Journal of Ambient Intelligence and Humanized Computing (2023) 14:11685–11713
https://doi.org/10.1007/s12652-022-03731-1

ORIGINAL RESEARCH

Economic load dispatch using memetic sine cosine algorithm

Mohammed Azmi Al-Betar[1,2] · Mohammed A. Awadallah[3,4] · Raed Abu Zitar[5] · Khaled Assaleh[1]

Received: 22 October 2020 / Accepted: 19 January 2022 / Published online: 7 February 2022

Abstract
In this paper, the economic load dispatch (ELD) problem which is an important problem in electrical engineering is tackled using a hybrid sine cosine algorithm (SCA) in a form of memetic technique. ELD is tackled by assigning a set of generation units with a minimum fuel costs to generate predefined load demand with accordance to a set of equality and inequality constraints. SCA is a recent population based optimizer turned towards the optimal solution using a mathematical-based model based on sine and cosine trigonometric functions. As other optimization methods, SCA has main shortcoming in exploitation process when a non-linear constraints problem like ELD is tackled. Therefore, β-hill climbing optimizer, a recent local search algorithm, is hybridized as a new operator in SCA to empower its exploitation capability to tackle ELD. The proposed hybrid algorithm is abbreviated as SCA-βHC which is evaluated using two sets of real-world generation cases: (i) 3-units, two versions of 13-units, and 40-units, with neglected Ramp Rate Limits and Prohibited Operating Zones constraints. (ii) 6-units and 15-units with Ramp Rate Limits and Prohibited Operating Zones constraints. The sensitivity analysis of the control parameters for SCA-βHC is initially studied. The results show that the performance of the SCA-βHC algorithm is increased by tuning its parameters in proper value. The comparative evaluation against several state-of-the-art methods show that the proposed method is able to produce new best results for some tested cases as well as the second-best for others. In a nutshell, hybridizing βHC optimizer as a new operator for SCA is very powerful algorithm for tackling ELD problems.

Keywords Economic load dispatch · Hybrid metaheuristics · Sine cosine algorithm · β-hill climbing optimizer · Power systems

1 Introduction

Economic load dispatch (ELD) is non-travail optimization problem in power system domain. It concerns with satisfying the generation load demand at minimum total fuel cost for a specific period of operation with respect to power-balance equality constraints and power-output inequality constraints with sometimes ramp rate limits and prohibited operating zones. ELD has been formulated as an optimization problem with non-linear, non-convex constrained features that cannot be easily manipulated by calculus-based methods (El-Keib et al. 1994; Panigrahi et al. 2012; Chen and Ding 2015).

✉ Mohammed A. Awadallah
ma.awadallah@alaqsa.edu.ps

Mohammed Azmi Al-Betar
mohbetar@bau.edu.jo

Raed Abu Zitar
raed.zitar@sorbonne.ae

Khaled Assaleh
k.assaleh@ajman.ac.ae

[1] Artificial Intelligence Research Center (AIRC), College of Engineering and Information Technology, Ajman University, Ajman, United Arab Emirates

[2] Department of Information Technology, Al-Huson University College, Al-Balqa Applied University, Irbid, Jordan

[3] Department of Computer Science, Al-Aqsa University, P.O. Box 4051, Gaza, Palestine

[4] Artificial Intelligence Research Center (AIRC), Ajman University, Ajman, United Arab Emirates

[5] Sorbonne University Center of Artificial Intelligence, Sorbonne University-Abu Dhabi, Abu Dhabi, United Arab Emirates

Therefore, the approximation and constraint-handling are the most successful techniques in tackling such problem.

Among the approximation methods used intensively for ELD is metaheuristic-based approaches (Al-Betar et al. 2018b). Metaheuristics are general optimization model can be adapted for a wide range of optimization problems. Metaheuristics are divided into two main categories due to the number of initial solutions: population-based algorithms and local search-based algorithms (Blum and Roli 2003). Population-based algorithms are very powerful in exploration and weak in exploitation. Oppositely, local search based methods are very powerful in exploitation and weak in exploration (Al-Betar et al. 2014). In order to complement the advantages of both population and local search methods to provide a suitable trad-off between exploitation and exploration process, the hybridized mechanism is normally suggested (Blum et al. 2011). This type of hybridization is called memetic algorithm (MA). In MA, the cultural selection mechanism in local search algorithms represented as meme is combined with the natural selection principle built up in the population-based framework represented as gene (Ong and Keane 2004). The meme behaves like a local/individual improvement agent in cultural evolution. Both gene and meme reflect the natural and cultural selection for generality (gene) and problem-specificity (meme) (Ong et al. 2006; Al-Betar et al. 2014).

