Appl Intell (2018) 48:2628–2660
https://doi.org/10.1007/s10489-017-1097-7

A novel quasi-oppositional chaotic antlion optimizer for global optimization

Subhodip Saha[1] · V. Mukherjee[1]

Published online: 15 December 2017

Abstract This paper proposes a novel quasi-oppositional chaotic antlion optimizer (ALO) (QOCALO) for solving global optimization problems. ALO is a population based algorithm motivated by the unique hunting behavior of antlions in nature and exhibits strong influence in solving global and engineering optimization problems. In the proposed QOCALO algorithm of the present work, the initial population is generated using the quasi-opposition based learning (QOBL) and the concept of QOBL based generation jumping is utilized inside the main searching strategy of the proposed algorithm. Utilization of QOBL ensures better convergence speed of the proposed algorithm and it also provides better exploration of the search space. Alongside the QOBL, a chaotic local search (CLS) is also incorporated in the proposed QOCALO algorithm. The CLS guides local search around the global best solution that provides better exploitation of the search space. Thus, a better trade-off between exploration and exploitation holds for the proposed algorithm which makes it robust. It is observed that the proposed algorithm offers better results than the original ALO in terms of solution quality and convergence speed. The proposed QOCALO algorithm is implemented and tested, successfully, on nineteen mathematical benchmark test functions of varying complexities and the experimental results are compared to those offered by the basic ALO and some other recently developed nature inspired algorithms. The efficacy of the proposed algorithm is further utilized to solve three real world engineering optimization problems *viz.* (a) the placement and sizing problem of distributed generators in radial distribution networks, (b) the congestion management problem in power transmission system and (c) the optimal design of pressure vessel.

Keywords Antlion optimizer · Chaotic local search · Distributed generation · Global optimization · Metaheuristic · Quasi-opposition based learning

1 Introduction

The term optimization in mathematics and engineering refers to the process of finding the best solution for a particular problem, subject to a set of constraints. Mathematical optimization techniques, which borrowed ideas from fundamental geometry and calculus, were the tools to solve engineering optimization problems till the mid-eighties. These mathematical optimization techniques are deterministic in nature and some of them are gradient based methods which employ derivation of the search space. This increases complexity of the problem and often leads to entrapment into local optima.

With the rapid improvement in computational efficiency in the last few decades, metaheuristic algorithms have been widely utilized as the primary tool for solving real world engineering optimization problems. The metaheuristic algorithms are problem independent and driven by stochastic operators that assist them to avoid local optima. Evolutionary algorithms (EAs) are stochastic, nature inspired metaheuristic algorithms that evolve a randomly-picked solution set from the search space of the problem toward

✉ Subhodip Saha
subhodip_saha@ymail.com

V. Mukherjee
vivek_agamani@yahoo.com

[1] Department of Electrical Engineering, Indian Institute of Technology (Indian School of Mines), Dhanbad, Jharkhand, India

the global best solution in an iterative manner. Some of the widely used nature inspired metaheuristic algorithms include genetic algorithm (GA) (based on Darwin's theory of evolution) [1], particle swarm optimization (PSO) (which mimics the foraging behavior of a swarm of birds or fishes) [2], differential evolution (DE) (based on evolution theory) [3], artificial bee colony (ABC) algorithm (that depicts the foraging behavior of honey bees) [4] and ant colony optimization (ACO) (inspired from the behavior of real ant colonies searching for food) [5]. These algorithms have grown interest among the researchers since the last two decades by offering good quality solutions to science and engineering optimization problems. In contrast to the success of these algorithms, there also exist a handful of new metaheuristics which have been developed in the recent years and this development is driven by the shortcoming of the no free lunch (NFL) theorem [6]. According to the NFL theorem, there is no such optimization technique that can solve all kinds of optimization problems. If any optimizer performs well for a set of problems, then there is no guarantee that it will offer good results for a different set of optimization problems. This inspires researchers to develop and propose novel metaheuristic algorithms for global optimization and also sets up a new line of research. Some of the recently developed nature inspired metaheuristics are firefly algorithm (FA) [7], league championship algorithm [8], gravitational search algorithm (GSA) [9], bat algorithm (BA) [10], cuckoo search (CS) [11], teaching-learning based optimization (TLBO) [12], krill herd algorithm [13], mine blast algorithm [14], symbiotic organisms search (SOS) [15], antlion optimizer (ALO) [16], water wave optimization [17], moth flame optimization (MFO) [18], sperm whale algorithm [19], multi-verse optimization (MVO) [20], whale optimization algorithm (WOA) [21] and crow search algorithm (CSA) [22]. Beside this, a large number of research works have been published in the last few years which improve the effectiveness of an existing metaheuristic algorithm by either integrating an ingenious search strategy into its original framework (see [23–28]) or by hybridizing two or more metaheuristics (refer [29–32]).

Population based metaheuristic algorithms commence the search process in two fundamental phases, namely, exploration and exploitation. In exploration phase, an extensive stochastic search is carried out over the entire search space which improves the diversity of the solution. The exploitation phase is followed by the exploration phase and it aims to improve the quality of the solution by continuing local search around the promising regions of the search space, already obtained in the exploration phase. It is very important to keep a good balance between exploration and exploitation while designing an algorithm because any improper handling of these phases may produce sub-optimal solution leading to stagnation into the local optima.

Recently, a nature inspired metaheuristic algorithm (i.e., ALO [16]) has been proposed by Mirjalili which is based on the hunting behavior of antlion larvae. The ALO algorithm has offered better results in terms of solution accuracy and convergence mobility as compared to some other popular methods like PSO, GA, FA, BA and CS [16] in solving global optimization problems. Also, the ALO algorithm has successfully been utilized to solve some engineering optimization problems (such as, design of ship propeller [16], wind based hydrothermal scheduling [33, 34], automatic generation control [35], design of linear discrete filters [36] and so on).

Motivated by the NFL theorem [6] and the previous researches for enhancing the performance of an existing metaheuristic, a new modified ALO algorithm has been proposed in the present work which is designated as quasi-oppositional chaotic ALO (QOCALO). The proposed QOCALO algorithm is having a quasi-opposition based learning (QOBL) strategy and a chaotic local search (CLS) inside the framework of the original ALO. The concept of QOBL helps to explore new areas of the search space and provides better exploration. On the other hand, CLS directs the search process around the most promising areas of the search space which provides better exploitation. Thus, a better balance between exploration and exploitation holds in case of the proposed QOCALO algorithm which makes this newly developed algorithm more robust as compared to its original counterpart. The performance of the proposed algorithm is validated by employing it for solving nineteen mathematical benchmark functions which include seven unimodal, six multimodal and six composite benchmark functions. The effectiveness and the superiority of the proposed algorithm have been established by comparing the performance of the proposed QOCALO with the basic ALO algorithm and some other recently developed nature inspired metaheuristics, such as MFO, MVO, WOA and CSA.

The main contribution of this paper lies in the framing of a new and robust variant of ALO algorithm which is capable to solve global optimization problems more effectively than the original ALO in terms of solution accuracy and convergence mobility. The proposed QOCALO algorithm has been utilized to solve three real world engineering optimization problems, such as, (a) the placement and sizing problem of distributed generators (DGs) in radial distribution networks, (b) the congestion management (CM) problem in power transmission system and (c) the optimal design problem of pressure vessel.

The rest of this paper is organized as follows: In Section 2, ALO algorithm is briefly described. Section 3 illustrates the proposed QOCALO algorithm. The simulation results of benchmark test functions are presented and discussed in Section 4. In Section 5, the placement and sizing problem of DGs has been solved by the proposed QOCALO algorithm. The CM problem has been solved using the proposed algorithm in Section 6. In Section 7, the proposed QOCALO algorithm is utilized to determine the optimal design of pressure vessel. Finally, the conclusions of the present work are drawn in Section 8 along with focusing on some future research directions.

2 The ALO algorithm

The ALO algorithm [16] is proposed by Mirjalili in 2015. The ALO is a population based stochastic search algorithm which mimics the predatory behavior of antlions in the nature. The antlions (or doodlebugs) belong to the *Myrmeleontidae* family of insects and *Neuroptera* order. The average lifespan of antlions may take up to three years, out of which only 3-5 weeks are spent in adulthood and the rest of the lifetime is spent as larvae. The antlions carry out a unique process of hunting in their larvae stage and preferably hunt for ants, which leads to their unique name. The antlion larva digs conical pits in the sand with the help of its massive jaw and waits at the bottom of the pit for the prey to be trapped in the sand pits. The edges of the sand pits are sharp enough to make insects (rather ants) fall down at the bottom of the pit. Once a prey falls inside the trap, the antlion throws sands out of the pit toward the prey such that the prey fails to escape from the pit and slides at the bottom of the pit. The prey is then consumed by the antlion and the leftovers are thrown away from the pit. In this way, the antlion prepares the pit for its next hunt [16]. In case of the ALO algorithm, the ants behave as the search agents, which move over the entire search space, whereas the antlions hunt them and become fitter.

During the course of optimization of ALO algorithm, the positions of ants and antlions over the search space are stored in the following two matrices, namely, P_a and P_{al} (see (1) and (2))

$$P_a = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,d} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,d} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,d} \end{bmatrix} \tag{1}$$

$$P_{al} = \begin{bmatrix} al_{1,1} & al_{1,2} & \cdots & al_{1,d} \\ al_{2,1} & al_{2,2} & \cdots & al_{2,d} \\ \vdots & \vdots & \vdots & \vdots \\ al_{n,1} & al_{n,2} & \cdots & al_{n,d} \end{bmatrix} \tag{2}$$

where P_a stores the positions of ants over the search space, d is the number of decision variables and P_{al} is the matrix consisting of the positions of antlions hiding in the search space. The number of ants and antlions in the search space are equal and are represented by n. The position matrices (i.e. P_a and P_{al}) are utilized to evaluate the objective function and the fitness values are stored in the following two vectors, namely, F_a and F_{al} (refer (3) and (4))

$$F_a = \begin{bmatrix} f\left(\left[a_{1,1}, a_{1,2}, \cdots, a_{1,d}\right]\right) \\ f\left(\left[a_{2,1}, a_{2,2}, \cdots, a_{2,d}\right]\right) \\ \vdots \\ f\left(\left[a_{n,1}, a_{n,2}, \cdots, a_{n,d}\right]\right) \end{bmatrix} \tag{3}$$

$$F_{al} = \begin{bmatrix} f\left(\left[al_{1,1}, al_{1,2}, \cdots, al_{1,d}\right]\right) \\ f\left(\left[al_{2,1}, al_{2,2}, \cdots, al_{2,d}\right]\right) \\ \vdots \\ f\left(\left[al_{n,1}, al_{n,2}, \cdots, al_{n,d}\right]\right) \end{bmatrix} \tag{4}$$

where F_a stores the fitness value of each ant while F_{al} stores the fitness value of each antlion and f denotes the objective function.

The optimization process of the ALO algorithm is governed by six operating steps which are, basically, the adaptation of the real hunting deeds of antlions found in nature. The step-by-step operations of the ALO algorithm are discussed in the following six sub-sections.

2.1 Creation of random walks for ants

In the ALO algorithm, it is considered that the ants move freely over the entire search space in search for food until they get trapped by antlions. As the movement of the ants is stochastic in nature, it is modeled based on random walk and may be represented by (5) [16]

$$\begin{aligned} X(t) = [0, &\, cumsum(2r(t_1) - 1), cumsum(2r(t_2) - 1), \cdots, \\ &\, cumsum(2r(t_{\max}) - 1)] \end{aligned} \tag{5}$$

where t is the step of random walk (or current iteration), $t_{\max}$ is the maximum iteration, *cumsum* computes the cumulative sum and $r(t)$is a function defined in (6).

$$r\left(t\right) = \begin{cases} 1 \; if \; rand > 0.5 \\ 0 \; otherwise \end{cases} \tag{6}$$

Here, *rand* is a uniformly distributed random number in the interval [0,1]. At each step of the optimization process, the positions of the ants are updated with the help of this random walk. In order to maintain the random walks inside the search boundaries, (5) is normalized to (7)

$$X_i^t = \frac{\left(X_i^t - \alpha_i\right) \times \left(\delta_i^t - \gamma_i^t\right)}{\left(\beta_i - \alpha_i\right)} + \gamma_i^t \tag{7}$$

where X_i^t is the ant position in the i-th dimension of a d-dimensional space at the t-th iteration, α_i and β_i are the minimum and the maximum of random walks, respectively, for the i-th variable, δ_i^t and γ_i^t are the upper and the lower bounds of the i-th variable at the t-th iteration, respectively.

2.2 Building of traps using roulette wheel

It is assumed that each ant is to be trapped by only one selected antlion. Also, the fitter the antlion is, the higher the chance of hunting ants as the fitter antlion makes a better trap. Hence, the selection of the antlion is done using a roulette wheel operator which is based on the fitness of the antlions during the optimization process [16].

2.3 Entrapment of ants in antlion's pit

As per the previous discussion, the random walks of the ants get affected by the antlion traps. To model this mathematically, the boundary of ant movement is adjusted in each iteration, such that the ant moves in a hyperspace around the selected antlion trap. The upper and lower bounds of the ant dimension are computed in each iteration following (8) and (9) [16]

$$\delta^t = antlion_j^t + \delta^t \tag{8}$$

$$\gamma^t = antlion_j^t + \gamma^t \tag{9}$$

where δ^t and γ^t are the upper and the lower boundaries of all ant dimensions at current iteration t and $antlion_j^t$ is the position of the j-th selected antlion at the current iteration t. Thus, the ant movement is restricted in a d-dimensional hyperspace with a boundary of (γ^t, δ^t), using (8) and (9), in order to entrap the ants in the antlion pits.

2.4 Sliding of ants toward the antlion

When an antlion realizes that an ant has fallen into its pit, it throws sands toward the ant, such that the ant never escapes and slides down toward the antlion waiting at the bottom of the pit. Hence, the radius of the random walk of ants should be adaptively decreased to mathematically model this behavior of antlions and it is governed by the following three equations [16]:

$$\gamma^t = \frac{\gamma^t}{I} \tag{10}$$

$$\delta^t = \frac{\delta^t}{I} \tag{11}$$

$$I = 10^{\omega} \cdot \frac{t}{t_{\max}} \tag{12}$$

where ω is a constant that takes up a value between 1 and 6 depending on (13).

$$\omega = \begin{cases} 2 \; if \; t > 0.1 \times t_{\max} \\ 3 \; if \; t > 0.5 \times t_{\max} \\ 4 \; if \; t > 0.75 \times t_{\max} \\ 5 \; if \; t > 0.9 \times t_{\max} \\ 6 \; if \; t > 0.95 \times t_{\max} \\ 1 \; otherwise \end{cases} \tag{13}$$

From (10) – (13), it is clear that with an increase in the current iteration, the value of I is increased and, hence, the radius of the random walk of ants is reduced. This ensures shrinking of search space gradually and offers better exploitation.

2.5 Catching of ants and rebuilding the pit

The final stage of hunt occurs when the ant reaches at the bottom of the antlion's pit. At this stage, the antlion catches the ant and drags it inside the sand to devour it. For the sake of mathematical modeling of this stage, it is assumed that the ingestion of ant by the antlion takes place when the ant becomes fitter than its corresponding antlion. The antlion then updates its position to the latest position of the hunted ant, i.e. improves its fitness and, hence, the chance of catching a new prey is increased. This process of catching ants is guided by (14) for the minimization of the objective function [16]

$$antlion_j^t = ant_i^t \;\; if \;\; f\left(ant_i^t\right) < f\left(antlion_j^t\right) \tag{14}$$

where ant_i^t is the position of the i-th ant at the tth iteration and $f\left(ant_i^t\right)$ is its corresponding fitness value. This is analogous to the rebuilding of the pit to enhance the chance of hunting.

2.6 Application of elitism

The ALO algorithm applies elitism to its search strategy by preserving the best solution (i.e. the fittest antlion) obtained at each generation. This fittest antlion (or elite antlion) is assumed to influence the movement of each ant. Hence, it is considered that each ant is about to take a random walk around the antlion selected by the roulette wheel and the elite antlion simultaneously and this may be represented by (15) [16]

$$ant_i^t = \frac{RW_{antlion}^t + RW_{elite}^t}{2} \tag{15}$$

where $RW_{antlion}^t$ represents the random walk around the antlion selected by roulette wheel at the t-th iteration and RW_{elite}^t represents the random walk around the elite antlion at the t-th iteration. The pseudo-code of the ALO algorithm is provided in Algorithm 1 [16].

Algorithm 1 Pseudo-code of ALO

```
Define objective function $f(X)$; $X=(x_1,x_2,...,x_d)$          % $d$ is dimension of the problem
Randomly generate the initial population of ants and antlions
Calculate the fitness of initial antlions and sort them
Find the best antlion and consider it as the elite
while $(t < iter_{\max})$                                      % $iter_{\max}$ is the maximum number of iterations
    for $i$ =1: $n$                                            % $n$ is the number of ants
        Select an antlion using roulette wheel
        Update $\delta^t$ and $\gamma^t$ using (8) and (9), respectively
        Update the radius of random walk of ant using (10) and (11)
        Create a random walk using (5) and normalize it by using (7)
        Update the ant position using (15)
    end for
    Calculate the fitness of all ants
    If an ant becomes fitter than its corresponding antlion, then it is replaced using (14)
    Update the position of the elite if any antlion becomes fitter than it
end while
Return the elite
```

3 The proposed QOCALO algorithm

The framework of the proposed QOCALO algorithm of the present work is formulated by simultaneous integration of QOBL and CLS strategies into the basic ALO algorithm. The different components of the proposed algorithm are discussed in the following sub-sections.

