans. Evol. Comput. 3 (1999) 82–102.
[10] O.K. Erol, I. Eksin, A new optimization method: big bang–big crunch, Adv. Eng. Softw. 37 (2006) 106–111.
[11] E. Rashedi, H. Nezamabadi-Pour, S. Saryazdi, GSA: a gravitational search algorithm, Inform. Sci. 179 (2009) 2232–2248.
[12] A. Kaveh, S. Talatahari, A novel heuristic optimization method: charged system search, Acta Mech. 213 (2010) 267–289.
[13] R.A. Formato, Central force optimization, Prog. Electromagn. Res. 77 (2007) 425–491.
[14] B. Alatas, ACROA: artificial chemical reaction optimization algorithm for global optimization, Expert Syst. Appl. 38 (2011) 13170–13180.

[15] A. Hatamlou, Black hole: A new heuristic optimization approach for data clustering, Inform. Sci. 222 (2013) 175–184.
[16] A. Kaveh, M. Khayatazad, A new meta-heuristic method: ray optimization, Comput. Struct. 112 (2012) 283–294.
[17] H. Du, X. Wu, J. Zhuang, Small-World Optimization Algorithm for Function Optimization. International Conference on Natural Computation, Springer, 2006, pp. 264–273.
[18] H. Shah-Hosseini, Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimisation, Int. J. Comput. Sci. Eng. 6 (2011) 132–140.
[19] F.F. Moghaddam, R.F. Moghaddam, M. Cheriet, Curved space optimization: A random search based on general relativity theory, 2012, arXiv preprint arXiv:12082214.
[20] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of ICNN'95 - International Conference on Neural Networks, vol. 4, 1995, pp. 1942–1948.
[21] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (abc) algorithm, J. Glob. Optim. 39 (2007) 459–471.
[22] M. Dorigo, V. Maniezzo, A. Colorni, Ant system: optimization by a colony of cooperating agents, IEEE Trans. Syst. Man Cybern. B 26 (1996) 29–41, http://dx.doi.org/10.1109/3477.484436.
[23] X.-S. Yang, Firefly Algorithms for Multimodal Optimization. International Symposium on Stochastic Algorithms, Springer, 2009, pp. 169–178.
[24] A.H. Gandomi, A.H. Alavi, Krill herd: a new bio-inspired optimization algorithm, Commun. Nonlinear Sci. Numer. Simul.. 17 (2012) 4831–4845.
[25] X.-S. Yang, A New Metaheuristic Bat-Inspired Algorithm. Nature Inspired Cooperative Strategies for Optimization, NICSO 2010, Springer, 2010, pp. 65–74.
[26] A.H. Gandomi, X.-S. Yang, A.H. Alavi, Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems, Engi. Comput. 29 (2013) 17–35.
[27] S.A. Uymaz, G. Tezel, E. Yel, Artificial algae algorithm (aaa) for nonlinear global optimization, Appl. Soft Comput. 31 (2015) 153–171.
[28] M.S. Kiran, TSA: Tree-seed algorithm for continuous optimization, Expert Syst. Appl. 42 (2015) 6686–6698.
[29] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Soft. 69 (2014) 46–61.
[30] J. James, V.O. Li, A social spider algorithm for global optimization, Appl. Soft Comput. 30 (2015) 614–627.
[31] S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, Knowl.-Based Syst. 89 (2015) 228–249.
[32] S. Mirjalili, A.H. Gandomi, S.Z. Mirjalili, S. Saremi, H. Faris, S.M. Mirjalili, Salp swarm algorithm: A bio-inspired optimizer for engineering design problems, Adv. Eng. Softw. 114 (2017) 163–191.
[33] S. Mirjalili, A. Lewis, The whale optimization algorithm, Adv. Eng. Softw. 95 (2016) 51–67.
[34] A. Kaveh, N. Farhoudi, A new optimization method: dolphin echolocation, Adv. Eng. Softw. 59 (2013) 53–70.
[35] S.-C. Chu, P.-W. Tsai, Pan J.-S., CaT swarm optimization, in: Pacific Rim International Conference on Artificial Intelligence, Springer, 2006, pp. 854–858.