The recent metaheuristic population-based algorithm called Sine Cosine Algorithm (SCA) has been established to utilize the Sine and cosine trigonometric functions features for updating the solution (Mirjalili 2016b). It has a distinguished features over other population-based algorithms: Firstly, it has no derived values in the initial search, therefore, naive optimization users can use it easily. Secondly, it has intelligent operators that can iterativly accumulate a problem-specific knowledge making use of survival-of-the-fittest principle through Sine and Cosine trigonometric functions. In a nutshell, it is simple in concepts, robust in its convergence behaviour, easy to use for any problem, sound and complete. Therefore, SCA has been widely deployed for many optimization problems like optimal power flow (Attia et al. 2018), feature selection (Sindhu et al. 2017), short-term hydrothermal scheduling (Das et al. 2018), object tracking (Nenavath et al. 2018), image segmentation (Oliva et al. 2018), unit commitment problem (Reddy et al. 2018), automatic voltage regulator system (Hekimoğlu 2019), loading margin stability in power system (Mahdad and Srairi 2018), the optimal design of a shell and tube evaporator (Turgut 2017), detection of pathological brain (Nayak et al. 2018), designing bend photonic crystal waveguides (Mirjalili et al. 2020), wireless sensor nodes localiser (Hamouda and Abohamama 2020), abrupt motion tracking (Zhang et al. 2020), optimal distributed generators allocation in radial distribution systems (Raut and Mishra 2020), nonlinear bilevel programming problems (Abo-Elnaga and El-Shorbagy 2020), traveling salesman problem (Tawhid and Savsani 2019), truss structures (Gholizadeh and Sojoudizadeh 2019), train multilayer perceptrons (Gupta and Deep 2020), parameter optimization of support vector regression (Li et al. 2018), text categorization (Belazzoug et al. 2019), and etc.

SCA likes other population-based algorithm when dealing with real-world optimization problems with combinatorial nature and constrained features, it requires careful amendments or hybridization with strong component from other heuristic algorithm to improve its performance (Gupta and Deep 2019a). Therefore, several modified versions of SCA such as: (i) hybridized with local-based algorithms (Chen et al. 2019; Liu et al. 2019; Tawhid and Savsani 2019), (ii) hybridized with other population-based algorithms (Nenavath and Jatoth 2018; Chegini et al. 2018), (iii) hybridized with other operators borrowed from other optimization algorithms (Liu et al. 2019; Gupta and Deep 2019b), and (iv) modified its operators to be in line with the problem nature (Gupta and Deep 2019a; Sindhu et al. 2017; Long et al. 2019; Chen et al. 2020). In particular, the SCA is integrated with gaussian local search and random mutation in order to enhance the diversity of the population (Liu et al. 2019). In Long et al. (2019), a new nonlinear conversion parameter based on gaussian function is introduced to strike a right balance between exploration and exploitation, and a modified position-updating equation is proposed to accelerate convergence and avoid the problem of local optima. In another study, the SCA is combined with Nelder-Mead simplex concept and the opposition-based learning scheme (Chen et al. 2019). In this method, the Nelder-Mead simplex concept is used to enhance the exploitation, while opposition-based learning scheme is used to improve the diversification of the population.

As aforementioned, to improve the performance of the SCA to deals with ELD problem with constrained features, a successful local search-based algorithm can be hybridized with the SCA optimization framework as a new operator, thus the meme and gene concepts can be utilized. The recent local search-based algorithm introduced by Al-Betar (2017) is called β-hill climbing (βHC) optimizer.