3.1 QOBL: a concept

Conventional metaheuristic algorithms begin the search process using a set of randomly generated initial solutions and progress toward the global best one. Hence, the rate of convergence depends on the distance between the initial solution set or the initial population and the global optimum solution. If the randomly generated solution is too far away from the global optimum one, then the algorithm takes considerably longer time to converge and, hence, the convergence rate becomes very poor. To avoid this problem and to improve the convergence rate, Tizhoosh [37] has suggested for consideration of both the randomly generated solutions and their opposite solutions simultaneously. According to [38], the random guess is far away from the global optimum than its opposite guess for 50% cases. Hence, starting the search process with an initial solution set that consists of the best of the two sets (i.e. the randomly generated solution set and its opposite solution set) helps to improve the convergence mobility. The attributes of QOBL are defined as follows.

3.1.1 Opposite point

Let x be a real number in one-dimensional search space with search interval $[a, b]$. Then, the corresponding opposite number (ox) is defined by (16)

$$ox = a + b - x \tag{16}$$

In case of a d-dimensional search space, let $X = (x_1, x_2, ..., x_d)$ be a candidate solution, where $(x_1, x_2, ..., x_d) \in R$ and $x_i \in [a_i, b_i]\, \forall i \in \{1, 2, ..., d\}$. The opposite point of X is considered as $OX = (ox_1, ox_2, ..., ox_d)$ and is defined by (17)

$$ox_i = a_i + b_i - x_i \tag{17}$$

3.1.2 Quasi-opposite point

For one-dimensional search space, the quasi-opposite number (qox) is defined as a random number between the center of the search space $\left(\frac{a+b}{2}\right)$ and the opposite number (ox), as in (18)

$$qox = rand\left(\left(\frac{a+b}{2}\right), ox\right) \tag{18}$$

Again, for d-dimensional search space, the quasi-opposite point of X is considered as $QOX = (qox_1, qox_2, ..., qox_d)$ and is, mathematically, defined by (19).

$$qox_i = rand\left(\left(\frac{a_i + b_i}{2}\right), ox_i\right) \tag{19}$$

The quasi-opposite point has a higher chance to be closer to the global optimum as compared to the opposite point [39] and, hence, QOBL is more capable to improve the convergence rate. Recently, QOBL has been applied in different metaheuristic algorithms to improve its performance and some of them are quasi-oppositional DE [39], quasi-oppositional harmony search algorithm (HSA) [40], quasi-oppositional TLBO [41] and quasi-oppositional group search algorithm [42].

3.1.3 QOBL based population initialization

The initial population of the conventional metaheuristic algorithms is generated randomly without having any prior knowledge of the solution space. In this case, the QOBL based population initialization may achieve fitter candidate solutions as the simultaneous consideration of the randomly generated initial positions and their quasi-opposite positions improves the quality of the initial population and accelerates the search process by exploring the powerful regions of the search space. The pseudo-code of QOBL based population initialization is presented in Algorithm 2.

Algorithm 2 Pseudo-code of QOBL based population initialization

```
Generate initial random population $X$
for $i = 1: n$                                                % $n$ is the population size
    for $j = 1: d$                                            % $d$ is the dimension of the problem
        $OX_{i,j} = a_j + b_j - X_{i,j}$                      % $OX$ is the opposite of the initial population $X$
        $M_{i,j} = (a_j + b_j)/2$                             % $[a,b]$ is the search boundary
        if $(X_{i,j} < M_{i,j})$
            $QOX_{i,j} = M_{i,j} + (OX_{i,j} - M_{i,j}) \times rand$     % $QOX$ is the quasi-opposite of $X$
        else $QOX_{i,j} = OX_{i,j} + (M_{i,j} - OX_{i,j}) \times rand$
        end if else
    end for
end for
Select $n$ number of fittest candidate solutions as initial population from the set $\{X, QOX\}$
```

3.1.4 QOBL based generation jumping

The evolutionary process of the algorithm may be forced to jump to a new candidate solution by applying the concept of QOBL based generation jumping. This quasi-opposite population jumping has higher probability to produce fitter candidate solution than the existing one [39]. The QOBL based generation jumping is associated with a parameter (namely, jumping rate (j_r)) which decides whether to remain at the position of the candidate solution generated by the search process of the algorithm or jump over to the quasi-opposite position of the candidate solution. The process of QOBL based generation jumping is depicted in Algorithm 3.

Algorithm 3 Pseudo-code of QOBL based generation jumping

```
Set the value of $j_r$                                  % $j_r$ is the jumping rate
if $rand < j_r$
  for $i = 1: n$                                        % $n$ is the population size
    for $j = 1:d$                                       % $d$ is the dimension of the problem
      $OX_{i,j} = a_j + b_j - X_{i,j}$                  % $OX$ is the opposite of the ant population $X$
      $M_{i,j} = (a_j + b_j)/2$                         % $[a, b]$ is the search boundary
      if $(X_{i,j} < M_{i,j})$
        $QOX_{i,j} = M_{i,j} + (OX_{i,j} - M_{i,j}) \times rand$   % $QOX$ is the quasi-opposite of $X$
      else $QOX_{i,j} = OX_{i,j} + (M_{i,j} - OX_{i,j}) \times rand$
      end if else
    end for
  end for
end if
```

3.2 CLS based optimization

To improve the performance of ALO in terms of solution quality, a CLS based search strategy has been incorporated in the present work. The CLS also helps to prevent the optimizer to be trapped into local optima. In recent years, a wide variety of CLS based metaheuristic algorithms has been proposed and some of them are CLS based PSO [43–48], chaotic binary PSO [49, 50], CLS based DE [51–53], chaotic differential bee colony optimization [54], chaotic bee colony algorithm [55, 56], chaotic HSA [57], chaotic TLBO [58] and CLS based SOS [59, 60]. In contrast to the performance of these algorithms, the effectiveness of the CLS strategy to improve the search performance of the metaheuristic algorithms may be justified.

The CLS paradigm of the present work is applied to the global best solution obtained from the traditional ALO algorithm. This is done so as there is higher probability to achieve better solution in the vicinity of the global optimum solution [44]. The CLS intensifies the search process toward a more promising region and, thus, improves exploitation. The CLS is stopped when a better solution is found or the local search limit is reached.

3.2.1 Chaotic map

Chaos is a non-linear, dynamic and deterministic system which is very sensitive to its initial condition [61]. Due to some special characteristic features (like ergodicity and non-repetition), the chaotic system has the potential to carry out search process at a higher speed than the normal stochastic search which is probabilistic in nature [50]. There is a wide variety of chaotic maps available in the literature (see [62, 63]). In the present work, the well-known logistic map is utilized to carry out the CLS. The logistic map is, mathematically, formulated as in (20) [51]

$$Ch_{k+1} = \mu \times Ch_k \times (1 - Ch_k)\ , \quad Ch \in (0, 1)\ , \quad Ch \neq 0.25,\ 0.5\ , 0.75 \quad (20)$$

where Ch_k is the chaotic variable at the k-th generation. The initial value of the chaotic sequence (i.e. Ch_0) is a random number generated by using *rand* function. The behavior of the logistic map is controlled by the parameter μ and it is depicted in Fig. 1. At $\mu = 4$, the logistic function shows thorough chaotic behavior and, hence, the value of μ is set to 4. The chaotic sequence generated by the logistic map over 100 numbers of iterations is shown in Fig. 2.

3.2.2 The proposed CLS strategy

The CLS strategy is, generally, employed to refine the quality of the previous best solution. In the present work, a new candidate solution is generated in the neighborhood of the previous best solution following (21)

$$p_{i,j} = p_{g,j} + Ch \times \left(p_{k1,j} - p_{k2,j}\right) \tag{21}$$

where $p_{i,j}$ is the j-th dimension of the i-th solution generated in the vicinity of the previous best solution, $p_{g,j}$ is the j-th dimension of the previous best solution, Ch is the chaotic number generated by the chaotic sequence expressed in (20), $k1$ and $k2$ are two mutually exclusive integers randomly chosen from $\{1, 2, \ldots, n\}$ (n being the population size). Thus, the CLS strategy, guided by (21), well maintains the population diversity as it produces enough random solutions. The detailed process of the proposed CLS strategy is presented in Algorithm 4.

Algorithm 4 Pseudo-code of the proposed CLS strategy

```
Set CLS limit $K$
Initialize the chaotic sequence by generating $Ch_0$
for $i = 1: n$                                          % $n$ is the population size
    for $j = 1: K$
        Update the chaotic sequence $Ch$ using (20)
        Pick $k1$ and $k2$ randomly from $\{1,2,...,n\}$, such that $k1 \neq k2, k1 \neq i$ and $k2 \neq i$
        Generate a new solution $ant_{i,j} = elite_j + Ch \times (x_{k1,j} - x_{k2,j})$ according to (21)
        if $f(ant_i) < f(antlion_i)$
            $x_i = ant_i$
            $f(antlion_i) = f(ant_i)$
            break
        end if
    end for
    if $f(antlion_i) < f(elite)$
        $elite = x_i$
        $f(elite) = f(antlion_i)$
    end if
end for
```

3.3 The proposed algorithm

The proposed algorithm is constructed by integrating both the QOBL and the CLS based search strategies into the basic structure of the original ALO algorithm. The search process of the proposed QOCALO algorithm is presented in detail in Algorithm 5.

Algorithm 5 Pseudo-code of the proposed QOCALO algorithm

```
Define objective function $f(X)$; $X = (x_1, x_2, ..., x_d)$          % $d$ is dimension of the problem
Set the values of $j_r$ and $K$
Generate the initial population of ants and antlions using Algorithm 2
Calculate the fitness of initial antlions and sort them
Find the best antlion and consider it as the elite
while $(t < iter_{\max})$                                 % $iter_{\max}$ is the maximum number of iterations
    for $i$ =1: $n$                                                 % $n$ is the number of ants
        Select an antlion using roulette wheel
        Update $\delta^t$ and $\gamma^t$ using (8) and (9), respectively
        Update the radius of random walk of ant using (10) and (11)
        Create a random walk using (5) and normalize it by using (7)
        Update the ant position using (15)
    end for
    Calculate the fitness of all ants
    if $rand < j_r$
        for $i$ =1: $n$
            Determine the quasi-opposite ant positions using Algorithm 3
            Calculate the quasi-opposite ant fitness
            if $f(quasi\_opposite\_ant_i) < f(ant_i)$
                $ant_i = quasi\_opposite\_ant_i$
                $f(ant_i) = f(quasi\_opposite\_ant_i)$
            end if
        end for
        Update antlion positions and fitness based on the ants using (14)
        Update the elite
    else perform the CLS according to Algorithm 4
    end if else
end while
Return the elite
```

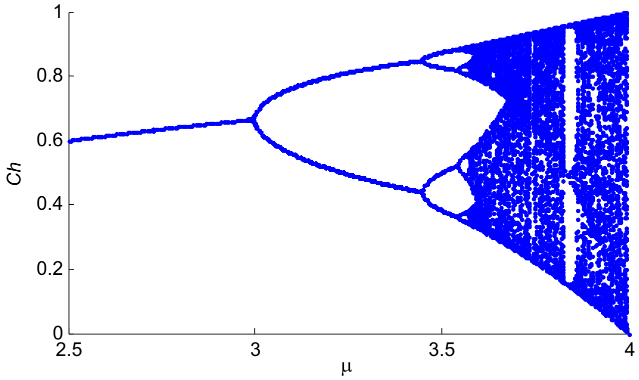


Fig. 1 Bifurcation diagram of the logistic map

4 Experimental results pertaining to function optimization problem

4.1 Details of test functions and experimental setup

The optimization capability of the proposed QOCALO algorithm is investigated on nineteen benchmark test functions. These test functions may be grouped into three categories, *viz.* unimodal, multimodal and composite functions. Unimodal functions have a single optimum and, hence, they are useful for examining the exploitation and the convergence behavior of the algorithm [16]. On the other hand, multimodal functions have multiple optima, one of which is global optimum and the rest are local optima. The optimization algorithm should converge to the global optimum avoiding all the local optima. Hence, the multimodal functions are benchmarked to examine the exploration of the algorithm. Finally, the composite functions are the rotated, shifted, combined and biased versions of other unimodal and multimodal test functions. The composite functions have a large number of local optima and the shape of search space is different in different regions of the search space which resembles the real search space for practical problems [16]. Therefore, the composite functions are benchmarked to check whether an algorithm is capable of maintaining a proper balance between exploration and exploitation. The details of the unimodal, the multimodal and the composite test functions are listed in Tables 1, 2 and 3 [16].

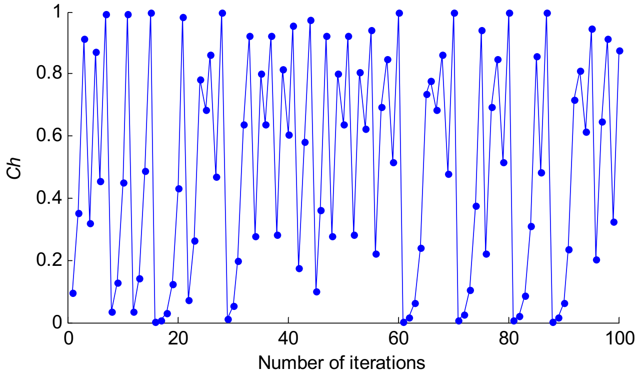


Fig. 2 Distribution of logistic map over 100 numbers of iterations

The simulations are carried out in MATLAB 2013a computing environment on a 2.3 GHz core i5 personal computer with 3.86 GB RAM. Due to the stochastic nature of the evolutionary optimization techniques, each test function is tested for 100 independent trial runs to minimize the statistical error and the best, the worst, the mean and the standard deviation of the results are obtained over the trial runs. The proposed algorithm is tested on 10, 30 and 200-dimensional versions of the unimodal and multimodal test functions to establish the scalability of the proposed algorithm.

4.2 Non-parametric statistical test set up

The metrics, such as the mean and the standard deviation, compare the overall performance of the algorithms while statistical tests verify the superiority of an algorithm over the others by considering the results of each trial run. The statistical tests are necessary as the metaheuristic algorithms are stochastic in nature. In the present work, a non-parametric pair-wise Wilcoxon rank-sum test is performed to verify whether the simulation results of the proposed QOCALO are significantly superior to the simulation results of other reported algorithms. The null hypothesis of this test states that the median of the difference of the two samples is zero (i.e. the solution sets of the algorithms are not statistically different). The test is implemented at a significance level of 5% and a p-value less than 0.05 points toward a significant difference between the two samples (i.e. the null hypothesis is rejected).

4.3 Parameter setting

In the present work, the optimization capability of the proposed QOCALO algorithm is tested by comparing its performance with the other recently developed nature inspired algorithms such as MFO, MVO, WOA, CSA and the original ALO. Each trial run employs 30 antlions to carry out the search process over 1000 iterations for 10 and 30-dimensional test problems. The number of antlions and iterations are considered as 100 and 5000, respectively, for 200-dimensional test problems. These settings are considered in line with [16] to maintain a fair comparison between the proposed QOCALO and the basic ALO. The parameters of the other reported algorithms are considered as of their original published works (see [18, 20–22]).

Table 1 Detailed description of unimodal benchmark functions

Function	Dimension	Range	GM
$F_1(x) = \sum_{i=1}^{d} x_i^2$	10, 30, 200	[−100, 100]	0
$F_2(x) = \sum_{i=1}^{d} \lvert x_i \rvert + \prod_{i=1}^{d} \lvert x_i \rvert$	10, 30, 200	[−10, 10]	0
$F_3(x) = \sum_{i=1}^{d} \left(\sum_{j=1}^{i} x_j \right)^2$	10, 30, 200	[−100, 100]	0
$F_4(x) = \max \{ \lvert x_i \rvert \} \quad 1 \le i \le d$	10, 30, 200	[−100, 100]	0
$F_5(x) = \sum_{i=1}^{d-1} \left(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right)$	10, 30, 200	[−30, 30]	0
$F_6(x) = \sum_{i=1}^{d} (x_i + 0.5)^2$	10, 30, 200	[−100, 100]	0
$F_7(x) = \sum_{i=1}^{d} i x_i^4 + random\,(0, 1)$	10, 30, 200	[−1.28, 1.28]	0

*d denotes the number of variables, GM is the global minima

The parameter j_r of the proposed algorithm controls the convergence rate and, hence, it is to be carefully selected. A higher value of j_r may reduce the population diversity at a very faster rate and leads to premature convergence. An experiment has been conducted to set the value of j_r. Figure 3 shows the variation in the number of iterations required for convergence with varying j_r for the function F4. It may be observed from Fig. 3 that, a j_r value beyond 0.6 results in premature convergence. Hence, the value of j_r is set to 0.6.