[36] M. Yazdani, F. Jolai, Lion optimization algorithm (LOA): a nature-inspired metaheuristic algorithm, J. Comput. Des. Eng. 3 (2016) 24–36.
[37] P.C. Lee, C.J. Moss, Wild female African elephants (Loxodonta africana) exhibit personality traits of leadership and social integration, J. Compar. Psychol. 126 (2012) 224.
[38] R.O. Peterson, A.K. Jacobs, T.D. Drummer, L.D. Mech, D.W. Smith, Leadership behavior in relation to dominance and reproductive status in gray wolves, Canis lupus, Canad. J. Zool. 80 (2002) 1405–1412, [Online]. Available: https://doi.org/10.1139/z02-124.
[39] I.D. Couzin, J. Krause, R. James, G.D. Ruxton, N.R. Franks, Collective memory and spatial sorting in animal groups, J. Theoret. Biol. 218 (2002) 1–11.
[40] T. Vicsek, A. Zafeiris, Collective motion, Phys. Rep. 517 (2012) 71–140.
[41] B. Ferdinandy, K. Ozogány, T. Vicsek, Collective motion of groups of self-propelled particles following interacting leaders, Physica A 479 (2017) 467–477.
[42] R. Lukeman, Y.-X. Li, L. Edelstein-Keshet, Inferring individual rules from collective behavior, Proc. Natl. Acad. Sci. (2010) 201001763.
[43] I. Ahmed, D.Q. Ly, W. Ahmed, Collective behavior of self-propelled particles in the presence of moving obstacles, Mater. Today Proc.. 4 (2017) 65–74.
[44] D.S. Cambui, A.S. de Arruda, M. Godoy, Critical exponents of a self-propelled particles system, Physica A 444 (2016) 582–588.
[45] I. Giardina, Collective behavior in animal groups: theoretical models and empirical studies, HFSP J. 2 (2008) 205–219.
[46] K. Socha, M. Dorigo, Ant colony optimization for continuous domains, Eur. J. Oper. Res. 185 (2008) 1155–1173.
[47] K. Krishnan, D. Ghose, Detection of multiple source locations using a glowworm metaphor with applications to collective robotics, in: Swarm Intelligence Symposium, 2005 SIS 2005 Proceedings 2005 IEEE, IEEE, 2005, pp. 84–91.
[48] A.K. Qin, X. Li, Differential evolution on the CEC-2013 single-objective continuous optimization testbed, in: Evolutionary Computation, CEC, 2013 IEEE Congress on, IEEE, 2013, pp. 1099–1106.
[49] A.Y. Lam, V.O. Li, J. James, Real-coded chemical reaction optimization, IEEE Trans. Evol. Comput. 16 (2012) 339–353.
[50] K. Price, R.M. Storn, J.A. Lampinen, Differential Evolution: A Practical Approach To Global Optimization, Springer Science & Business Media, 2006.
[51] W.-N. Chen, J. Zhang, Y. Lin, N. Chen, Z.-H. Zhan, H.S.-H. Chung, et al., Particle swarm optimization with an aging leader and challengers, IEEE Trans. Evol. Comput. 17 (2013) 241–258.
[52] A.K. Qin, V.L. Huang, P.N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, IEEE Trans. Evol. Comput. 13 (2009) 398–417.
[53] M. Reyes-Sierra, C.C. Coello, Multi-objective particle swarm optimizers: A survey of the state-of-the-art, Int. J. Comput. Intell. Res. 2 (2006) 287–308.
[54] C.A.C. Coello, G.T. Pulido, M.S. Lechuga, Handling multiple objectives with particle swarm optimization, IEEE Trans. Evol. Comput. 8 (2004) 256–279.
[55] The benchmark functions, 2017, [Online] Available at: https://www.sfu.ca/~ssurjano/optimization.html.
[56] M. Jamil, X.-S. Yang, A literature survey of benchmark functions for global optimisation problems, Int. J. Math. Model. Numer. Optim. 4 (2013) 150–194, http://dx.doi.org/10.1504/IJMMNO.2013.055204.
[57] N.H. Awad, M.Z. Ali, J.J. Liang, Qu. B.Y., P.N. Suganthan, Problem Definitions and Evaluation Criteria for the CEC 2017 Special Session and Competition on Single Objective Bound Constrained Real-Parameter Numerical Optimization, Technical Report, Nanyang Technological University, Singapore, November 2016, [Online]. Available at: http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC2017/CEC2017.htm.