In principle, βHC optimizer is a local search-based algorithm initiated with a single solution. Iteratively, this solution is moved to its neighbouring solution using two operators: $\mathcal{N}$-operator for neighbouring search and β-operator for exploration. The greedy selection mechanism is used to compete between the current and neighbouring solution. This process is repeated until the stop condition is met. The main advantages of this algorithm is its simplicity and easy-to-use. Through its maneuver behaviour of navigating the search space and escaping the local optima, βHC optimizer has been successfully tailored and hybridized with other population-based algorithms for many optimization problems such as examination timetabling (Al-Betar 2021), feature selection (Ghosh et al. 2020), ECG and EEG signal denoising (Alyasseri et al. 2018, 2017), Generating Substitution-Boxes (Alzaidi et al. 2018), gene selection (Alomari et al. 2018b), economic load dispatch problem (Al-Betar et al. 2018a), multiple-reservoir scheduling (Alsukni et al. 2019), Sudoku game (Al-Betar et al. 2017), text document clustering (Abualigah et al. 2017a, b), classification problems (Alomari et al. 2018a; Alweshah et al. 2020), and mathematical optimization functions (Abed-alguni and Alkhateeb 2020). The main motivation behind using βHC as meme is its feature in digging deeply in the search space regions to which the population-based algorithm is reached. Through $\mathcal{N}$-operator, the βHC exploit the features of the current solution and move to its neighbouring solution. On the other side, through the β-operator, the βHC is able to escape the local minima using uniform mutation strategy. The main limitation of βHC is the parameter tuning where the study used βHC should study the proper values of its control parameters.

In this paper, the economic load dispatch (ELD) problem is tackled using hybridizing SCA as a gene agents with βHC as a meme agent. This is to empower the performance of SCA in tackling ELD problem. The proposed hybrid algorithm is abbreviated as SCA-βHC which is evaluated using two sets of real-world generation cases: (i) 3-units, two versions of 13-units, and 40-units, with neglected Ramp Rate Limits and Prohibited Operating Zones constraints. (ii) 6-units and 15-units with Ramp Rate Limits and Prohibited Operating Zones constraints. The sensitivity analysis of the control parameters for SCA-βHC is initially studied to show their impact on the proposed SCA-βHC algorithm. The comparative evaluation reveals that the SCA-βHC is able to efficiently tackle ELD problem. In conclusion, SCA-βHC is a successful algorithm for power systems such as ELD.

The remaining parts of this paper is organized as follow: the ELD problem is defined and formulated in terms of optimization methods in Sect. 2. The proposed methodology is presented in Sect. 3. Thereafter, the results and discussion with comparative evaluations for all real world test cases is provided in Sect. 4. Finally, the conclusions with suggested future directions are given in Sect. 5.

2 Foundations to economic load dispatch problem

The Economic load dispatch (ELD) problem is modeled in optimization context to generate a constant load demand by means of assigning each generation unit with minimum fuel costs with respect to equality and inequality constraints. In this section, the mathematical formulations of ELD is provided. Thereafter, the previous ELD methods are discussed.

2.1 ELD formulations

The ELD can be formulated as a challenging optimization problem that could be a valuable target for many optimization techniques. As any optimization problem, the objective function for such problem need to be formulated to reflect the exact mathematical model that could be utilized by the optimization techniques.

The ELD problem under consideration here has N generation units, each unit has a range of fuel cost that it can use. The objective here is to assign each unit with the proper amount of allowed fuel cost that results in a total minimum of fuel cost for all units. The optimization has to satisfy a set of constraints such that the optimum solutions will always be feasible. In the sections below, the mathematical formulations of the main optimization terms for ELD problem are provided.

2.2 Solution representation

The representation of the solution for the ELD is formulated using vector space. It is represented as a vector where $\boldsymbol{P} = (P_1, P_2, \dots, P_N)$, where the P_i is the generation unit number i which will take a value range $P_i \in [P_i^{min}, P_i^{max}]$. Please note that the value of P_i^{min} refers to the lower bound while P_i^{max} refers to the upper bound.

2.3 Objective function

The objective function that is used to evaluate ELD solution $\boldsymbol{P}$ in terms of fuel costs is formulated in Eq. (1).

$$\min F(\boldsymbol{P}) = \sum_{i=1}^{N} F_i(P_i). \tag{1}$$

where $F(\boldsymbol{P})$ is the total fuel cost for N units and F_i is a function used to calculate the fuel cost for unit P_i. The calculation of fuel cost $F_i(P_i)$ for unit P_i is given in Eq. (2)

$$F_i(P_i) = \lfloor a_i P_i^2 + b_i P_i + c_i + |e_i sin(f_i(P_i^{(min)} - P_i))| \rfloor. \tag{2}$$

The smooth coefficients (a_i, b_i, c_i) and the non-smooth coefficients (e_i, f_i) related to the unit i. The sinusoidal function shows the effect of the valve-point load obtained waves in the trend of the heat-rate. That revealed more challenges to avoid being stuck in local optimum solutions (Alsumait et al. 2010).