Another parameter of the proposed algorithm is the CLS limit (K), which is also set after conducting an experiment on function F5 with 30 dimensions. In this experiment, all the parameters of the proposed QOCALO algorithm are considered to be the same as the previous experiment while j_r is considered as 0.6 and K is varied as in Fig. 4. From Fig. 4, it is observed that the proposed algorithm yields the minimum fitness value at $K = 10$. Therefore, the K value is set to 10.

4.4 Simulation results for unimodal functions

The performance of the proposed QOCALO in optimizing unimodal functions is compared to the performances of the other reported algorithms and the detailed simulation results are listed in Tables 4, 5 and 6. The algorithms are ranked on the basis of their performance to optimize the set of seven unimodal functions. The algorithm that produces the minimum mean fitness value is considered to have rank 1 while in case of a tie equal ranks are given to the respective algorithms. It may be observed from Tables 4–6 that the proposed algorithm stands second to only MFO for function F6 with 10 dimensions while it shares the first rank with WOA for the 200 dimensional functions (F1 and F2) but for 30-dimensional case the proposed algorithm stands first for all the test functions. Hence, the proposed QOCALO dominates other algorithms in optimizing unimodal functions as it attains an overall rank 1 for all the three different dimensions. The p-values obtained from the pair-wise Wilcoxon rank-sum test for the unimodal functions with three different dimensions are listed in Table 7. It may be noted that the p-values are much less than 0.05 for almost all the instances which statistically establishes the superiority of the proposed QOCALO algorithm over the others. The '+' sign implies that the proposed QOCALO algorithm is significantly better than the other algorithm, '-' sign indicates that the other algorithm is significantly better than the proposed QOCALO and '=' sign signifies that there is no significant difference between the proposed QOCALO and the other algorithm. From Table 7 it may be observed that the proposed algorithm straightly outperformes MFO, MVO and CSA for all the unimodal functions for all the three different dimensions while it shows similar performance with ALO for the function F6 and with WOA for the functions F1, F2 and F7 with 200 dimensions. The comparative convergence profiles of the proposed QOCALO and the other compared algorithms are depicted in Figs. 5, 6 and 7. From the observation tables and convergence profiles it may be concluded that the proposed algorithm shows better performance in optimizing unimodal test functions which points to better exploitation and faster convergence behavior of the proposed algorithm as compared to other reported algorithms.

Table 2 Detailed description of multimodal benchmark functions

Function	Dimension	Range	GM
$F_8(x)=\sum_{i=1}^{d}-x_i\sin\left(\sqrt{\lvert x_i\rvert}\right)$	10, 30, 200	[−500, 500]	$-418.9829\times_d^*$
$F_9(x)=\sum_{i=1}^{d}\left(x_i^2-10\cos(2\pi x_i)+10\right)$	10, 30, 200	[−5.12, 5.12]	0
$F_{10}(x)=-20\exp\left(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d}x_i^2}\right)-\exp\left(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)\right)+20+e$	10, 30, 200	[−32, 32]	0
$F_{11}(x)=\frac{1}{4000}\sum_{i=1}^{d}x_i^2-\prod_{i=1}^{d}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$	10, 30, 200	[−600, 600]	0
$F_{12}(x)=\frac{\pi}{d}\left\{10\sin(\pi y_1)+\sum_{i=1}^{d-1}(y_i-1)^2\left[1+10\sin^2(\pi y_{i+1})\right]+(y_d-1)^2\right\}+\sum_{i=1}^{d}u(x_i,10,100,4)$			
$y_i=1+\frac{x_i+1}{4}$			
$u(x_i,a,k,m)=\begin{cases}k(x_i-a)^m & x_i>a\\ 0 & -a<x_i<a\\ k(-x_i-a)^m & x_i<-a\end{cases}$	10, 30, 200	[−50, 50]	0
$F_{13}(x)=0.1\left\{\sin^2(3\pi x_1)+\sum_{i=1}^{d}(x_i-1)^2\left[1+\sin^2(3\pi x_i+1)\right]+(x_d-1)^2\left[1+\sin^2(2\pi x_d)\right]\right\}+\sum_{i=1}^{d}u(x_i,5,100,4)$	10, 30, 200	[−50, 50]	0

*d denotes the number of variables, GM is the global minima

4.5 Simulation results for multimodal functions

The comparative simulation results of the proposed QOCALO and the other algorithms for multimodal test functions are provided in Tables 8, 9 and 10. From these tables it may be observed that the proposed algorithm gets an overall rank 1 for 10 and 30 dimensional test functions while it stands second to WOA for 200 dimensional functions. The p-values of the pair-wise Wilcoxon rank-sum test for the multimodal test functions are reported in Table 11. From Table 11, it may be noted that the proposed QOCALO produces significantly better results than MFO and MVO while it completely outperforms ALO and CSA by beating them in 17 instances. The only competitor to the proposed algorithm in this section is the WOA as it produces better results than the proposed algorithm for three instances and similar results for seven instances. Hence, it may be stated that the proposed algorithm has better capabilities to optimize lower and medium dimensional multimodal functions as compared to other reported algorithms but the performance deteriorates slightly when it comes to optimize very high dimensional problems. Considering the properties of the multimodal test problems and the performance of the proposed algorithm, it may be inferred that the proposed QOCALO algorithm exhibits a better exploration over the search space. The comparative convergence profiles for some selected test functions are depicted in Figs. 5–7, which indicate faster convergence mobility of the proposed algorithm.

4.6 Simulation results for composite functions

The simulation results for the composite test problems are tabulated in Table 12. These results are quite far from the global minima as compared to the previous cases and this is due to the complexity involved in the composite test functions. However, the proposed QOCALO algorithm yields better results in optimizing composite test problems compared to other reported algorithms. The proposed algorithm attains an overall rank 1 in solving composite test problems compared to its original counterpart (i.e. ALO) which stands second. The p-values obtained from the pair-wise Wilcoxon rank-sum test are reported in Table 13 which also shows the superiority of the proposed algorithm over other reported algorithms. From Table 13 it may be observed that the proposed algorithm produces significantly better results than MFO, WOA and CSA for all the six composite test functions. The proposed algorithm produces significantly better results than the original ALO for four functions while it produces statistically similar results for the functions *CF2* and *CF4*. These results indicate that the proposed QOCALO algorithm is capable of maintaining a better balance between exploration and exploitation compared to

Table 3 Detailed description of composite benchmark functions

Function	Dimension	Range	GM
$CF1$	10	[−5, 5]	0
$f_1, f_2, f_3, ..., f_{10} = F_1$			
$[\sigma_1, \sigma_2, \sigma_3, ..., \sigma_{10}] = [1,\ 1,\ 1, ...,\ 1]$			
$[\lambda_1, \lambda_2, \lambda_3, ..., \lambda_{10}] = \left[5/100,\ 5/100,\ 5/100, ..., 5/100\right]$			
$CF2$	10	[−5, 5]	0
$f_1, f_2, f_3, ..., f_{10} = F_{11}$			
$[\sigma_1, \sigma_2, \sigma_3, ..., \sigma_{10}] = [1,\ 1,\ 1, ...,\ 1]$			
$[\lambda_1, \lambda_2, \lambda_3, ..., \lambda_{10}] = \left[5/100,\ 5/100,\ 5/100, ..., 5/100\right]$			
$CF3$	10	[−5, 5]	0
$f_1, f_2, f_3, ..., f_{10} = F_{11}$			
$[\sigma_1, \sigma_2, \sigma_3, ..., \sigma_{10}] = [1,\ 1,\ 1, ...,\ 1]$			
$[\lambda_1, \lambda_2, \lambda_3, ..., \lambda_{10}] = [1,\ 1,\ 1, ...,\ 1]$			
$CF4$	10	[−5, 5]	0
$f_1, f_2 = F_{10}$			
$f_3, f_4 = F_9$			
$f_5, f_6 = Weierstrass\ Function^*$			
$f_7, f_8 = F_{11}$			
$f_9, f_{10} = F_1$			
$[\sigma_1, \sigma_2, \sigma_3, ..., \sigma_{10}] = [1,\ 1,\ 1, ...,\ 1]$			
$[\lambda_1, \lambda_2, \lambda_3, ..., \lambda_{10}] = \left[5/32,\ 5/32,\ 1,\ 1,\ 5/0.5,\ 5/0.5,\ 5/100,\ 5/100,\ 5/100,\ 5/100\right]$			
$CF5$	10	[−5, 5]	0
$f_1, f_2 = F_9$			
$f_3, f_4 = Weierstrass\ Function^*$			
$f_5, f_6 = F_{11}$			
$f_7, f_8 = F_{10}$			
$f_9, f_{10} = F_1$			
$[\sigma_1, \sigma_2, \sigma_3, ..., \sigma_{10}] = [1,\ 1,\ 1, ...,\ 1]$			
$[\lambda_1, \lambda_2, \lambda_3, ..., \lambda_{10}] = \left[1/5,\ 1/5,\ 5/0.5,\ 5/0.5,\ 5/100,\ 5/100,\ 5/32,\ 5/32,\ 5/100,\ 5/100\right]$			
$CF6$	10	[−5, 5]	0
$f_1, f_2 = F_9$			
$f_3, f_4 = Weierstrass\ Function^*$			
$f_5, f_6 = F_{11}$			
$f_7, f_8 = F_{10}$			
$f_9, f_{10} = F_1\ [\sigma_1, \sigma_2, \sigma_3, ..., \sigma_{10}] = [0.1,\ 0.2,\ 0.3, 0.4,\ 0.5,\ 0.6,\ 0.7,\ 0.8,\ 0.9,\ 1]$			
$[\lambda_1, \lambda_2, \lambda_3, ..., \lambda_{10}] = \begin{bmatrix} 0.1 * 1/5,\ 0.2 * 1/5,\ 0.3 * 5/0.5,\ 0.4 * 5/0.5,\ 0.5 * 5/100, \\ 0.6 * 5/100,\ 0.7 * 5/32,\ 0.8 * 5/32,\ 0.9 * 5/100,\ 1 * 5/100 \end{bmatrix}$			

*Weierstrass Function may be found in Appendix

original ALO and other reported algorithms. The comparative convergence profiles for the composite test functions are presented in Fig. 8, which shows that the proposed QOCALO algorithm offers the best convergence mobility among all the reported nature inspired algorithms.

5 Solution of placement and sizing problem of DGs in radial distribution network

The aim of this section is to check whether the proposed QOCALO algorithm is capable to solve real world

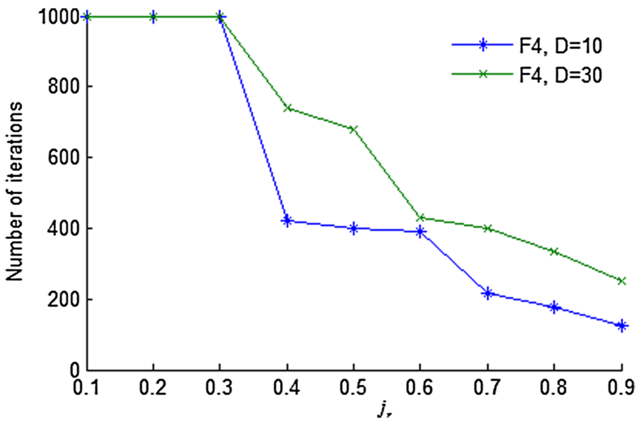


Fig. 3 Variation in number of iterations required for convergence with varying j_r

constrained engineering optimization problems rather than mathematical test problems. In pursuit of this, the problem of optimal placement and sizing of DGs (as first application of the proposed QOCALO algorithm) is solved using the proposed QOCALO algorithm. The considered problem is a power system optimization problem and is a topic of research since the last decade. Various metaheuristic algorithms have been utilized by the researchers to solve this problem and some of these reported algorithms are GA [64–66], PSO [67, 68], DE [69], ACO [70], ABC [71] and GA-PSO [72].

5.1 Mathematical formulation of the problem

The multi-objective problem of finding the optimal locations and sizes of real power DGs is associated with the reduction of real and reactive power loss indices and improvement of voltage profile while keeping the line MVA flow within its specified limit. The detailed mathematical formulation of the objective functions is provided in the following seven sub-sections.

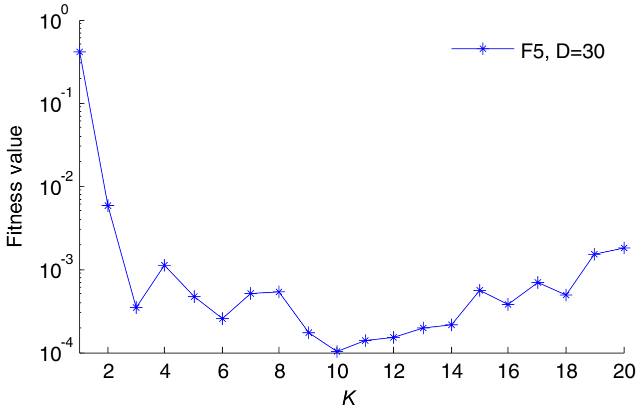


Fig. 4 Variation in fitness value with varying K

5.1.1 Real and reactive power loss indices

The real and reactive power loss indices of the system are defined in (22) and (23), respectively [66]

$$ILP = \frac{P_{LDG}}{P_L} \tag{22}$$

$$ILQ = \frac{Q_{LDG}}{Q_L} \tag{23}$$

where P_L and Q_L are the total real and reactive power losses of the distribution system without DGs while P_{LDG} and Q_{LDG} are the total real and reactive power losses of the system after installing DGs. The real power loss (P_{loss}) and the reactive power loss (Q_{loss}) are calculated, in order, using (24) and (25)

$$P_{loss} = \sum_{i=1}^{nbr} R_i \, |I_i|^2 \tag{24}$$

$$Q_{loss} = \sum_{i=1}^{nbr} X_i \, |I_i|^2 \tag{25}$$

where $R_i$and X_i are the resistance and reactance of the i-th branch; I_i is the branch current of the i-th branch and *nbr* is the total number of branches in the network.

5.1.2 Voltage profile index

Installation of DGs in the distribution system greatly improves the voltage at each node of the system (except the first node). The voltage profile index is defined as

$$IVD = \max_{i=2}^{n} \left(\frac{\left|\overline{V}_{no\min al}\right| - \left|\overline{V}_i\right|}{\left|\overline{V}_{no\min al}\right|} \right) \tag{26}$$

where $\overline{V}_{no\min al}$ is 1.03 p.u. for 38-node system and 1.00 p.u. for 69-node system [73]; $\overline{V}_i$ is the voltage at the i-th node of the system and n is the total number of nodes.

5.1.3 MVA capacity index

Incorporation of DGs in distribution networks, significantly, changes the power flow in different sections of the network.

Table 4 Comparative results for 10 dimensional unimodal test functions

Function	Metrics	ALO [Studied]	MFO [Studied]	MVO [Studied]	WOA [Studied]	CSA [Studied]	QOCALO [Proposed]
F_1	Mean (Rank)	2.86E-09 (5)	3.38E-30 (3)	0.0047 (6)	6.33E-157 (2)	1.83E-09 (4)	0 (1)
	SD	1.44E-09	7.61E-30	0.0021	2.00E-156	1.78E-09	0
	Best	7.10E-10	3.24E-32	0.0014	5.19E-174	8.81E-11	0
	Worst	6.91E-09	3.51E-29	0.0080	6.33E-156	5.65E-09	0
F_2	Mean (Rank)	2.91E-05 (4)	1.20E-18 (3)	0.0218 (6)	4.28E-107 (2)	8.41E-05 (5)	0 (1)
	SD	2.31E-05	1.69E-18	0.0080	1.09E-106	8.33E-05	0
	Best	8.91E-06	2.87E-20	0.0131	1.81E-114	8.33E-06	0
	Worst	8.12E-05	5.97E-18	0.0391	3.45E-106	2.94E-04	0
F_3	Mean (Rank)	1.36E-05 (3)	8.13E-07 (2)	0.0237 (5.5)	0.0237 (5.5)	1.47E-05 (4)	0 (1)
	SD	2.57E-05	1.77E-06	0.0153	0.0278	1.57E-05	0
	Best	5.01E-07	6.35E-11	0.0072	1.11E-05	1.09E-06	0
	Worst	1.02E-04	7.61E-06	0.0537	7.17E-02	4.23E-05	0
F_4	Mean (Rank)	1.27E-04 (2)	0.0525 (6)	0.0395 (5)	5.98E-04 (4)	1.44E-04 (3)	0 (1)
	SD	1.20E-04	0.0891	0.0152	0.0010	9.23E-05	0
	Best	2.49E-05	1.39E-06	0.0217	4.65E-09	1.95E-05	0
	Worst	4.51E-04	0.2767	0.0726	0.0027	4.11E-04	0
F_5	Mean (Rank)	8.9049 (4)	4.27E+00 (2)	1.23E+01 (6)	5.89E+00 (3)	1.09E+01 (5)	3.78E-04 (1)
	SD	0.0129	2.05E+00	2.06E+01	1.54E-01	2.14E+01	6.97E-04
	Best	8.8788	5.70E-03	1.44E+00	5.62E+00	2.32E+00	2.93E-06
	Worst	8.9205	6.3775	70.3947	6.0944	87.9675	1.62E-03
F_6	Mean (Rank)	2.47E-09 (4)	3.76E-30 (1)	3.90E-03 (6)	3.04E-05 (5)	1.64E-09 (3)	2.17E-10 (2)
	SD	8.70E-10	7.66E-30	1.20E-03	1.28E-05	1.37E-09	1.64E-10
	Best	1.14E-09	1.23E-32	1.90E-03	1.09E-05	3.68E-10	2.46E-11
	Worst	4.37E-09	3.34E-29	6.30E-03	4.98E-05	4.84E-09	5.31E-10
F_7	Mean (Rank)	0.0062 (6)	4.60E-03 (5)	1.80E-03 (4)	2.85E-04 (2)	1.70E-03 (3)	4.21E-05 (1)
	SD	0.0020	2.60E-03	1.00E-03	2.12E-04	1.00E-03	3.70E-05
	Best	0.0024	2.00E-03	9.18E-04	9.71E-06	6.98E-04	4.87E-06
	Worst	0.0092	1.26E-02	3.90E-03	6.79E-04	4.20E-03	1.14E-04
Average rank		4	3.14	5.5	3.36	3.86	1.14
Overall rank		5	2	6	3	4	1

In order to avoid overloading of the lines, it is very important to keep the line MVA flow within its maximum allowable limits. Hence, the MVA capacity index is defined in (27)

$$IC = \max_{i=1}^{nbr} \left(\frac{|\overline{S}_{ij}|}{|\overline{CS}_{ij}|} \right) \tag{27}$$

where $\overline{S}_{ij}$ is the MVA flow in the line connecting nodes i and j while $\overline{CS}_{ij}$ is the MVA capacity limit of the line connecting nodes i and j.