[58] R. Kommadath, P. Kotecha, Teaching learning based optimization with focused learning and its performance on ceC2017 functions, in: Evolutionary Computation, CEC, 2017 IEEE Congress on, IEEE, 2017, pp. 2397–2403, http://dx.doi.org/10.1109/CEC.2017.7969595.
[59] A. Kumar, R.K. Misra, D. Singh, Improving the local search capability of effective butterfly optimizer using covariance matrix adapted retreat phase, in: Evolutionary Computation, CEC, 2017 IEEE Congress on, IEEE, 2017, pp. 1835–1842, http://dx.doi.org/10.1109/CEC.2017.7969524.
[60] N.H. Awad, M.Z. Ali, P.N. Suganthan, Ensemble sinusoidal differential covariance matrix adaptation with euclidean neighborhood for solving CEC2017 benchmark problems, in: Evolutionary Computation, CEC, 2017 IEEE Congress on, IEEE, 2017, pp. 372–379, http://dx.doi.org/10.1109/CEC.2017.7969336.
[61] M. Molga, C. Smutnicki, Test functions for optimization needs, 2005, p. 101.
[62] X.-S. Yang, Test problems in optimization, 2010, arXiv preprint arXiv:1008.0549.
[63] E. Zitzler, K. Deb, L. Thiele, Comparison of multiobjective evolutionary algorithms: Empirical results, Evol. Comput. 8 (2000) 173–195.
[64] S. Mostaghim, J. Teich, Strategies for finding good local guides in multi-objective particle swarm optimization, mopso, in: Swarm Intelligence Symposium, 2003 SIS'03 Proceedings of the 2003 IEEE, IEEE, 2003, pp. 26–33.
[65] D.A. Van Veldhuizen, Multiobjective Evolutionary Algorithms: Classifications, Analyzes, and New Innovations, (Ph.D dissertation), Dept. Elec. Comput. Eng. Graduate School of Eng. Air Force Inst. Technol. Wright-Patterson AFB, OH, May 1999.
[66] J.S. Arora, Introduction to Optimum Design, McGraw-Hill, New York, 1989.
[67] C.A.C. Coello, Use of a self-adaptive penalty approach for engineering optimization problems, Comput. Ind. 41 (2000) 113–127.
[68] 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 (2) (1994) 405–411.
[69] D. Kvalie, Optimization of Plane Elastic Grillages, (Ph.D. thesis), Norges Teknisk Naturvitenskapelige Universitet, Norway, 1967.
[70] 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. 13 (5) (2013) 2592–2612.
[71] Q. He, L. Wang, An effective co-evolutionary particle swarm optimization for constrained engineering design problems, Eng. Appl. Artif. Intell. 20 (2007) 89–99.
[72] F. Huang, L. Wang, Q. He, An effective co-evolutionary differential evolution for constrained optimization, Appl. Math. Comput. 186 (2007) 340–356.
[73] M. Mahdavi, M. Fesanghary, E. Damangir, An improved harmony search algorithm for solving optimization problems, Appl. Math. Comput. 188 (2007) 1567–1579.

[74] A. Carlos, C. Coello, Constraint-handling using an evolutionary multiobjective optimization technique, Civil Eng. Syst. 17 (2000) 319–346.
[75] K. Deb, An efficient constraint handling method for genetic algorithms, Comput. Methods Appl. Mech. Engrg. 186 (2000) 311–338.
[76] K.S. Lee, Z.W. Geem, A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice, Comput. Methods Appl. Mech. Engrg. 194 (2005) 3902–3933.
[77] A. Askarzadeh, A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm, Comput. Struct. 169 (2016) 1–12.
[78] M.-Y. Cheng, D. Prayogo, Symbiotic organisms search: a new metaheuristic optimization algorithm, Comput. Struct. 139 (2014) 98–112.
[79] X.S. Yang, Nature-Inspired Metaheuristic Algorithms, Luniver press, 2010.
[80] N. Acharya, P. Mahat, N. Mithulananthan, An analytical approach for DG allocation in primary distribution network, Int. J. Electr. Power Energy Syst. 28 (2006) 669–678.