2.4 Constraints

In each ELD solution, two main constraints and two extended constraints should be fulfilled to ensure the feasibility of the ELD solution. In the next sections, those constraints are mathematically formulated.

2.4.1 Equality constraints

The first constraint is the balance power between total system generation $\sum_{i=1}^{N} P_i$ and the overall system loads P_D with power losses P_L as mathematically shown in Eq. (3).

$$\sum_{i=1}^{N} P_i - P_D - P_L = 0 \tag{3}$$

where the transmission losses of the system is formulated in terms of quadratic function of the power generating outputs as shown in Eq. (4)

$$P_L = \sum_{k=1}^{N}\sum_{i=1}^{N} P_k B_{ki} P_i + \sum_{k=1}^{N} B_{0k} P_k + B_{00}. \tag{4}$$

Where B_{0k} and B_{00} denote the i^{th} elements of the loss coefficient vector, while B_{ki} represent the loss coefficient square of ki^{th} element. In the literature (Cai et al. 2012b; Alsumait et al. 2010; Cai et al. 2012a; Lin et al. 2011; Tsai et al. 2011), the transmission losses are simplified by considering $P_L = 0$.

2.4.2 Inequality constraints

Equality constraint is the second of those constraints that specify the lower and upper value ranges $(P_i \in [(P_i^{(min)}), (P_i^{(max)})])$ of the generation unit P_i as calculated in Eq. (5).

$$P_i^{(min)} \leq P_i \leq P_i^{(max)}, i \in N_s \tag{5}$$

2.4.3 Ramp rate limits

The inequality constraints are sometimes restricted by the ramp rate limit. It occurs between two periods of operations. The process of the unit generation will be either increased or decreased. This is done according to upper and lower ramp rate periods formalized as follows:

Increasing power generation is constrained as:

$$P_i - P_i^0 \leq UR_i$$

Decreasing power generation is constrained as:

$$P_i^0 - P_i \leq DR_i$$

Note that, P_i^0 is the previous time-period of the power generation for unit i. UR_i and DR_i refers to the upper and lower ramp rate limits, respectively. Accordingly, the inequality constraints are adjusted by ramp rate limits as shown below:

$$\max(P_i^{(min)}, UR_i - P_i) \leq P_i \leq \min(P_i^{(max)}, P_i^0 - DR_i)$$

2.4.4 Prohibited operating zones

Due to physical limitations of machine components or instability due to vibrations in the bearing of the shaft, the power generation units could be restricted by prohibited operating zones. This will result in cost fluctuations that can be detected. Discontinuities will consequently appear in cost curves corresponding to the prohibited operating zones. The quest to avoid operating in those regions will result on economizing the production. Formally, for a generating unit i,

$$P_i \leq \check{P}^{pz} \text{ and } P_i \geq \hat{P}^{pz}$$

Note that $\check{P}^{pz}$ and $\hat{P}^{pz}$ represent the lower and upper limits of a given prohibited zone for generating unit i, respectively.

Finally, In case the Prohibited Operating Zones constraint are considered in the ELD problem, the fuel costs is formulated in Eq. (6).

$$\min F(\boldsymbol{P}) = \sum_{i=1}^{N} F_i(P_i) + \alpha \left[\sum_{j=1}^{m} P_i(violation)_j\right]^2. \tag{6}$$

Where α is the penalty factor for operating zone constraint, and p_i (violation) is the indicator of falling into prohibited operating zone (Safari and Shayeghi 2011).

2.5 Previous ELD methods

Several metaheuristic-based algorithms have been recently introduced. Those algorithms in many cases mimic the natural behaviors of living creatures such as humans, animals, and plants. The metaheuristc-based algorithms are general optimization algorithm. They span the whole search space and use intelligent operators to tweak workable solutions. They use carefully-selected parameter values to come up with optimal solutions (Blum and Roli 2003). Examples of recent metaheuristic-based algorithms are Seeker optimization algorithm (Dai et al. 2006), Dragonfly Algorithm (Mirjalili 2016a), Coronavirus herd immunity optimizer (Al-Betar et al. 2020a), Multi-verse Optimizer (Mirjalili et al. 2016), Ant Lion Optimizer (Mirjalili 2015a), Harris Hawks optimization (Heidari et al. 2019), Moth-Flame Optimization Algorithm (Mirjalili 2015b), Crow search algorithm (Askarzadeh 2016), sine cosine algorithm (Mirjalili 2016b), Grasshopper Optimization Algorithm (Saremi et al. 2017), Whale Optimization Algorithm (Mirjalili and Lewis 2016), and many others as reported in (Fausto et al. 2020). They are normally categorized into: Single solution-based methods and Population-based methods (BoussaïD et al. 2013).