5.1.4 Voltage stability index (VSI)

The VSI of distribution system is defined as in (28) [74]

$$VSI(m2) = |V(m1)|^4 - 4[P(m2)x(jj) - Q(m2)r(jj)]^2 \\ -4[P(m2)r(jj) + Q(m2)x(jj)]\,|V(m1)|^2 \tag{28}$$

where $m1$ is the sending end node, $m2$ is the receiving end node, jj is the branch number, $P(m2)$ is the real power load fed through node $m2$, $Q(m2)$ is the reactive power load fed through node $m2$, $r(jj)$ is the resistance of the branch jj, $x(jj)$ is the reactance of the branch jj and $V(m1)$ is the voltage at node $m1$.

For stable operation of the system, the value of VSI should be greater than zero. The node having the lowest value of VSI ($VSI_{\min}$) is the weakest node of the system in terms of voltage stability and is more vulnerable to voltage collapse. Therefore, the objective should be the improvement of the value of $VSI_{\min}$.

5.1.5 Multi-objective problem formulation

In the present work, the problem of placement and sizing of DGs is considered as a multi-objective optimization

Table 5 Comparative results for 30 dimensional unimodal test functions

Function	Metrics	ALO [Studied]	MFO [Studied]	MVO [Studied]	WOA [Studied]	CSA [Studied]	QOCALO [Proposed]
F_1	Mean (Rank)	2.21E-08 (3)	2.76E-04 (4)	3.17E-01 (6)	6.63E-161 (2)	1.44E-02 (5)	0 (1)
	SD	3.79E-09	2.78E-04	1.43E-01	9.95E-161	6.70E-03	0
	Best	1.52E-08	3.27E-05	1.73E-01	1.50E-173	4.50E-03	0
	Worst	2.76E-08	8.71E-04	6.95E-01	2.80E-160	2.79E-02	0
F_2	Mean (Rank)	6.30E-05 (3)	4.00E+01 (6)	3.41E-01 (4)	1.32E-108 (2)	1.15E+00 (5)	0 (1)
	SD	4.36E-06	2.25E+01	7.42E-02	2.41E-108	5.69E-01	0
	Best	5.41E-05	1.00E+01	2.62E-01	1.70E-114	4.94E-01	0
	Worst	6.91E-05	8.00E+01	4.96E-01	6.15E-108	2.68E+00	0
F_3	Mean (Rank)	3.49E-08 (2)	1.58E+04 (6)	4.91E+01 (4)	1.08E+04 (5)	4.04E+01 (3)	0 (1)
	SD	5.51E-09	1.40E+04	2.13E+01	6.83E+03	1.85E+01	0
	Best	2.31E-08	4.74E+02	2.11E+01	7.57E+02	1.23E+01	0
	Worst	4.08E-08	3.68E+04	8.88E+01	1.82E+04	7.37E+01	0
F_4	Mean (Rank)	5.74E-05 (2)	6.70E+01 (6)	9.23E-01 (3)	6.96E+00 (5)	2.53E+00 (4)	0 (1)
	SD	2.55E-06	9.04E+00	3.30E-01	6.73E+00	7.95E-01	0
	Best	5.28E-05	4.48E+01	6.48E-01	1.68E-02	1.37E+00	0
	Worst	6.09E-05	7.94E+01	1.71E+00	1.76E+01	3.58E+00	0
F_5	Mean (Rank)	23.5503 (2)	1.87E+04 (6)	4.19E+01 (4)	2.68E+01 (3)	9.84E+01 (5)	6.18E-04 (1)
	SD	6.84E+00	3.66E+04	2.38E+01	1.01E-01	1.23E+02	1.00E-03
	Best	6.10E+00	4.42E+00	2.84E+01	2.67E+01	2.61E+01	3.43E-07
	Worst	2.83E+01	9.00E+04	1.03E+02	2.70E+01	4.89E+02	3.40E-03
F_6	Mean (Rank)	9.83E-06 (2)	2.24E-04 (3)	3.29E-01 (6)	1.85E-02 (5)	1.43E-02 (4)	1.16E-08 (1)
	SD	9.28E-06	2.85E-04	9.70E-02	9.10E-03	1.02E-02	3.96E-09
	Best	7.08E-07	3.29E-05	2.01E-01	5.10E-03	5.10E-03	4.85E-09
	Worst	3.62E-05	1.00E-03	5.29E-01	3.09E-02	4.34E-02	1.92E-08
F_7	Mean (Rank)	5.82E-05 (2)	1.37E-01 (6)	1.63E-02 (4)	3.97E-04 (3)	2.22E-02 (5)	1.33E-05 (1)
	SD	3.72E-05	7.26E-02	6.60E-03	2.07E-04	8.10E-03	1.60E-05
	Best	3.08E-05	4.82E-02	9.50E-03	1.29E-04	9.20E-03	5.52E-07
	Worst	1.55E-04	3.08E-01	2.92E-02	8.03E-04	3.70E-02	5.55E-05
Average rank		2.28	5.28	4.43	3.57	4.43	1
Overall rank		2	6	4	3	5	1

problem taking into account all the separate objective functions, as discussed in the previous sub-sections. The fitness function of the current problem may be defined as of (29) [73]

$$MOF = \alpha_1 \cdot ILP + \alpha_2 \cdot ILQ + \alpha_3 \cdot IVD + \alpha_4 \cdot IC + \alpha_5 \cdot \left(\frac{1}{VSI_{\min}}\right) \tag{29}$$

where $\sum_{k=1}^{5} \alpha_k = 1 \wedge \alpha_k \in [0, 1]$. The values of these weighting factors (α_k) are taken from [73].

The fitness function is minimized subject to a variety of operational constraints to be satisfied. These operational constraints are listed in the Sections 5.1.6 and 5.1.7

5.1.6 Equality constraint

The equality constraint of the current problem is defined in (30)

$$P_{SS} = \sum_{i=2}^{n} P_D(i) + \sum_{j=1}^{nbr} P_{loss}(j) - \sum_{k=1}^{ndg} P_{DG}(k) \tag{30}$$

where $P_D(i)$ is the real power demand of the i-th node, $P_{loss}(j)$ is the real power loss of the j-th branch, $P_{DG}(k)$

Table 6 Comparative results for 200 dimensional unimodal test functions

Function	Metrics	ALO [Studied]	MFO [Studied]	MVO [Studied]	WOA [Studied]	CSA [Studied]	QOCALO [Proposed]
F_1	Mean (Rank)	7.56E-05 (3)	4.75E+04 (6)	7.7448 (5)	0 (1.5)	0.1319 (4)	0 (1.5)
	SD	2.90E-05	2.07E+04	0.6658	0	0.0150	0
	Best	5.31E-05	2.05E+04	6.4274	0	0.1119	0
	Worst	1.26E-04	7.42E+04	8.7106	0	0.1512	0
F_2	Mean (Rank)	111.7400 (4)	308.7839 (5)	4.82E+43 (6)	0 (1.5)	9.2010 (3)	0 (1.5)
	SD	16.5900	68.9071	1.16E+44	0	1.2435	0
	Best	88.9800	172.4439	5.19E+11	0	6.3217	0
	Worst	132.5200	404.0773	3.66E+44	0	10.3028	0
F_3	Mean (Rank)	1.56E-03 (2)	2.91E+05 (5)	6.15E+04 (4)	1.25E+06 (6)	722.729 (3)	0 (1)
	SD	2.10E-03	9.38E+04	5.68E+03	2.12E+05	42.1527	0
	Best	3.57E-04	1.72E+05	5.46E+04	8.92E+05	649.9713	0
	Worst	5.20E-03	4.59E+05	7.06E+04	1.60E+06	788.3612	0
F_4	Mean (Rank)	29.02 (3)	95.1487 (6)	52.7664 (4)	85.3655 (5)	6.2327 (2)	0 (1)
	SD	2.6892	1.0824	3.1854	16.6892	0.4390	0
	Best	25.9900	92.5989	47.8223	39.2006	5.4510	0
	Worst	32.8500	96.4829	58.2010	95.7731	6.9401	0
F_5	Mean (Rank)	459.2400 (4)	9.16E+07 (6)	836.1108 (5)	194.0499 (2)	290.5407 (3)	0.0504 (1)
	SD	5.6670	1.32E+08	262.4703	0.2314	62.1739	0.0458
	Best	452.8000	3.90E+05	497.2091	193.5862	206.3859	0.0158
	Worst	467.2500	4.01E+08	1.32E+03	194.2968	386.4816	0.1253
F_6	Mean (Rank)	1.19E-04 (2)	4.14E+04 (6)	7.9594 (5)	0.0144 (3)	0.1334 (4)	8.0340e-05 (1)
	SD	1.33E-04	1.75E+04	0.8460	0.0022	0.0305	1.34E-04
	Best	5.64E-05	1.78E+04	6.9024	0.0115	0.0998	1.33E-05
	Worst	3.58E-04	7.82E+04	9.5409	0.0189	0.1905	3.21E-04
F_7	Mean (Rank)	0.4154 (5)	429.8194 (6)	0.1946 (4)	1.23E-04 (2)	0.1193 (3)	4.9180e-06 (1)
	SD	0.0826	342.6595	0.0262	9.91E-05	0.0106	4.30E-06
	Best	0.3506	99.9309	0.1521	2.84E-06	0.1039	2.60E-06
	Worst	0.5547	1.28E+03	0.2237	2.83E-04	0.1331	1.26E-05
Average rank		3.28	5.71	4.71	3	3.14	1.14
Overall rank		4	6	5	2	3	1

is the real power generated by the k-th DG, ndg is the total number of DGs and P_{SS} is the total real power delivered by the sub-station.

5.1.7 Inequality constraints

The inequality constraints of the current problem are defined in (31)–(33)

$$P_{DG}^{\min} \leq P_{DG} \leq P_{DG}^{\max} \tag{31}$$

$$V_i^{\min} \leq V_i \leq V_i^{\max} \tag{32}$$

$$S_{ij} \leq S_{ij}^{\max} \tag{33}$$

where S_{ij} is the thermal capacity of the line connecting nodes i and j. The minimum and the maximum values of V_i for 38-node system are 0.95 p.u. and 1.03 p.u., respectively, while the same for 69-node system are, in order, 0.90 p.u. and 1.00 p.u.

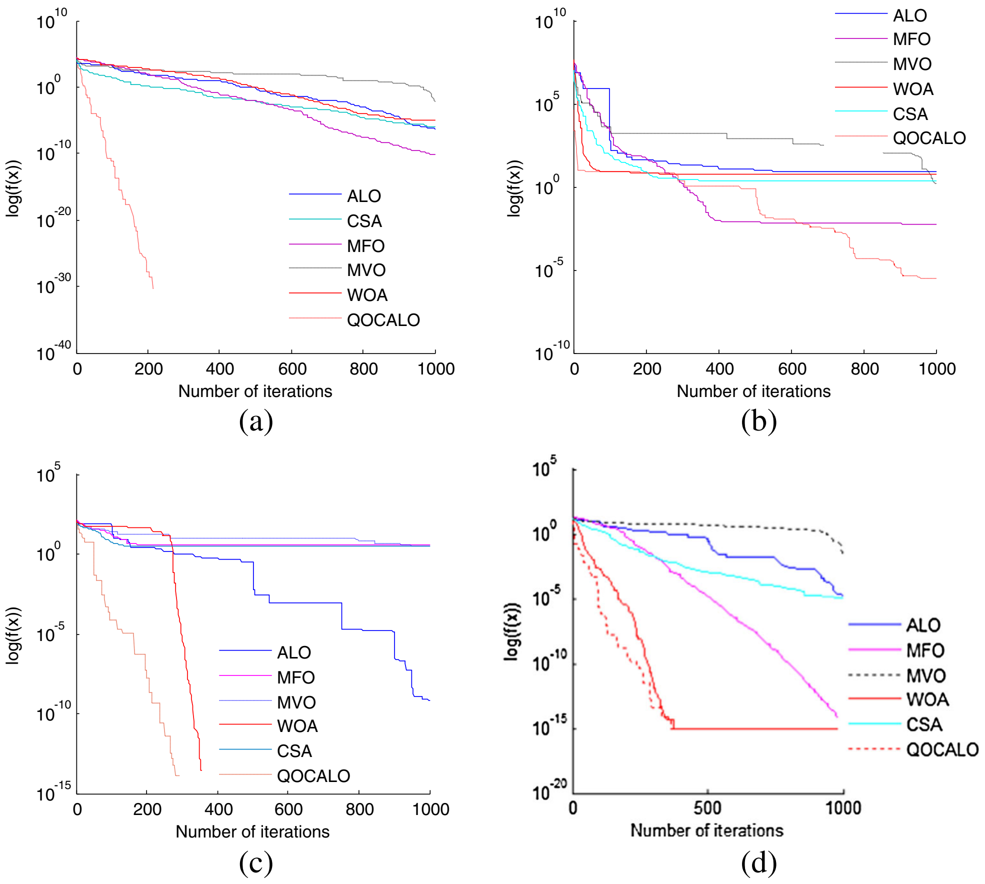
5.2 Simulation results and discussion

The current multi-objective problem is solved individually using the proposed QOCALO algorithm, its original counterpart (i.e. ALO) and other recently developed nature inspired algorithms such as MFO, MVO, WOA and CSA and the performances are compared to those yielded by the chaotic ABC (CABC) algorithm available in [73]. The simulations are carried out in the same computing environment

Table 7 Results of Wilcoxon rank-sum test for unimodal test functions

Function	Dimension		ALO vs QOCALO	MFO vs QOCALO	MVO vs QOCALO	WOA vs QOCALO	CSA vs QOCALO
F_1	10	p-value	6.38E-05	6.38E-05	6.38E-05	6.99E-04	6.38E-05
		Sign	+	+	+	+	+
F_2		p-value	6.38E-05	6.38E-05	6.38E-05	6.99E-04	6.38E-05
		Sign	+	+	+	+	+
F_3		p-value	6.38E-05	6.38E-05	6.38E-05	6.99E-04	6.38E-05
		Sign	+	+	+	+	+
F_4		p-value	6.38E-05	6.38E-05	6.38E-05	6.99E-04	6.38E-05
		Sign	+	+	+	+	+
F_5		p-value	1.82E-04	1.82E-04	1.82E-04	0.032	1.82E-04
		Sign	+	+	+	+	+
F_6		p-value	1.82E-04	1.82E-04	1.82E-04	0.032	7.68E-04
		Sign	+	-	+	+	+
F_7		p-value	1.82E-04	1.82E-04	1.82E-04	0.007	1.82E-04
		Sign	+	+	+	+	+
F_1	30	p-value	6.38E-05	6.38E-05	6.34E-05	6.38E-05	6.38E-05
		Sign	+	+	+	+	+
F_2		p-value	6.38E-05	6.15E-05	6.38E-05	6.38E-05	6.38E-05
		Sign	+	+	+	+	+
F_3		p-value	6.38E-05	6.38E-05	6.38E-05	6.38E-05	6.38E-05
		Sign	+	+	+	+	+
F_4		p-value	6.38E-05	6.38E-05	6.38E-05	6.38E-05	6.38E-05
		Sign	+	+	+	+	+
F_5		p-value	1.82E-04	1.82E-04	1.82E-04	1.81E-04	1.82E-04
		Sign	+	+	+	+	+
F_6		p-value	1.82E-04	1.82E-04	1.82E-04	1.82E-04	1.82E-04
		Sign	+	+	+	+	+
F_7		p-value	0.0013	1.82E-04	1.82E-04	1.82E-04	1.82E-04
		Sign	+	+	+	+	+
F_1	200	p-value	0.0079	0.0079	0.0079	1	0.0079
		Sign	+	+	+	=	+
F_2		p-value	0.0079	0.0079	0.0079	1	0.0079
		Sign	+	+	+	=	+
F_3		p-value	0.0079	0.0079	0.0079	0.0079	0.0079
		Sign	+	+	+	+	+
F_4		p-value	0.0079	0.0079	0.0079	0.0079	0.0079
		Sign	+	+	+	+	+
F_5		p-value	0.0079	0.0079	0.0079	0.0079	0.0079
		Sign	+	+	+	+	+
F_6		p-value	0.0952	0.0079	0.0079	0.0079	0.0079
		Sign	=	+	+	+	+
F_7		p-value	0.0079	0.0079	0.0079	0.0952	0.0079
		Sign	+	+	+	=	+
Total count of +/=/-			20/1/0	20/0/1	21/0/0	18/3/0	21/0/0

with the same algorithmic parameters as in the case of function optimization problems. The algorithms are run for 100 independent trial runs and each run is associated with 100 numbers of iterations.