[81] T. Gozel, M. Hocaoglu, An analytical method for the sizing and siting of distributed generators in radial systems, Int. J. Electr. Power Syst. Res. 79 (2009) 912–918.
[82] I. Pisica, C. Bulac, M. Eremia, Optimal distributed generation location and sizing using genetic algorithms, in: 15th Int. Conf. On Intelligent System Applications to Power Systems, ISAP '09, Curitiba, 2009, pp. 1–6.
[83] L.Y. Wong, S.R.A. Rahim, M.H. Sulaiman, O. Aliman, Distributed generation installation using particle swarm optimization, in: 4th Int. Conf. On Power Engineering and Optimization, 23–24 June 2010, pp. 159–163.
[84] J. Olamaei, T. Niknam, G. Gharehpetian, Application of particle swarm optimization for distribution feeder reconfiguration considering distribution generators, Appl. Math. Comput. 201 (2008) 575–586.
[85] T. Niknam, S.I. Taheri, J. Aghaei, S. Tabatabaei, M. Nayeripour, A modified honey bee mating optimization algorithm for multiobjective placement of renewable energy resources, Appl. Energy 88 (2011) 4817–4830.
[86] E.S. Ali, S.M. Abd-Elazim, A.Y. Abdelaziz, Lion optimization algorithm for optimal location and sizing of renewable distributed generations, Renew. Energy 101 (2017) 1311–1324.
[87] W. Tan, M.S. Hassan, M. Majid, H. Rahman, Allocation and sizing of DG using cuckoo search algorithm, in: IEEE Int. Conf. Power Energy, 2012, pp. 133–138.
[88] A. Abdelaziz, Y. Hegazy, W. El-Khattam, M. Othman, A multi-objective optimization for sizing and placement of voltage-controlled distributed generation using supervised big bang-big crunch method, Electr. Power Compon. Syst. 43 (1) (2015) 105–117.
[89] H. Sebaa, K.R. Guerrichc, T. Bouktir, Optimal sizing and placement of renewable energy source in large scale power system using ABC technique in presence of UPFC, in: 2014 International Renewable and Sustainable Energy Conference, IRSEC, 17–19 Oct. 2014, pp. 294–299.
[90] N. Mohandas, R. Balamurugan, L. Lakshminarasimman, Optimal location and sizing of real power DG units to improve the voltage stability in the distribution system using ABC algorithm united with chaos, Electr. Power Energy Syst. 66 (2015) 41–52.
[91] S. Devi, M. Geethanjali, Optimal location and sizing determination of distributed generation and DSTATCOM using particle swarm optimization algorithm, Electr. Power Energy Syst. 62 (2014) 562–570.
[92] A. Mohamed Imran, M. Kowsalya, D.P. Kothari, A novel integration technique for optimal network reconfiguration and distributed generation placement in power distribution networks, Electr. Power Energy Syst. 63 (2014) 461–472.
[93] A. Zeinalzadeh, Y. Mohammadi, M.H. Moradi, Optimal multi objective placement and sizing of multiple DGs and shunt capacitor banks simultaneously considering load uncertainty via MOPSO approach, Electr. Power Energy Syst. 6 (2015) 336–349.
[94] S. Kansal, V. Kumar, B. Tyagi, Hybrid approach for optimal placement of multiple DGs of multiple types in distribution networks, Electr. Power Energy Syst. 75 (2016) 226–235.
[95] K. Muthukumar, S. Jayalalitha, Optimal placement and sizing of distributed generators and shunt capacitors for power loss minimization in radial distribution networks using hybrid heuristic search optimization technique, Electr. Power Energy Syst. 78 (2016) 299–319.
[96] K. Muthukumara, S. Jayalalitha, Integrated approach of network reconfiguration with distributed generation and shunt capacitors placement for power loss minimization in radial distribution networks, Appl. Soft Comput. 52 (2017) 1262–1284.
[97] T.T. Nguyen, A.V. Truong, T.A. Phung, A novel method based on adaptive cuckoo search for optimal network reconfiguration and distributed generation allocation in distribution network, Electr. Power Energy Syst. 78 (2016) 801–815.