Solutions that depend on starting with a single state, begin with one stochastic solution and keep iteratively updating and modifying this solution moving from one state to another until a steady state called "local minima" is reached. These methods usually span the search space deep in the form of a trajectory that resembles depth-first method, but it lacks wide search mechanism. Some of those solutions that solved the ELD include simulated annealing (Wong and Fung 1993), GRASP (Neto et al. 2017), β-Hill climbing (Al-Betar et al. 2018a), and tabu search (Lin et al. 2002).

On the other hand, population-based methods start with population of random solutions (individuals) that work in parallel to scan all possible solutions. They exchange bits and pieces of information and use a level of randomness to be more creative and exploitative. Those techniques has the ability to scan a wide search space regions at the same time but it cannot rigorously navigate the scanned regions to find their local optima (Blum and Roli 2003). Several population-based methods are used to solve ELD such as Particle Swarm Optimization (Qin et al. 2017; Jain et al. 2018), Ant colony Optimizer (Pothiya et al. 2010), Cuckoo Search Algorithm (Basu and Chowdhury 2013), Firefly Algorithm (Sinha et al. 2003), Krill Herd Algorithm (Mandal et al. 2014), Artificial Bee Colony (Hemamalini and Simon 2010), Flower Pollination Algorithm (Shilaja and Ravi 2017), Social Spider Algorithm (James and Li 2016), Crow Search Algorithm (Mohammadi and Abdi 2018), Artificial Algae Algorithm (Kumar and Dhillon 2018), Moth Flame Algorithm (Tripati et al. 2018; Elsakaan et al. 2018), Fireworks Algorithm (Jadoun et al. 2018; Pandey et al. 2018), Harmony Search Algorithm (Pandi et al. 2011), Ant Lion Optimizer (Kamboj et al. 2017), Dragonfly Algorithm (Suresh and Sreejith 2017), Genetic Algorithm (Subbaraj et al. 2011), Environmental Adaptation Method (Sharma et al. 2017), and Biogeography-based Algorithm (Bhattacharya and Chattopadhyay 2010a).

In order to improvise an efficient and effective search mechanism, we need to hybridize the two types of search methods; the single solution-based methods and the population-based search methods. When we hybridize, we expect to harvest the merits of both methods (Blum and Roli 2003). We expect to have an algorithm that performs deep search in addition to wide search all accompanied with intelligent information exchange operators and survivability for the fittest selection mechanism. The most successful hybrid techniques used for solving the ELD are as follows:

1. The Sequential Quadratic Programming is hybridized with Differential Evolution (Elaiw et al. 2013).
2. The Sequential Quadratic Programming is hybridized with the Particle Swarm Optimization (Victoire and Jeyakumar 2004; Elaiw et al. 2013).
3. The Genetic Algorithm is integrated with Shuffled Frog Leaping Algorithm (Roy et al. 2013).
4. The Quadratic Programming is combined with chaotic Differential Evolution (Coelho and Mariani 2006).
5. The Bacterial Foraging Approach is hybridized with Genetic Algorithm (Elattar 2015).
6. The Particle Swarm Optimization is hybridized with Differential Evolution (Parouha and Das 2016).
7. The GRASP algorithm is combined with Differential Evolution (Neto et al. 2017).
8. The local search is hybridized with Artificial Bee Colony algorithm (Özyön and Aydin 2013).
9. Differential Evolution is hybridized with Chemical Reaction Optimization (Roy et al. 2014).
10. Biogeography-based Optimization is hybridized with Differential Evolution (Bhattacharya and Chattopadhyay 2010c).

A recent study (Le Yang 2020) have improvised an algorithm based on the selections of hyper-heuristics (SHHA) to solve the problem of dynamic economic emission dispatch (DEED). A mathematical model of the multi-objective dynamic economic and environmental load distribution problem including wind energy, solar power, and energy storage. The authors provided a new way to solve the coupled spatial-temporal scheduling problems in the power systems. They used a non dominated sorting strategy to guide the screen of parent individuals. They tried to transform infeasible solutions toward feasible ones to enhance the quality of solutions and convergence.