(a)

(b)

(c)

(d)

Fig. 5 Comparative convergence profiles of fitness function value for 10 dimensional functions **a** F3, **b** F5, **c** F9 and **d** F10

5.2.1 Case study 1: 38-node system

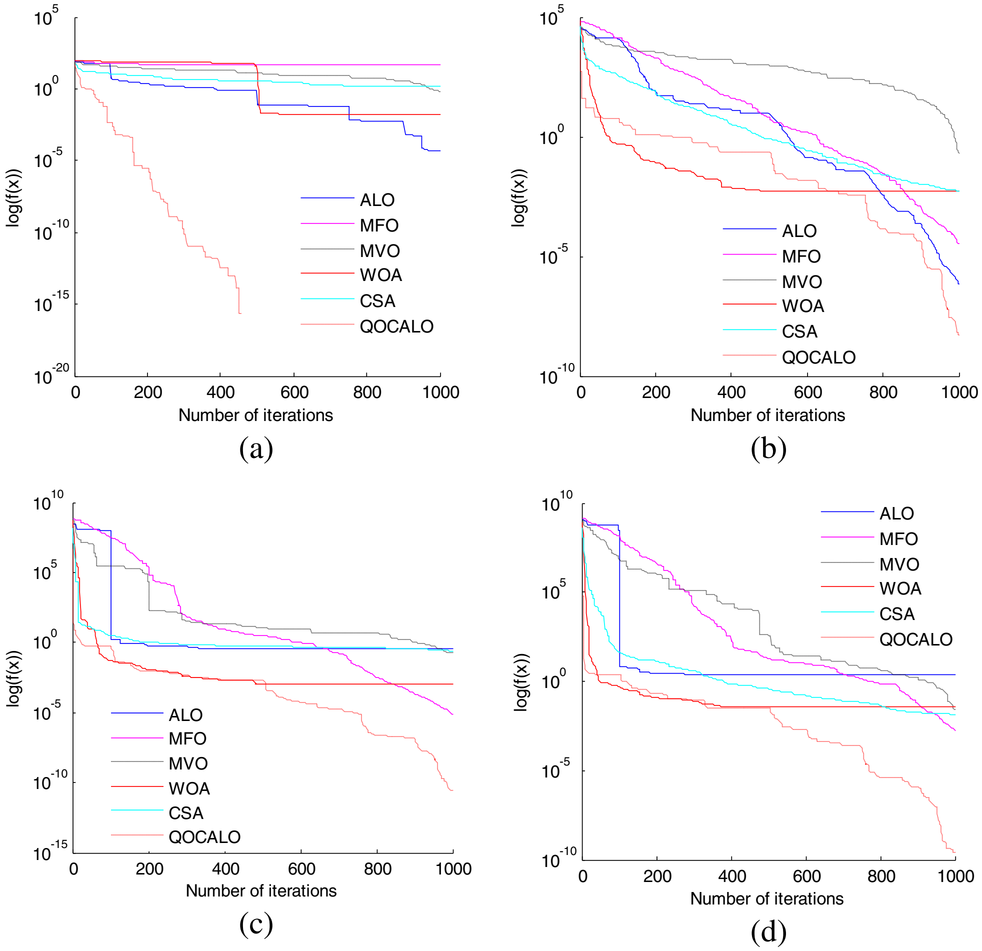
The effectiveness of the proposed QOCALO is tested by implementing it on the 38-node radial distribution system. The 38-node radial distribution network consists of 38 nodes and 37 branches. The details of branch and load data for this test system may be found in [66]. The system has the base values of 100 MVA and 23 kV. The total real and reactive power losses of this system without the installation of DGs are 0.20206 p.u. and 0.13474 p.u., respectively. In the current problem, three real power DGs with unity power factor are considered to be incorporated in the distribution network. The sizes of the DGs are considered in the range 0 - 1.2 p.u. [73].

The simulation results yielded by the proposed QOCALO and the other algorithms for this test system are provided in Table 14. From Table 14, it may be observed that the proposed QOCALO yields better values of real and reactive power losses after successfully integrating three DGs into the system, as compared to the basic ALO, MFO, MVO, WOA and CSA. Also, the proposed QOCALO algorithm successfully minimizes the values of the indices *ILP* and *ILQ*, as compared to the basic ALO, MFO, MVO, WOA, CSA and CABC [73]. It may further be noted that the proposed QOCALO yields better value of $VSI_{\min}$ than the basic ALO, MVO and CSA. The line MVA flows are within their respective limits for both the proposed QOCALO and the other reported algorithm. Finally, the proposed algorithm produces better overall fitness value compared to the other recent algorithms except MFO and CABC. Hence, it may be stated that the proposed QOCALO algorithm successfully finds the optimal locations and sizes of DGs while satisfying all the operating constraints and performs better than the basic ALO and some of the other recently developed algorithms. The comparative convergence profile of the proposed QOCALO and the other reported algorithms is shown in Fig. 9.

5.2.2 Case study 2: 69-node system

The 69-node radial distribution system has 69 nodes and 68 branches. The details of system branch and load data for this test system may be found in [73]. The base values for

Fig. 6 Comparative convergence profiles of fitness function value for 30 dimensional functions **a** F4, **b** F6, **c** F12 and **d** F13



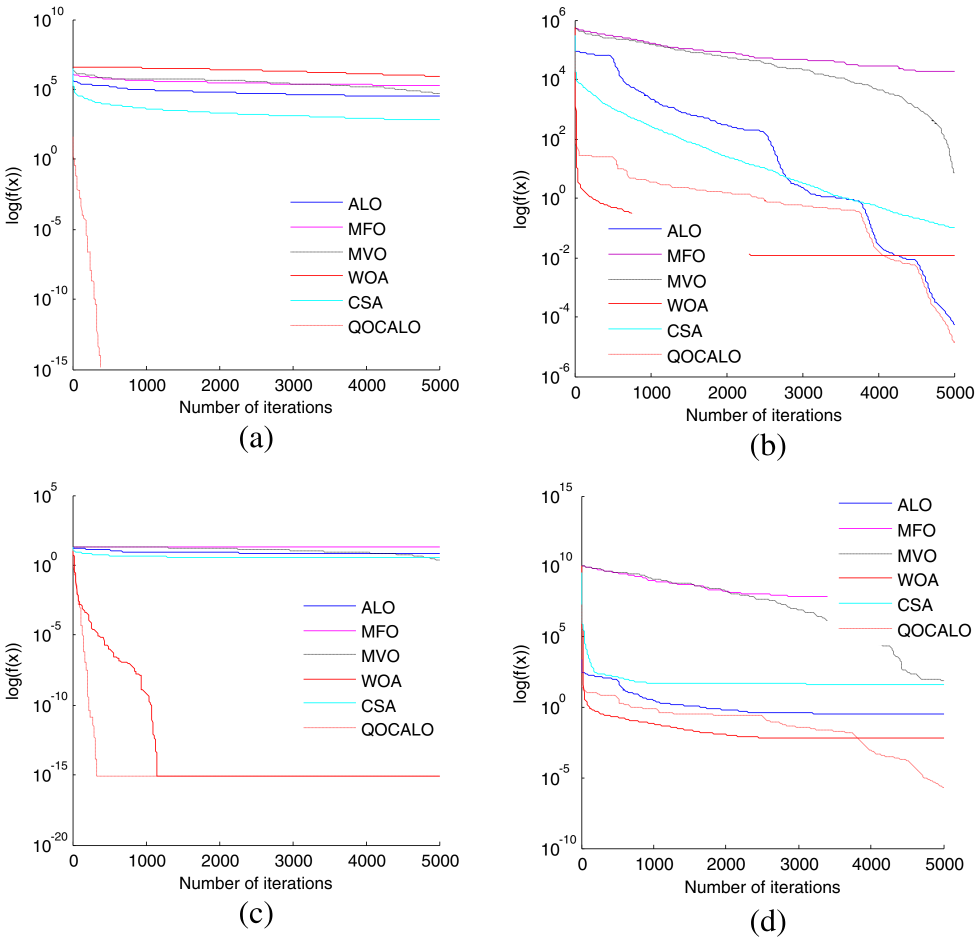
this system are 100 MVA and 12.66 kV. The total real and reactive power demand for this test system are 3.8021 MW and 2.6945 MVAr, respectively. The real and reactive power losses of the system before installation of DGs are 224.97 kW and 102.12 kVAr, respectively. Alike the previous case, in this case also three numbers of real power DG units are incorporated into the system to minimize the power loss and to improve the voltage profile. The DGs are operated at unity power factor and the sizes of the DGs are in the range 0 - 1.2 MW [73].

The simulation results for this test system are given in Table 15. From Table 15, it may be noted that the proposed QOCALO algorithm yields lesser values of the indices *ILP* and *ILQ* as compared to the basic ALO and the other reported algorithms except CABC. Both the real and the reactive power losses of the system offered by the proposed QOCALO are lesser than the losses offered by the original ALO and other reported algorithms except CABC. Also, it may be found from Table 15 that the proposed algorithm yields the best values for the indices *IVD* and $VSI_{\min}$ as compared to the basic ALO, CABC [73] and other reported algorithms. In the context of solving the current multi-objective optimization problem, the proposed QOCALO algorithm produces minimum value of the fitness function compared to other algorithms referred therein. Hence, it may be remarked that the proposed QOCALO algorithm performs better than its original counterpart and produces competitive results to other reported algorithms in solving the current problem. A comparative convergence profile of fitness function value for the considered case is portrayed in Fig. 10.

6 Solution of CM problem in power transmission system

The aim of this section is to minimize the congestion cost in power transmission system by rescheduling the real power outputs of the generators while satisfying all the operating constraints. This may be treated as the second real world

Fig. 7 Comparative convergence profiles of fitness function value for 200 dimensional functions **a** F3, **b** F6, **c** F10 and **d** F13



(a)

(b)

(c)

(d)

engineering optimization application of the proposed algorithm. The considered power system optimization problem is solved using the proposed QOCALO algorithm and the simulation results are compared to those offered by the original ALO and some other recently reported algorithms, such as, BA, chaotic BA, flower pollination algorithm (FPA) and FA [75].

6.1 Mathematical problem formulation

The transmission CM problem may be formulated as in (34) [76]

$$\text{Minimize } C = \sum_{j\varepsilon N_g} (C_j \Delta P^{+}_{Gj} + D_j \Delta P^{-}_{Gj}) \quad \$/\text{h} \tag{34}$$

where C represents the total CM cost ($/h), C_j and D_j represent the increment and decrement price bids ($/MWh), respectively, submitted by the generating company for j-th generator, ΔP^{+}_{Gj} and ΔP^{-}_{Gj} represent the real power increment of j-th generator (MW) and the real power decrement of j-th generator (MW), respectively and N_g is the total number of generators.

6.1.1 Equality constraints

The equality constraints of the current problem are stated in (35)–(38) [75, 76]

$$P_{Gk} - P_{Dk} = |V_j|\,|V_k|\,|Y_{kj}|\cos(\delta_k - \delta_j - \theta_{kj})\,; \qquad j = 1, 2, ..., N_b \tag{35}$$

$$Q_{Gk} - Q_{Dk} = |V_j|\,|V_k|\,|Y_{kj}|\sin(\delta_k - \delta_j - \theta_{kj})\,; \qquad j = 1, 2, ..., N_b \tag{36}$$

$$P_{Gk} = P^{C}_{Gk} + \Delta P^{+}_{Gk} - \Delta P^{-}_{Gk}\,; \qquad k = 1, 2, ..., N_g \tag{37}$$

$$P_{Dj} = P^{C}_{Dj}\,; \qquad j = 1, 2, ..., N_d \tag{38}$$

Table 8 Comparative results for 10 dimensional multimodal test functions

Function	Metrics	ALO [Studied]	MFO [Studied]	MVO [Studied]	WOA [Studied]	CSA [Studied]	QOCALO [Proposed]
F_8	Mean (Rank)	−2.55E+03 (6)	−3.34E+03 (2)	−2.89E+03 (4)	−3.94E+03 (1)	−2.78E+03 (5)	−3.17E+03 (3)
	SD	576.6912	399.4665	330.8130	328.4855	248.0872	406.2075
	Best	−3.97E+03	−4.07E+03	−3.52E+03	−4.19E+03	−3.32E+03	−3.72E+03
	Worst	−2.03E+03	−2.64E+03	−2.41E+03	−3.35E+03	−2.47E+03	−2.47E+03
F_9	Mean (Rank)	1.29E-09 (3)	20.0022 (6)	9.2552 (5)	0 (1.5)	5.3064 (4)	0 (1.5)
	SD	3.52E-10	10.2337	5.0737	0	2.1162	0
	Best	6.72E-10	3.9798	2.9874	0	2.9849	0
	Worst	1.88E-09	46.7629	20.8950	0	8.9546	0
F_{10}	Mean (Rank)	2.35E-05 (4)	4.81E-15 (3)	5.35E-01 (5)	1.95E-15 (2)	1.15E+00 (6)	8.88E-16 (1)
	SD	6.15E-06	1.12E-15	0.8561	1.72E-15	0.9438	0
	Best	1.31E-05	4.44E-15	1.85E-02	8.88E-16	1.10E-05	8.88E-16
	Worst	3.00E-05	7.99E-15	2.01E+00	4.44E-15	72.81E+00	8.88E-16
F_{11}	Mean (Rank)	0.2100 (5)	0.1578 (4)	0.2755 (6)	0 (1.5)	0.1281 (3)	0 (1.5)
	SD	0.1073	8.18E-02	0.1152	0	0.0600	0
	Best	0.0763	0.0394	0.1078	0	0.0246	0
	Worst	0.4771	0.2829	0.4471	0	0.2630	0
F_{12}	Mean (Rank)	1.14E-10 (3)	6.30E-31 (1)	2.40E-04 (5)	6.62E-05 (4)	1.87E-01 (6)	4.38E-12 (2)
	SD	6.23E-11	9.92E-31	1.08E-04	3.46E-05	3.86E-01	4.88E-12
	Best	4.59E-11	4.91E-32	1.24E-04	2.70E-05	3.14E-10	1.72E-13
	Worst	2.32E-10	3.60E-30	4.54E-04	1.28E-04	1.24E+00	1.68E-11
F_{13}	Mean (Rank)	6.66E-10 (2)	2.20E-03 (5)	6.07E-04 (4)	3.11E-04 (3)	5.80E-03 (6)	1.77E-11 (1)
	SD	6.88E-10	4.50E-03	2.90E-04	1.24E-04	6.90E-03	1.28E-11
	Best	1.74E-10	1.35E-32	1.49E-04	8.71E-05	3.93E-09	8.63E-13
	Worst	2.54E-09	1.10E-02	1.10E-03	4.73E-04	2.10E-02	5.18E-11
Average rank		3.83	3.5	4.83	2.17	5	1.67
Overall rank		4	3	5	2	6	1

where P_{Gk} and Q_{Gk} are the real and reactive power generated at bus k respectively; P_{Dk} and Q_{Dk} are the real and reactive load at bus k, respectively; $V_j$and V_k are the voltages at busses j and k, respectively; δ_j and δ_k are the bus voltage angles of busses j and k, respectively; θ_{kj} is the admittance angle of the line connected between busses k and j; N_b and N_d are the total number of busses and loads, respectively; P_{Gk}^C and P_{Dj}^C are the real power produced by generator k and the real power consumed by load bus j, respectively, as obtained by the market clearing procedure.

6.1.2 Inequality constraints

The inequality constraints associated with the current problem are stated in (39)–(43) in line with [75, 76]

$$P_{Gk}^{\min} \le P_{Gk} \le P_{Gk}^{\max}, \quad \forall k \in Ng \tag{39}$$

$$Q_{Gk}^{\min} \le Q_{Gk} \le Q_{Gk}^{\max}, \quad \forall k \in Ng \tag{40}$$

$$(P_{Gk}-P_{Gk}^{\min}) = \Delta P_{Gk}^{\min} \le \Delta P_{Gk} \le \Delta P_{Gk}^{\max} = (P_{Gk}^{\max}-P_{Gk}) \tag{41}$$

$$V_n^{\min} \le V_n \le V_n^{\max}, \quad \forall n \in N_b \tag{42}$$

$$P_{ij} \le P_{ij}^{\max} \tag{43}$$

where the superscripts *min* and *max* represent the minimum and the maximum values of the respected variables, respectively.