[98] A. Khodabakhshian, M.H. Andishgar, Simultaneous placement and sizing of DGs and shunt capacitors in distribution systems by using IMDE algorithm, Electr. Power Energy Syst. 82 (2016) 599–607.
[99] M. Kefayat, A. Lashkar Ara, S.A. Nabavi Niaki, A hybrid of ant colony optimization and artificial bee colony algorithm for probabilistic optimal placement and sizing of distributed energy resources, Energy Convers. Manage. 92 (2015) 149–161.
[100] G. Weng, F. Huang, Y. Tang, J. Yan, Y. Nan, H. He, Fault-tolerant location of transient voltage disturbance source for DG integrated smart grid, Electr. Power Syst. Res. 144 (2017) 13–22.
[101] S. Cheng, M.Y. Chen, P.J. Fleming, Improved multi-objective particle swarm optimization with preference strategy for optimal DG integration into the distribution system, Neurocomputing 148 (2015) 23–29.
[102] S. Devi, M. Geethanjali, Application of modified bacterial foraging optimization algorithm for optimal placement and sizing of distributed generation, Expert Syst. Appl. 41 (2014) 2772–2781.
[103] D.Rama. Prabha, T. Jayabarathi, Optimal placement and sizing of multiple distributed generating units in distribution networks by invasive weed optimization algorithm, Ain Shams Eng. J. 7 (2016) 683–694.
[104] S.A. ChithraDevi, L. Lakshminarasimman, R. Balamurugan, Stud krill herd algorithm for multiple DG placement and sizing in a radial distribution system, Eng. Sci. Technol. Int. J. 20 (2017) 748–759.
[105] W. Sheng, K.Y. Liu, Y. Liu, X. Meng, Y. Li, Optimal placement and sizing of distributed generation via an improved nondominated sorting genetic algorithm II, IEEE Trans. Power Deliv. 30 (2) (2015) 569–578.
[106] H.B. Tolabi, M.H. Ali, M. Rizwan, Simultaneous reconfiguration optimal placement of dstatcom, and photovoltaic array in a distribution system based on fuzzy-ACO approach, IEEE Trans. Sustain. Energy 6 (1) (2015) 210–218.
[107] R. Kollu, S.R. Rayapudi, V.L.N. Sadhu, Anovel method for optimal placement of distributed generation in distribution systems using HSDO, Int. Trans. Electr. Energy Syst. 24 (2014) 547–561.
[108] Power system data for IEEE 30-bus test system, 2000, Available at: http://www.ece.ubc.ca/~hameda/download_files/case30m.
[109] R.D. Zimmerman, C.E. Murillo-Sánchez, R.J. Thomas, MATPOWER: Steady-state operations, planning and analysis tools for power systems research and education, IEEE Trans. Power Syst. 26 (1) (2011) 12–19, http://dx.doi.org/10.1109/TPWRS.2010.2051168.
[110] H. Wang, C.E. Murillo-Sánchez, R.D. Zimmerman, R.J. Thomas, On computational issues of market-based optimal power flow, IEEE Trans. Power Syst. 22 (3) (2007) 1185–1193, http://dx.doi.org/10.1109/TPWRS.2007.901301.
[111] National Renewable Energy Laboratory, Distributed generation renewable energy estimate of costs, 2016, [Online] Available: https://www.nrel.gov/analysis/tech-lcoe-re-cost-est.html.
[112] H. Sebaa, K.R. Guerrichc, T. Bouktir, Optimal sizing and placement of renewable energy source in large scale power system using ABC technique in presence of UPFC, in: 2014 International Renewable and Sustainable Energy Conference, IRSEC, 17–19 Oct. 2014, pp. 294–299.
[113] M. Asghari, J. Mohammadi, Fuzzy multi-objective OPF considering voltage security and fuel emission minimization, in: Asia-Pacific Power and Energy Engineering Conference, APPEEC, 27–29 March 2012, p. 6.
[114] S. Ghosh, S.P. Ghoshal, S. Ghosh, Optimal sizing and placement of distributed generation in a network system, Electr. Power Energy Syst. 32 (2010) 849–856.
[115] F. Urganlı, E. Karatepe, Optimal wind turbine sizing to minimize energy loss, Electr. Power Energy Syst. 53 (2013) 656–663.