In another work (Pawan Preet Singh 2017), the ELD was tackled using SCA algorithm and Moth Flame Optimizer (MFO). Combinatorial analysis were performed among those techniques. The SCA was used for constrained optimization based on the concept of correlation mathematical model of sine and cosine functions. The MFO is a heuristic algorithm that utilizes the concept that the moth uses to converge toward the light. Both algorithms were used in solving benchmark problems of ELD. Combinatorial results show that the performance of the MFO is better than that of the SCA in view of various parameters used.

A modified sine cosine algorithm (MSCA), in Rizk M. Rizk-Allah (2020), is also used for ELD problem. In this method, a population of random solutions is initialized in the search domain to optimize both the combined environmental and economic objectives. Moreover, the diversity of MSCA is controlled using an opposition strategy. The modified version of SCA is used as a multi-objective solution with Pareto solutions. The proposed MSCA is evaluated against two real-world ELD cases with emission issues and non-smooth fuel cost. Results of MSCA showed more correctness and effectiveness.

In Babar et al. (2020), a sine cosine algorithm was used to solve convex and non-convex ELD problem. They used Differential Evolution (DE) mutation and crossover operators to equip the SCA. The DE algorithm induces diversity in the operations of SCA making it avoid local minima and premature convergence. To ensure optimum tracking results, the proposed algorithm is refined by SQP (Sequential Quadratic Programming). The results with those techniques showed improvements in convergence time and fuel cost as compared to the techniques reported in the literature.

3 Hybrid SCA for ELD

In this section, the proposed hybrid SCA with βHC optimizer (SCA-βHC) for ELD problem is presented. The SCA is the recent population-based algorithm initiated with a population of random solutions. These solutions are iterativly improved using Sine Cosine trigonometric functions. The SCA-βHC has five main steps which is visualized as shown in the flowchart in Fig. 1. In the procedural context, the pseudo-code of SCA-βHC is given in Algorithm 1.

Step 1: Initialize the SCA-βHC and ELD In order to solve any optimization problem using optimization method as SCA, the modeling in terms of optimization problem should be designed such as objective function and solution representations. In ELD, the objective function are modeled in Eq. 1. The ELD solution is represented as a vector $\boldsymbol{P} = (P_1, P_2, \dots, P_N)$ where N is the number of generating units.

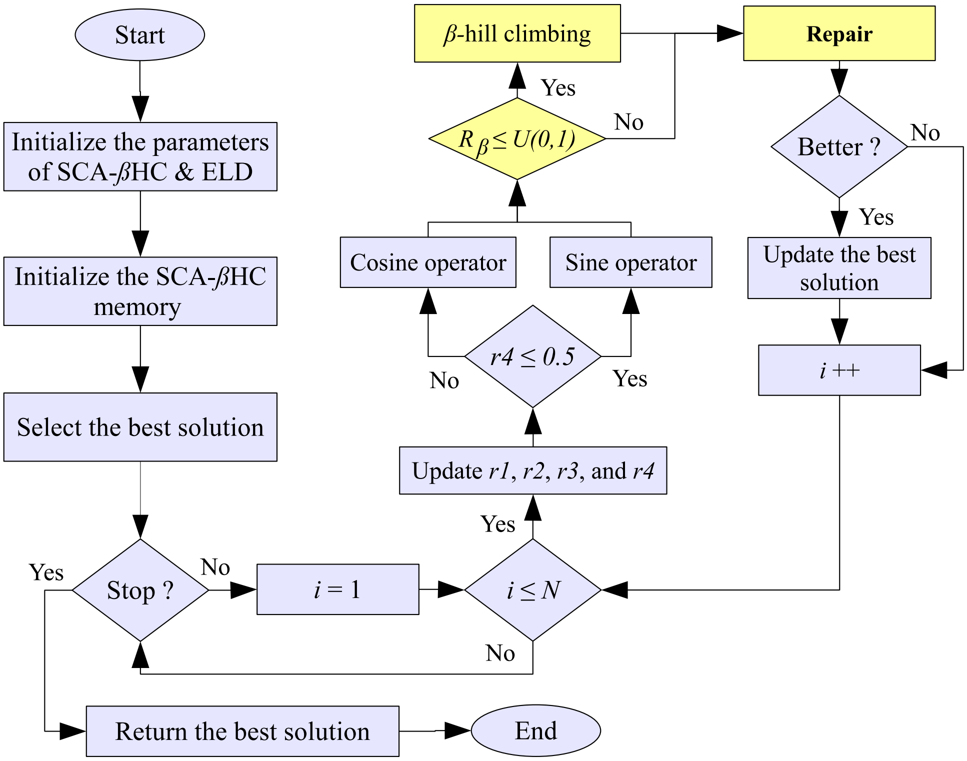
In the SCA, there are no control parameters. However, SCA has two algorithmic parameters such as Maximum iteration numbers (It_{Max}) and population size (M). For βHC optimizer, there are two control parameters:β and bw. β is responsible for diversifying the search while bw defines the distance bandwidth between the current solution and neighbouring solution. Finally, the parameter R_β) determines the rate of invoking βHC optimizer inside SCA.