6.1.3 Formulation of fitness function

The fitness function (*FF*) for achieving the desired minimum CM cost is given by (44)

$$FF = C + P \tag{44}$$

Table 9 Comparative results for 30 dimensional multimodal test functions

Function	Metrics	ALO [Studied]	MFO [Studied]	MVO [Studied]	WOA [Studied]	CSA [Studied]	QOCALO [Proposed]
F_8	Mean (Rank)	−3.40E+03 (6)	−8.71E+03 (2)	−7.64E+03 (3)	−1.24E+04 (1)	−6.96E+03 (4)	−6.85E+03 (5)
	SD	2.74E+02	8.65E+02	7.33E+02	2.47E+02	7.31E+02	1.76E+03
	Best	−3.85E+03	−1.02E+04	−9.29E+03	−1.26E+04	−8.47E+03	−9.92E+03
	Worst	−2.91E+03	−6.98E+03	−6.73E+03	−1.18E+04	−5.99E+03	−5.42E+03
F_9	Mean (Rank)	1.18E-08 (3)	149.9325 (6)	93.0047 (5)	0 (1.5)	21.1713 (4)	0 (1.5)
	SD	1.09E-09	42.2841	15.2142	0	9.2509	0
	Best	1.04E-08	73.6272	62.7730	0	10.9480	0
	Worst	1.39E-08	239.1442	117.5394	0	46.7674	0
F_{10}	Mean (Rank)	3.65E-05 (3)	1.44E+01 (6)	6.97E-01 (4)	2.66E-15 (2)	3.13E+00 (5)	8.88E-16 (1)
	SD	1.00E-06	8.17E+00	5.06E-01	1.87E-15	6.82E-01	0
	Best	3.49E-05	2.50E-03	1.29E-01	8.88E-16	2.50E+00	8.88E-16
	Worst	3.82E-05	2.00E+01	1.47E+00	4.44E-15	4.81E+00 8.88E-16	
F_{11}	Mean (Rank)	3.83E-08 (3)	4.70E-03 (4)	5.84E-01 (6)	0 (1.5)	9.18E-02 (5)	0 (1.5)
	SD	2.86E-09	1.00E-02	8.88E-02	0	3.11E-02	0
	Best	3.45E-08	1.04E-05	4.73E-01	0	4.62E-02	0
	Worst	4.30E-08	2.99E-02	7.22E-01	0	1.60E-01	0
F_{12}	Mean (Rank)	4.98E-01 (4)	2.10E-01 (3)	1.11E+00 (5)	1.90E-03 (2)	2.64E+00 (6)	3.73E-11 (1)
	SD	7.42E-02	2.60E-01	7.64E-01	4.89E-04	1.74E+00	6.11E-12
	Best	3.96E-01	7.93E-06	1.71E-01	1.20E-03	2.61E-01	2.85E-11
	Worst	6.24E-01	6.55E-01	2.53E+00	2.90E-03	7.29E+00	4.72E-11
F_{13}	Mean (Rank)	2.72E+00 (6)	1.02E-02 (2)	1.65E-01 (5)	9.66E-02 (4)	7.04E-02 (3)	4.57E-10 (1)
	SD	2.10E-01	6.30E-03	3.28E-01	5.50E-02	3.89E-02	1.47E-10
	Best	2.39E+00	1.10E-03	2.64E-02	3.62E-02	1.43E-02	2.76E-10
	Worst	2.99E+00	2.14E-02	1.10E+00	1.77E-01	1.32E-01	7.59E-10
Average rank		4.17	3.83	4.67	2	4.5	1.83
Overall rank		4	3	6	2	5	1

where, P is a penalty function, given by the following equation

$$P = pf_1 \sum_{i=1}^{N_l} P_l + pf_2 \sum_{j=1}^{N_b} P_v + pf_3 \sum_{k=1}^{N_g} P_s \tag{45}$$

where, pf_1, pf_2 and pf_3 are penalty factors which are user defined and N_l is the total number of transmission lines. The values of pf_1, pf_2 and pf_3 are taken as 10,000 [75, 76].

The terms P_l, P_v and P_s are described as of (46)–(48).

$$P_l = \begin{cases} 0 & if\ P_{ij} \le P_{ij}^{\max} \\ \left(P_{ij} - P_{ij}^{\max}\right)^2 & if\ P_{ij} > P_{ij}^{\max} \end{cases} \tag{46}$$

$$P_v = \begin{cases} 0 & if\ V_n^{\min} \le V_n \le V_n^{\max} \\ \left(V_n^{\min} - V_n\right)^2 & if\ V_n < V_n^{\min} \\ \left(V_n - V_n^{\max}\right)^2 & if\ V_n > V_n^{\max} \end{cases} \tag{47}$$

$$P_s = \begin{cases} 0 & if\ P_{Gk}^{\min} \le P_{Gk} \le P_{Gk}^{\max} \\ \left(P_{Gk}^{\min} - P_{Gk}\right)^2 & if\ P_{Gk} < P_{Gk}^{\min} \\ \left(P_{Gk} - P_{Gk}^{\max}\right)^2 & if\ P_{Gk} > P_{Gk}^{\max} \end{cases} \tag{48}$$

6.2 Simulation results and discussion

The CM problem of power transmission system is solved by the proposed QOCALO algorithm and the simulation results are compared with the results reported in [75]. The simulations are carried out in MATLAB with the same system configuration as in the previous cases. The number of antlions is considered to be 40 in line with [75], the value of j_r is set to 0.6 and the value of K is equal to the dimension of the considered problem.

6.2.1 Case study 1: Modified IEEE 30-bus system

The modified IEEE 30-bus system has six generator busses, twenty-four load busses and forty-one transmission lines.

Table 10 Comparative results for 200 dimensional multimodal test functions

Function	Metrics	ALO [Studied]	MFO [Studied]	MVO [Studied]	WOA [Studied]	CSA [Studied]	QOCALO [Proposed]
F_8	Mean (Rank)	−525 (5)	−4.47E+04 (3)	−4.96E+04 (2)	−8.31E+04 (1)	−4.11E+04 (4)	−123.8600 (6)
	SD	1442.5000	7.14E+03	2.11E+03	942.4065	3.78E+03	120.3500
	Best	−44.4652	−5.99E+04	−5.40E+04	−8.38E+04	−4.60E+04	−23.7500
	Worst	−1450	−3.95E+04	−4.70E+04	−8.11E+04	−3.29E+04	−578.3200
F_9	Mean (Rank)	745.4360 (4)	1.25E+03 (5.5)	1.25E+03 (5.5)	0 (1)	82.296 0(3)	4.12E-13 (2)
	SD	4.5418	155.9397	75.8458	0	35.3470	1.75E-13
	Best	740.2000	941.8822	1.12E+03	0	39.9360	2.27E-13
	Worst	752.2400	1.45E+03	1.35E+03	0	159.4928	6.28E-13
F_{10}	Mean (Rank)	6.7662 (4)	19.9384 (6)	7.8885 (5)	3.73E-15 (2)	4.1320 (3)	8.76E-16 (1)
	SD	0.4371	0.0244	8.0975	2.80E-15	0.5341	2.94E-17
	Best	6.2360	19.9009	2.3704	8.88E-16	3.3833	8.23E-16
	Worst	7.3250	19.9666	19.719	7.99E-15	5.2546	8.90E-16
F_{11}	Mean (Rank)	0.0404 (3)	384.1716 (6)	0.9914 (5)	0 (1)	0.1760 (4)	2.19E-16 (2)
	SD	0.0070	116.0675	0.0415	0	0.0272	9.97E-17
	Best	0.0333	218.0512	0.9281	0	0.1398	1.12E-16
	Worst	0.0513	554.8397	1.0472	0	0.2310	3.75E-16
F_{12}	Mean (Rank)	19.9740 (5)	2.18E+08 (6)	8.5388 (4)	6.79E-05 (2)	1.8069 (3)	1.40E-07 (1)
	SD	2.5526	2.76E+08	1.7293	1.07E-05	0.3847	2.35E-07
	Best	17.6000	7.29E+05	6.2706	5.26E-05	1.3528	2.50E-08
	Worst	23.6800	8.32E+08	12.8796	8.85E-05	2.4161	5.62E-07
F_{13}	Mean (Rank)	7.5264 (3)	4.81E+08 (6)	172.8302 (5)	0.0152 (2)	123.1807 (4)	2.77E-06 (1)
	SD	5.3247	4.27E+08	50.3106	0.0081	63.2580	7.07E-07
	Best	1.1359	4.09E+05	74.9779	0.0072	43.2961	1.93E-06
	Worst	15.2586	1.33E+09	241.5425	0.0297	219.0203	3.52E-06
Average rank		4	5.42	4.42	1.5	3.5	2.17
Overall rank		4	6	5	1	3	2

The total real and reactive power loads for this test system are 283.4 MW and 126.2 MVAr, respectively. The system data for the considered test system is taken from [75]. The details of contingencies considered for this case study are provided in Table 16.

In case 1A, congestion has occurred in the lines 1-7 and 7-8 due to the outage of the line 1-2. The real power flows in these congested lines are 147.463 MW and 136.292 MW, respectively, while the maximum allowable limit is 130 MW for both the lines. For secure operation of the system, the line overloads should be alleviated. In the present study, a QOCALO based CM scheme has been adopted to alleviate the line overloads. The simulation results for this case have been provided in Table 17. From Table 17, it may be observed that the proposed QOCALO provides the minimum CM cost compared to other metaheuristic methods. The amount of real power rescheduling (ΔP_G) as yielded by the proposed method for different generators has also been provided in Table 17. A positive value of ΔP_G suggests for increment in generation while a negative value suggests for decrement. The power flow in the previously congested lines comes under its specified limit after the application of the proposed QOCALO based CM technique (refer Table 16).

In case 1B, congestion is created in the system by considering the outage of the line 1-7 along with the increment in real and reactive power loads by 50%. As a result of this, the lines 1-2, 2-8 and 2-9 get overloaded as the power flows in these lines exceed their specified limit (refer Table 16). The QOCALO based CM technique is applied for relieving these line overloads and the simulation results are tabulated in Table 17. From Table 17, it may be noted that the proposed method yields the minimum value of CM cost as compared to other reported algorithms. Also, it may be observed that the power flows in the previously congested lines have been reduced and come under their specified limit (refer Table 16). Hence,

Table 11 Results of Wilcoxon rank-sum test for multimodal test functions

Function	Dimension		ALO vs QOCALO	MFO vs QOCALO	MVO vs QOCALO	WOA vs QOCALO	CSA vs QOCALO
F_8	10	p-value	0.011	0.623	0.140	0.087	0.064
		Sign	+	=	=	=	+
F_9		p-value	6.38E-05	6.38E-05	6.38E-05	1	6.29E-05
		Sign	+	+	+	=	+
F_{10}		p-value	8.68E-05	2.42E-05	8.68E-05	0.073	8.62E-05
		Sign	+	+	+	=	+
F_{11}		p-value	6.38E-05	6.38E-05	6.38E-05	1	6.38E-05
		Sign	+	+	+	=	+
F_{12}		p-value	1.82E-04	1.82E-04	1.82E-04	0.032	1.81E-04
		Sign	+	-	+	+	+
F_{13}		p-value	1.82E-04	0.139	1.82E-04	0.032	1.78E-04
		Sign	+	=	+	+	+
F_8	30	p-value	1.78E-04	0.0209	0.1036	1.78E-04	0.2406
		Sign	+	+	=	+	=
F_9		p-value	6.34E-05	6.38E-05	6.34E-05	1	6.34E-05
		Sign	+	+	+	=	+
F_{10}		p-value	6.34E-05	6.29E-05	6.38E-05	0.0137	6.29E-05
		Sign	+	+	+	+	+
F_{11}		p-value	6.38E-05	6.38E-05	6.34E-05	1	6.38E-05
		Sign	+	+	+	=	+
F_{12}		p-value	1.82E-04	1.82E-04	1.81E-04	1.82E-04	1.82E-04
		Sign	+	+	+	+	+
F_{13}		p-value	1.82E-04	1.82E-04	1.82E-04	1.82E-04	1.82E-04
		Sign	+	+	+	+	+
F_8	200	p-value	0.1508	0.0079	0.0079	0.0079	0.0079
		Sign	=	-	-	-	+
F_9		p-value	0.0079	0.0079	0.0079	0.0079	0.0079
		Sign	+	+	+	-	+
F_{10}		p-value	0.0079	0.0079	0.0079	0.2302	0.0079
		Sign	+	+	+	=	+
F_{11}		p-value	0.0079	0.0079	0.0079	0.0079	0.0079
		Sign	+	+	+	-	+
F_{12}		p-value	0.0079	0.0079	0.0079	0.0079	0.0079
		Sign	+	+	+	+	+
F_{13}		p-value	0.0079	0.0079	0.0079	0.0079	0.0079
		Sign	+	+	+	+	+
Total count of +/= /−			17/1/0	14/2/2	15/2/1	8/7/3	17/1/0

it may be concluded that the proposed method successfully clears out the line congestions in both the cases and also outperforms other algorithms in terms of solution quality.

The comparative convergence profiles of the proposed QOCALO and the basic ALO algorithms in solving the CM problem for the modified IEEE 30-bus system are portrayed in Fig. 11.

6.2.2 Case study 2: Modified IEEE 57-bus system

The modified IEEE 57-bus system has seven generator busses, fifty load busses and eighty transmission lines. The total real and reactive power loads are 1250.8 MW and 336 MVAr, respectively. The details of the system data may be found in [75]. The two different cases have been considered for this case study and the details are provided in Table 16.

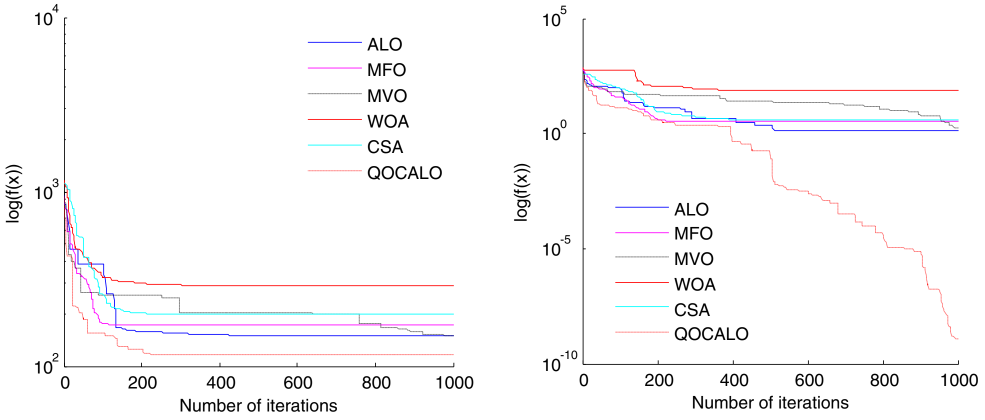
Table 12 Comparative results for composite test functions

Function	Metrics	ALO [Studied]	MFO [Studied]	MVO [Studied]	WOA [Studied]	CSA [Studied]	QOCALO [Proposed]
$CF1$	Mean (Rank)	1.02E-09 (2)	1.44E+02 (6)	7.53E-04 (4)	1.11E+02 (5)	5.54E-09 (3)	2.14E-10 (1)
	SD	5.47E-10	6.76E+01	2.66E-04	9.91E+01	4.54E-09	8.46E-11
	Best	5.07E-10	8.81E+01	3.92E-04	1.80E-01	7.03E-10	1.06E-10
	Worst	1.91E-09	3.00E+02	1.29E-03	3.01E+02	1.16E-08	2.97E-10
$CF2$	Mean (Rank)	19.8246 (2)	70.3808 (4)	145.4541 (5)	192.3626 (6)	69.4540 (3)	13.2784 (1)
	SD	11.6734	56.9920	121.8507	95.5703	82.9355	3.6187
	Best	3.6228	19.3815	27.8514	49.2888	4.7047	7.6684
	Worst	35.9047	172.0806	430.3414	385.1983	227.2069	16.6828
$CF3$	Mean (Rank)	189.2136 (2)	287.5861 (3)	293.9876 (4)	441.4643 (6)	308.0762 (5)	139.4965 (1)
	SD	29.2075	73.3075	141.3331	177.0612	68.5346	20.2512
	Best	148.3986	171.9626	150.4586	288.8680	198.3268	116.3955
	Worst	229.7251	396.0499	635.2630	775.9109	417.9269	166.7740
$CF4$	Mean (Rank)	359.7829 (4)	336.3123 (2)	359.7072 (3)	541.9167 (6)	485.3392 (5)	311.7008 (1)
	SD	63.7326	31.2419	99.1535	115.4126	95.3447	7.8190
	Best	259.7956	299.1165	269.7376	354.0743	315.2255	301.3733
	Worst	481.0727	408.2062	600.0152	730.3820	608.6779	325.4760
$CF5$	Mean (Rank)	4.3852 (2)	82.1068 (4)	143.1680 (5)	189.3241 (6)	27.0009 (3)	2.0568 (1)
	SD	1.7903	85.6326	195.9432	118.8072	40.1711	0.9599
	Best	1.2965	3.25E+00	1.67E+00	7.81E+01	3.91E+00	1.25E-09
	Worst	6.6117	224.7076	505.4436	457.4412	102.9989	2.9999
$CF6$	Mean (Rank)	502.6195 (3)	595.3108 (4)	892.5232 (6)	869.7328 (5)	502.1681 (2)	481.5000 (1)
	SD	1.6268	162.4725	31.6485	120.6725	0.9719	41.4263
	Best	500.3407	500.4224	802.4530	526.4697	500.3335	402.4940
	Worst	505.5645	902.6564	903.0041	913.3921	503.5358	502.1211
Average rank		2.5	3.83	4.5	5.67	3.5	1
Overall rank		2	4	5	6	3	1

Table 13 Results of Wilcoxon rank-sum test for composite test functions

Function		ALO vs QOCALO	MFO vs QOCALO	MVO vs QOCALO	WOA vs QOCALO	CSA vs QOCALO
$CF1$	p-value	5.82E-04	5.82E-04	5.82E-04	5.82E-04	5.82E-04
	Sign	+	+	+	+	+
$CF2$	p-value	0.1195	5.82E-04	5.82E-04	5.82E-04	0.0338
	Sign	=	+	+	+	+
$CF3$	p-value	0.0069	5.82E-04	0.0023	5.82E-04	5.82E-04
	Sign	+	+	+	+	+
$CF4$	p-value	0.2086	0.0023	0.6200	5.82E-04	0.0012
	Sign	=	+	=	+	+
$CF5$	p-value	0.0175	5.82E-04	5.82E-04	5.82E-04	5.82E-04
	Sign	+	+	+	+	+
$CF6$	p-value	0.0175	0.0175	5.82E-04	5.82E-04	0.0070
	Sign	+	+	+	+	+
Total count of $+/=/-$		4/2/0	6/0/0	5/1/0	6/0/0	6/0/0

Fig. 8 Comparative convergence profiles of fitness function value for composite functions **a** CF3 and **b** CF5



(a) (b)

In case 2A, the congestion is created in the system by considering the power flow limits of the lines 5-6 and 6-12 as 175 MW and 35 MW, respectively, instead of their original power flow limits of 200 MW and 50 MW. Thus, the lines 5-6 and 6-12 get overloaded at the base condition, as the power flow in those lines exceeds the new specified limit (refer Table 16). To relieve this line overload, the QOCALO based CM technique is applied and the simulation results are provided in Table 18. From Table 18, it may be noted that the proposed QOCALO yields the minimum cost of CM as compared to other reported algorithms. The power flows in the previously congested lines also come under their specified limit after successful implementation of the QOCALO based CM method (refer Table 16).

In case 2B, the congestion is created in the system by reducing the capacity of the line 2-3 from 85 MW to 20 MW. As a result of this, the line 2-3 becomes overloaded as the power flow in this line is 37.048 MW which is greater than its new specified limit (refer Table 16). The simulation results obtained after applying the proposed QOCALO based CM technique are provided in Table 18. From Table 18, it may be noted that the proposed QOCALO yields the minimum value of CM cost as compared to other reported algorithms. Also, the power flow in the previously congested line is reduced to be settled under its specified limit (refer Table 16). Thus, the proposed QOCALO algorithm outperforms its original counterpart in solving the transmission CM problem in terms of quality of solution.

The comparative convergence profiles of the proposed QOCALO and the basic ALO algorithms in solving the CM problem for the modified IEEE 57-bus system are portrayed in Fig. 12.

Table 14 Comparative results of optimal DG placement for 38-node system offered by different algorithms

Indices	CABC [73]	ALO [Studied]	MFO [Studied]	MVO [Studied]	WOA [Studied]	CSA [Studied]	QOCALO [Proposed]
ILP	0.3561	0.3610	0.3531	0.3785	0.3559	0.3591	0.3520
ILQ	0.3700	0.3759	0.3670	0.3919	0.3685	0.3695	0.3653
IVD	0.0267	0.0614	0.0563	0.0609	0.0562	0.0580	0.0577
IC	0.9854	0.9831	0.9830	0.9910	0.9830	0.9911	0.9830
$VSI_{\min}$	1.01	0.8736	0.8928	0.8752	0.8930	0.8862	0.8874
P_{LDG} (p.u.)	0.0672	0.0729	0.0713	0.0764	0.0719	0.0725	0.0711
Q_{LDG} (p.u.)	0.0466	0.0506	0.0494	0.0528	0.0496	0.0498	0.0492
MOF	0.5295	0.5522	0.5448	0.5624	0.5460	0.5504	0.5451
Optimal	14 (0.7546)	15 (0.5518)	14 (0.7563)	11 (1.0849)	13 (0.8534)	12 (0.9972)	14 (0.7565)
location-size	24 (1.0878)	24 (1.1757)	24 (1.0888)	23 (1.1068)	24 (0.9121)	25 (0.8625)	24 (1.0891)
(p.u.) pair	30 (1.1621)	30 (1.1838)	30 (1.1744)	30 (0.8809)	30 (1.1554)	30 (1.0411)	30 (1.1280)

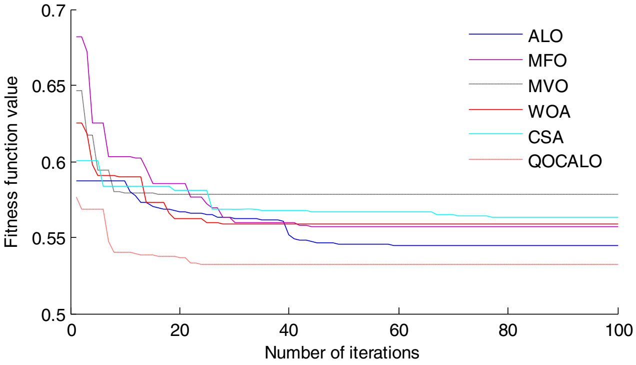


Fig. 9 Comparative convergence profiles of fitness function value for 38-node system pertaining to DG placement and sizing problem

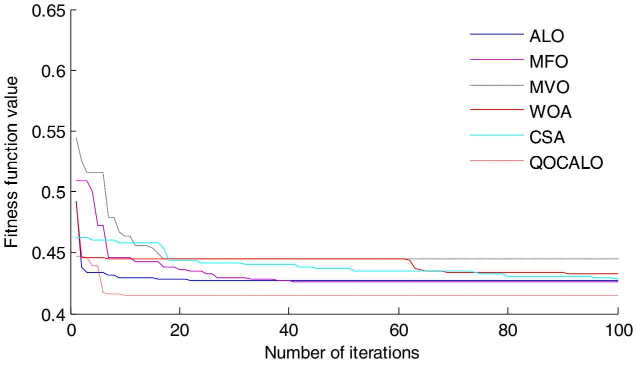


Fig. 10 Comparative convergence profiles of fitness function value for 69-node system pertaining to DG placement and sizing problem

7 Solution of pressure vessel design problem

The pressure vessel design problem is a well known design optimization problem and is considered as the third real world engineering optimization application of the proposed QOCALO algorithm. The objective of this problem is to find out an optimal design of the pressure vessel with the least fabrication cost. The details of the problem may be found in [11]. This design optimization problem can be formulated as follows:

Consider $X = [x_1\, x_2\, x_3\, x_4]$

$$\text{Minimize } f(X) = 0.6224 x_1 x_3 x_4 + 1.7781 x_2 x_3^2 + 3.1661 x_1^2 x_4 + 19.84 x_1^2 x_3 \tag{49}$$

$$\text{Subject to} \left.\begin{array}{l} g_1(X) = -x_1 + 0.0193 x_3 \le 0 \\ g_2(X) = -x_2 + 0.0095 x_3 \le 0 \\ g_3(X) = -\pi x_3^2 x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \le 0 \\ g_4(X) = x_4 - 240 \le 0 \end{array}\right\} \tag{50}$$

The variable range may be specified as in (51) [11].

$$\left.\begin{array}{l} 1 \times 0.0625 \le x_1, x_2 \le 99 \times 0.0625 \\ 10 \le x_3, x_4 \le 200 \end{array}\right\} \tag{51}$$

This problem is solved using the proposed QOCALO algorithm and the simulation results are compared to other metaheuristic algorithms, such as, MFO, GSA, PSO, GA, DE and ACO. The number of antlions is considered to be 40, the value of j_r is set to 0.6 and the value of K is equal to the dimension of the considered problem. The comparative simulation results are reported in Table 19. From Table 19, it may be easily observed that the proposed QOCALO yields the minimum fabrication cost as compared to other reported algorithms. Hence, it may be concluded that the proposed QOCALO outperforms its original counterpart and other popular swarm and evolution based algorithms in solving this highly constrained engineering design optimization problem.

Table 15 Comparative results of optimal DG placement for 69-node system offered by different algorithms

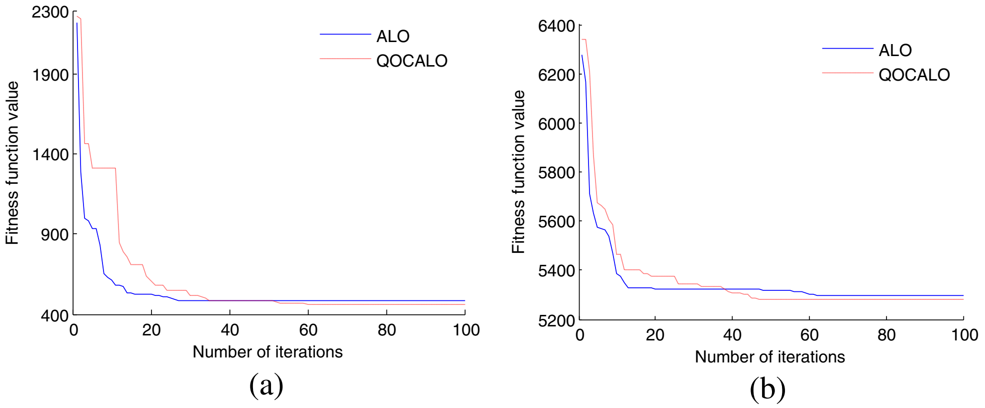
Indices	CABC [73]	ALO [Studied]	MFO [Studied]	MVO [Studied]	WOA [Studied]	CSA [Studied]	QOCALO [Proposed]
ILP	0.3186	0.3531	0.3408	0.3513	0.3464	0.3403	0.3323
ILQ	0.3515	0.3768	0.3658	0.3768	0.3706	0.3666	0.3604
IVD	0.0182	0.0261	0.0174	0.0265	0.0254	0.0192	0.0136
IC	0.8000	0.5761	0.5671	0.6013	0.5667	0.5738	0.5695
$VSI_{\min}$	0.9289	0.8995	0.9323	0.8845	0.9023	0.9253	0.9455
P_{LDG} (KW)	71.69	79.4448	76.6676	79.0274	77.9270	76.5546	74.7632
Q_{LDG} (KW)	35.90	38.4836	37.3504	38.4827	37.8461	37.4366	36.8055
MOF	0.4746	0.4392	0.4258	0.4468	0.4331	0.4285	0.4205
Optimal	17 (0.5627)	51 (0.9499)	61 (1.1845)	59 (0.6463)	8 (1.2000)	60 (0.8381)	12 (1.0688)
location-size	61 (1.2000)	61 (0.9204)	62 (0.6084)	63 (0.9824)	61 (1.2000)	61 (1.0486)	61 (1.2000)
(MW) pair	64 (0.5733)	62 (0.9877)	67 (1.1965)	69 (0.9075)	62 (0.5993)	67 (0.9994)	63 (0.6785)

Table 16 Details of simulated test cases for CM problem

Test system	Cases	Type of contingency	Congested lines	Line power flow (MW)		Line flow limit (MW)
				Before CM	After CM	
Modified IEEE 30-bus	1A	Outage of line 1-2	1-7	147.463	129.531	130
			7-8	136.292	120.366	130
	1B	Outage of line 1-7 and load at all busses increased by 50%	1-2	310.917	130	130
			2-8	97.353	63.335	65
			2-9	103.524	65	65
Modified IEEE 57-bus	2A	Reducing the capacity of the lines 5-6 and 6-12 from 200 MW to 175 MW and from 50 MW to 35 MW, respectively	5-6	195.971	174.904	175
			6-12	49.351	34.991	35
	2B	Reducing the capacity of the line 2-3 from 85 MW to 20 MW	2-3	37.048	19.902	20

Table 17 Comparative results of CM problem for modified IEEE 30-bus system offered by different algorithms

Parameters	BA [75]	CBA [75]	FPA [75]	FA [75]	ALO [75]	QOCALO [Proposed]
Case 1A						
Total congestion cost ($/h)	496.78	482.023	519.62	511.873	480.0438	462.1199
ΔP_{G1}	−9.0100	8.694	9.1278	8.778	9.0880	−8.1880
ΔP_{G2}	13.969	13.917	14.1400	15.000	15.0668	14.9757
ΔP_{G3}	0.102	0.016	0.206	0.106	0.0000	−0.0002
ΔP_{G4}	0.301	0.109	0.0188	0.065	0.0001	0.0014
ΔP_{G5}	0.523	0.349	0.189	0.173	0.0002	0.0031
ΔP_{G6}	0.035	0.317	1.013	0.618	0.0001	0.0011
Total generation rescheduled (MW)	23.943	23.402	24.703	24.742	24.1552	23.1695
Case 1B						
Total congestion cost ($/h)	5333.72	5302.4	5320.8	5304.4	5296.75	5281.1429
ΔP_{G1}	8.582	8.361	8.589	8.579	8.588	-8.5870
ΔP_{G2}	67.7200	76.402	74.024	75.995	76.400	75.4000
ΔP_{G3}	11.2700	0.071	0.0000	0.057	0.056	0.0700
ΔP_{G4}	49.4200	50.082	13.5174	42.994	42.844	42.6268
ΔP_{G5}	13.1500	17.227	43.865	23.832	24.571	25.145
ΔP_{G6}	14.4500	15.736	27.8900	16.5144	15.525	15.2695
Total generation rescheduled (MW)	164.6100	168.1500	167.896	167.974	167.9800	167.0983



(a) (b)

Fig. 11 Comparative convergence profiles of fitness function value for **a** case 1A and **b** case 1B pertaining to CM problem

Table 18 Comparative results of CM problem for modified IEEE 57-bus system offered by different algorithms

Parameters	BA [75]	CBA [75]	FPA [75]	FA [75]	ALO [75]	QOCALO [Proposed]
Case 2A						
Total congestion cost ($/h)	6268.6	5982.8	6340.8	6050.1	5896.548	5890.57
ΔP_{G1}	6.3650	26.3670	0.8760	5.6350	34.3670	59.5255
ΔP_{G2}	0.0590	2.8090	0.0000	2.5230	1.6090	6.9993
ΔP_{G3}	68.9020	0.7960	9.2470	0.5090	0.0000	0.0000
ΔP_{G4}	−0.4470	0.1020	−1.3620	0.1070	−1.8670	0.0000
ΔP_{G5}	−52.4000	−40.5550	−52.4790	−39.1510	−42.2540	−44.2541
ΔP_{G6}	−24.8700	−32.5960	−24.5480	−35.1120	−31.5960	−27.545
ΔP_{G7}	0.6360	40.3270	64.3340	62.1930	30.3270	3.2447
Total generation rescheduled (MW)	153.6000	143.5540	152.8490	145.2270	142.0200	141.5686
Case 2B						
Total congestion cost ($/h)	2361	2332	2912.6	2618.1	2317.6	2310.18
ΔP_{G1}	−0.6980	−0.3950	−0.0060	0.3700	−0.1074	0.0000
ΔP_{G2}	−27.8580	−28.0720	−35.6230	−27.5080	−28.2907	−28.0783
ΔP_{G3}	28.4500	28.6460	20.0970	31.6290	28.5930	28.7742
ΔP_{G4}	−0.1840	0.0473	0.0280	0.3300	0.1338	0.1510
ΔP_{G5}	0.6410	−0.1660	1.4290	−2.2540	−0.0503	0.0029
ΔP_{G6}	−0.0530	0.0000	−0.0300	−1.9350	−0.0088	0.0000
ΔP_{G7}	−0.0160	0.1745	13.9650	−0.5100	−0.0218	0.0000
Total generation rescheduled (MW)	57.9020	57.5020	71.1810	64.5390	57.2590	57.0064

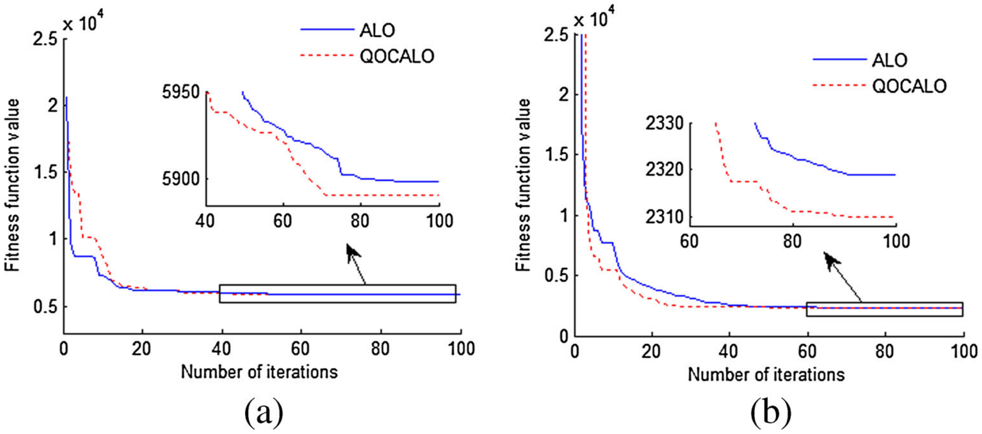


Fig. 12 Comparative convergence profiles of fitness function value for **a** case 2A and **b** case 2B pertaining to CM problem

Table 19 Comparative results of pressure vessel design problem offered by different algorithms

Parameters	MFO [18]	GSA [18]	PSO [77]	GA [78]	DE [79]	ACO [80]	ALO [Studied]	QOCALO [Proposed]
x_1	0.8125	1.1250	0.8125	0.8125	0.8125	0.8125	0.7869	0.7793
x_2	0.4375	0.6250	0.4375	0.4375	0.4375	0.4375	0.3873	0.3836
x_3	42.098445	55.988659	42.091266	42.097398	42.098411	42.103624	40.773169	40.376831
x_4	176.636596	84.4542025	176.746500	176.654050	176.637690	176.572656	197.692834	199.273421
Optimal cost	6059.7143	8538.8359	6061.0777	6059.9463	6059.7340	6059.0888	5981.1338	5884.2655

8 Conclusion and scope of future work

In the present article, a novel QOCALO algorithm is proposed for solving global optimization problems. The proposed algorithm is designed by combining the QOBL based search and the CLS based search strategies. The inclusion of QOBL based search technique improves the quality of the initial solution set and accelerates the search process. It greatly improves the convergence rate of the proposed QOCALO algorithm over its original counterpart. Also, the QOBL diversifies the search direction by exploring powerful regions of the search space which offers better exploration. On the other hand, the CLS guides local search around the global best solution which is the most promising region in the search space. It provides better exploitation and the quality of solution is also improved. It may be observed from the simulation results that the proposed QOCALO algorithm performs much better in optimizing all the three types of mathematical benchmark test functions (*viz.* unimodal, multimodal and composite) compared to its original counterpart. Hence, it may be concluded that a fair balance between exploration and exploitation is maintained in the proposed algorithm which makes it robust. The proposed QOCALO has also been utilized for solving three different real world engineering optimization problems. The simulation results show the superiority of the proposed QOCALO algorithm over the original ALO and the other reported algorithms. Hence, it may be concluded that the proposed algorithm is capable of solving real world engineering optimization problems with unknown search spaces. Thus, the proposed QOCALO may become a promising optimizing tool and may be further utilized to solve different nonlinear optimization problems in different fields of science and engineering studies.

Appendix

The detailed mathematical formulation of the Weierstrass function is provided in Table 20.

Table 20 Detail of Weierstrass function

Function name	Formulation
Weierstrass	$\sum_{i=1}^{d}\left(\sum_{k=0}^{k\max}\left[a^k\cos\left(2\pi b^k\left(x_i+0.5\right)\right)\right]\right) - d\sum_{k=0}^{k\max}\left[a^k\cos\left(2\pi b^k 0.5\right)\right],$ $a=0.5,\ b=3,\ k\max=20$

References

1. Goldberg DE (1989) Genetic algorithms in search, optimization and machine learning. Addison-Wesley, Boston
2. Kennedy J, Eberhart R (1995) Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks. Perth, pp 1942–1948
3. Storn R, Price K (1997) Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11(4):341–359
4. Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J Glob Optim 39:459–471
5. Dorigo M, Birattari M, Stutzle T (2006) Ant colony optimization. IEEE Comput Intell Mag 1:28–39
6. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1:67–82
7. Yang XS (2009) Firefly algorithms for multimodal optimization. In: Watanabe O, Zeugmann T (eds) Stochastic algorithms: foundations and applications. SAGA 2009. Lecture notes in computer science, vol 5792. Springer, Berlin, pp 169–178
8. Kashan AH (2014) League championship algorithm (LCA): an algorithm for global optimization inspired by sport championships. Appl Soft Comput 16:171–200
9. Rashedi E, Nezamabadi-pour H, Saryazdi S (2009) GSA: A gravitational search algorithm. Inf Sci 179:2232–2248
10. Yang XS (2010) A new metaheuristic bat-inspired algorithm. In: Gonzalez JR et al (eds) Nature inspired cooperative strategies for optimization. NISCO 2010. Studies in computational intelligence, vol 284. Springer, Berlin, pp 65–74
11. Gandomi AH, Yang XS, Alavi AH (2013) Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. Eng Comput 29:17–35
12. Rao RV, Savsani VJ, Balic J (2012) Teaching-learning-based optimization algorithm for unconstrained and constrained real-parameter optimization problems. Eng Optim 44:1447–1462
13. Gandomi AH, Alavi AH (2012) Krill herd: a new bio-inspired optimization algorithm. Commun Nonlinear Sci Numer Simul 17:4831–4845
14. Sadollah A, Bahreininejad A, Eskandar H, Hamdi M (2012) Mine blast algorithm for optimization of truss structures with discrete variables. Comput Struct 102-103:49–63
15. Cheng MY, Prayogo D (2014) Symbiotic organisms search: a new metaheuristic optimization algorithm. Comput Struct 139:98–112
16. Mirjalili S (2015) The ant lion optimizer. Adv Eng Softw 83:80–98
17. Zheng YJ (2015) Water wave optimization: a new nature-inspired metaheuristic. Comput Oper Res 55:1–11
18. Mirjalili S (2015) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowledge-Based Syst 89:228–249
19. Ebrahimi A, Khamehchi E (2016) Sperm whale algorithm: an effective metaheuristic algorithm for production optimization problems. J Nat Gas Sci Eng 29:211–222
20. Mirjalili S, Mirjalili SM, Hatamlou A (2016) Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Comput Appl 27:495–513
21. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
22. Askarzadeh A (2016) A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. Comput Struct 169:1–12

23. Gang M, Wei Z, Xiaolin C (2012) A novel particle swarm optimization algorithm based on particle migration. Appl Math Comput 218:6620–6626
24. Zou F, Wang L, Hei X, Chen D, Yang D (2014) Teaching-learning-based optimization with dynamic group strategy for global optimization. Inf Sci 273:112–131
25. Guo L, Wang GG, Gandomi AH, Alavi AH, Duan H (2014) A new improved krill herd algorithm for global numerical optimization. Neurocomputing 138:392–402
26. Jordehi AR (2015) Enhanced leader PSO (ELPSO): a new PSO variant for solving global optimisation problems. Appl Soft Comput 26:401–417
27. Ghasemi M, Ghavidel S, Gitizadeh M, Akbari E (2015) An improved teaching-learning-based optimization algorithm using lévy mutation strategy for non-smooth optimal power flow. Int J Electr Power Energy Syst 65:375–384
28. Kiran MS, Hakli H, Gunduz M, Uguz H (2015) Artificial bee colony algorithm with variable search strategy for continuous optimization. Inf Sci 300:140–157
29. Mirjalili S, Hashim SZM (2010) A new hybrid PSOGSA algorithm for function optimization. In: International conference on computer and information application. Tianjin, pp 374–377
30. Huang J, Gao L, Li X (2015) An effective teaching-learning-based cuckoo search algorithm for parameter optimization problems in structure designing and machining processes. Appl Soft Comput 36:349–356
31. Li Z, Nguyen TT, Chen S, Truong TK (2015) A hybrid algorithm based on particle swarm and chemical reaction optimization for multi-object problems. Appl Soft Comput 35:525–540
32. Mahi M, Baykan OK, Kodaz H (2015) A new hybrid method based on particle swarm optimization, ant colony optimization and 3-opt algorithms for traveling salesman problem. Appl Soft Comput 30:484–490
33. Dubey HM, Pandit M, Panigrahi BK (2016) Ant lion optimization for short-term wind integrated hydrothermal power generation scheduling. Int J Electr Power Energy Syst 83:158–174
34. Dubey HM, Pandit M, Panigrahi BK (2016) Hydro-thermal-wind scheduling employing novel ant lion optimization technique with composite ranking index. Renew Energy 99:18–34
35. Raju M, Saikia LC, Sinha N (2016) Automatic generation control of a multi-area system using ant lion optimizer algorithm based PID plus second order derivative controller. Int J Electr Power Energy Syst 80:52–63
36. Nair SS, Rana KPS, Kumar V, Chawla A (2017) Efficient modeling of linear discrete filters using ant lion optimizer. Circuits Syst Signal Process 36:1535–1568
37. Tizhoosh HR (2006) Opposition-based learning: a new scheme for machine intelligence. In: International conference on computational intelligence for modeling, control and automation 2005 and international conference on intelligent agents. Web technologies and internet commerce. Vienna, pp 695–701
38. Tizhoosh HR (2005) Reinforcement learning based on actions and opposite actions. In: ICGST conference on artificial intelligence and machine learning. Cairo, pp 94–98
39. Rahnamayan S, Tizhoosh HR, Salama MMA (2007) Quasi-oppositional differential evolution. In: Proceedings of the IEEE congress on evolutionary computation (CEC'2007). Singapore, pp 2229–2236
40. Shiva CK, Shankar G, Mukherjee V (2015) Automatic generation control of power system using a novel quasi-oppositional harmony search algorithm. Int J Electr Power Energy Syst 73:787–804
41. Sultana S, Roy PK (2014) Multi-objective quasi-oppositional teaching learning based optimization for optimal location of distributed generator in radial distribution systems. Int J Electr Power Energy Syst 63:534–545
42. Basu M (2016) Quasi-oppositional group search optimization for multi-area dynamic economic dispatch. Int J Electr Power Energy Syst 78:356–367
43. Liu B, Wang L, Jin YH, Tang F, Huang DX (2005) Improved particle swarm optimization combined with chaos. Chaos, Solitons Fractals 25:1261–1271
44. Xiang T, Liao X, Wong K (2007) An improved particle swarm optimization algorithm combined with piecewise linear chaotic map. Appl Math Comput 190:1637–1645
45. Hefny HA, Azab SS (2010) Chaotic particle swarm optimization. In: International conference on informatics and systems. Cairo, pp 1–8
46. Xia X (2012) Particle swarm optimization method based on chaotic local search and roulette wheel mechanism. Phys Procedia 24:269–275
47. Xu W, Geng Z, Zhu Q, Gu X (2013) A piecewise linear chaotic map and sequential quadratic programming based robust hybrid particle swarm optimization. Inf Sci 218:85–102
48. Turgut OE (2016) Hybrid chaotic quantum behaved particle swarm optimization algorithm for thermal design of plate fin heat exchangers. Appl Math Model 40:50–69
49. Li P, Xu D, Zhou Z, Lee W, Zhao B (2016) Stochastic optimal operation of microgrid based on chaotic binary particle swarm optimization. IEEE Trans Smart Grid 7:66–73
50. Bharti KK, Singh PK (2016) Opposition chaotic fitness mutation based adaptive inertia weight BPSO for feature selection in text clustering. Appl Soft Comput 43:20–34
51. Jia D, Zheng G, Khan MK (2011) An effective memetic differential evolution algorithm based on chaotic local search. Inf Sci 181:3175–3187
52. He Y, Xu Q, Yang S, Han A, Yang L (2014) A novel chaotic differential evolution algorithm for short-term cascaded hydroelectric system scheduling. Int J Electr Power Energy Syst 61:455–462
53. Zhang J, Lin S, Qiu W (2015) A modified chaotic differential evolution algorithm for short-term optimal hydrothermal scheduling. Int J Electr Power Energy Syst 65:159–168
54. Lu P, Zhou J, Zhang H, Zhang R, Wang C (2014) Chaotic differential bee colony optimization algorithm for dynamic economic dispatch problem with valve-point effects. Int J Electr Power Energy Syst 62:130–143
55. Alatas B (2010) Chaotic bee colony algorithms for global numerical optimization. Expert Syst Appl 37:5682–5687
56. Bharti KK, Singh PK (2016) Chaotic gradient artificial bee colony for text clustering. Soft Comput 20:1113–1126
57. Pan QK, Wang L, Gao L (2011) A chaotic harmony search algorithm for the flow shop scheduling problem with limited buffers. Appl Soft Comput 11:5270–5280
58. He X, Rao Y, Huang J (2016) A novel algorithm for economic load dispatch of power systems. Neurocomputing 171:1454–1461
59. Saha S, Mukherjee V (2016) Optimal placement and sizing of DGs in RDS using chaos embedded SOS algorithm. IET Gener Transm Distrib 10:3671–3680
60. Saha S, Mukherjee V (2017) A novel chaos-integrated symbiotic organisms search algorithm for global optimization. Soft Comput. https://doi.org/10.1007/s00500-017-2597-4
61. Kapitaniak T (1995) Continuous control and synchronization in chaotic systems. Chaos, Solitons Fractals 6:237–244
62. Tavazoei MS, Haeri M (2007) Comparison of different one-dimensional maps as chaotic search pattern in chaos optimization algorithms. Appl Math Comput 187:1076–1085
63. Alatas B, Akin E, Ozer AB (2009) Chaos embedded particle swarm optimization algorithms. Chaos, Solitons Fractals 40:1715–1734
64. Popović DH, Greatbanks JA, Begović M, Pregelj A (2005) Placement of distributed generators and reclosers for distribution

network security and reliability. Int J Electr Power Energy Syst 27:398–408

65. Borges CLT, Falcão DM (2006) Optimal distributed generation allocation for reliability, losses, and voltage improvement. Int J Electr Power Energy Syst 28:413–420
66. Singh D, Singh D, Verma KS (2009) Multiobjective optimization for DG planning with load models. IEEE Trans Power Syst 24:427–436
67. Nafar M (2012) PSO Based optimal placement of DGs in distribution systems considering voltage stability and short circuit level improvement. J Basic Appl Sci Res 2:703–709
68. Ganguly S, Sahoo NH, Das D (2013) Multi-objective particle swarm optimization based on fuzzy-pareto-dominance for possibilistic planning of electrical distribution systems incorporating distributed generation. Fuzzy Sets Syst 213:47–73
69. Nayak MR, Dash SK, Rout PK (2012) Optimal placement and sizing of distributed generation in radial distribution system using differential evolution algorithm. In: Panigrahi et al (eds) Swarm evolutionary, and memetic computing. SEMCCO 2012. Lecture notes in computer science. Springer, Berlin, pp 133–142
70. Falaghi H, Haghifam MR (2007) ACO based algorithm for distributed generation sources allocation and sizing in distribution systems. In: Proceedings of IEEE Lausanne POWERTECH. Lausanne, pp 555–560
71. Abu-Mouti FS, El-Hawary ME (2011) Optimal distributed generation allocation and sizing in distribution systems via artificial bee colony algorithm. IEEE Trans Power Deliv 26:2090–2101
72. Moradi MH, Abedini M (2012) A combination of genetic algorithm and particle swarm optimization for optimal DG location and sizing in distribution systems. Int J Electr Power Energy Syst 34:66–74
73. Mohandas N, Balamurugan R, Lakshminarasimman L (2015) Optimal location and sizing of real power DG units to improve the voltage stability in the distribution system using ABC algorithm united with chaos. Int J Electr Power Energy Syst 66:41–52
74. Chakravorty M, Das D (2001) Voltage stability analysis of radial distribution networks. Int J Electr Power Energy Syst 23:129–135
75. Verma S, Mukherjee V (2016) Optimal real power rescheduling of generators for congestion management using a novel ant lion optimizer. IET Gener Transm Distrib 10:2548–2561
76. Balaraman S, Kamaraj N (2011) Transmission congestion management using particle swarm optimization. J Electr Syst 7:54–70
77. He Q, Wang L (2007) An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Eng Appl Artif Intell 20:89–99
78. Coello Coello CA, Mezura Montes E (2002) Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. Adv Eng Inform 16:193–203
79. Li L, Huang Z, Liu F, Wu Q (2007) A heuristic particle swarm optimizer for optimization of pin connected structures. Comput Struct 85:340–349
80. Kaveh A, Talatahari S (2010) An improved ant colony optimization for constrained engineering design problems. Eng Comput 27:155–182

Subhodip Saha was born in 1990 at Kalna, Burdwan, West Bengal, India. He received his B. Tech degree in electrical engineering from Haldia Institute of Technology, Haldia, West Bengal, India in 2012 and M. Tech degree in power system engineering from Indian School of Mines, Dhanbad, Jharkhand, India in 2015. He is currently working toward the Ph.D. degree in electrical engineering at Indian Institute of Technology (Indian School of Mines), Dhanbad, Jharkhand, India. His research interest includes distributed generation, evolutionary and swarm intelligence based algorithms and their application in solving different power system optimization problems. He will be available at subhodip_saha@ymail.com.

V. Mukherjee was born in 1970 at Raina, Burdwan, West Bengal, India. He received his graduation in electrical engineering and post graduation in power system from B.E. College, Shibpur, Howrah, India and B.E. College (Deemed University), Shibpur, Howrah, India, respectively. He received his Ph.D. degree from NIT, Durgapur, India. Presently, he is an associate professor in the department of electrical engineering, Indian Institute of Technology (Indian School of Mines), Dhanbad, Jharkhand, India. His research interest includes application of soft computing intelligence to various fields of power systems. He will be available at vivek_agamani@yahoo.com.

Dr. Mukherjee is a member of The Institution of Engineers (India).