Step 2: Initialize SCA-βHC memory The SCA-βHC memory (SBM) is a matrix of size $N \times M$ where N is the number of generation units while M is the population size as formulated is Eq. (7). The objective function value of any solution is calculated as formulated in Eq. (1). The solutions in *SBM* are then sorted in ascending order (i.e., $F(\boldsymbol{P}^1) \leq F(\boldsymbol{P}^2) \leq \dots \leq F(\boldsymbol{P}^M)$).

Algorithm 1 SCA-βHC pseudo-code

```
{———- Step 1: Initialize the SCA-$\beta$HC parameters ———-}
Initialize the parameters ($N$, $M$, $bw$,$\beta$ and $It_{Max}$).
{———- Step 2: Generate the initial search agents ———-}
$P_j^i = P_j^{(min)} + (P_j^{(max)} - P_j^{(min)}) \times U(0,1), \quad \forall j = 1, 2, \dots, N$, and $\forall i = 1, 2, \dots, M$.
Calculate the fitness of each solution $P_j^i$
Find the best search agent $P^{gbest}$
{———- Step 3: Update the position of each search agent ———-}
while ($t \leq It_{Max}$) do
  $r1 = a - t\frac{a}{It_{Max}}$
  for $i$ = 1 to $M$ do
    for $j$ = 1 to $N$ do
      $r_2, r_3, r_4 \in [0, 1]$
      if ($r_4$ <0.5) then
        ${P'}_j^i = P_j^i + r_1 \times \sin(r_2) \times \left|r_3 P^{gbest} - P_j^i\right|$
      else
        ${P'}_j^i = P_j^i + r_1 \times \cos(r_2) \times \left|r_3 P^{gbest} - P_j^i\right|$
      end if
    end for
    Maintaining ($\boldsymbol{P'}^i$)        {See Algorithm 2}
    if ($rnd() < R_\beta$) then
      $\beta$HC($\boldsymbol{P'}^i$)        { See Algorithm 3}
    end if
    if $f(\boldsymbol{P'}^i) < f(\boldsymbol{P}^i)$ then
      $\boldsymbol{P}^i = \boldsymbol{P'}^i$
      $f(\boldsymbol{P}^i) = f(\boldsymbol{P'}^i)$
    end if
    if ($f(\boldsymbol{P'}^i) < f(P^{gbest})$) then
      $P^{gbest} = \boldsymbol{P'}^i$        {Update the best search agent}
    end if
  end for
  $t = t + 1$
end while
Return the best search agent $P^{gbest}$
```

Fig. 1 The flowchart of the proposed SCA-βHC algorithm



$$\mathbf{SBM} = \begin{bmatrix} P_1^1 & P_2^1 & \cdots & P_N^1 \\ P_1^2 & P_2^2 & \cdots & P_N^2 \\ \vdots & \vdots & \cdots & \vdots \\ P_1^M & P_2^M & \cdots & P_N^M \end{bmatrix} \begin{bmatrix} F(\boldsymbol{P}^1) \\ F(\boldsymbol{P}^2) \\ \vdots \\ F(\boldsymbol{P}^M) \end{bmatrix} \tag{7}$$

Step 3: SCA-βHC Evolution Loop This is the evolution part of the proposed SCA-βHC. Four main operators are iterated toward the optimal ELD solution: (i) Sine Operator, (ii) Cosine Operator, (iii) Maintaining Process, and (iv) β hill climbing optimizer. These operators are consecutively executed.

Step 3.1: Sine operator This operator is called Sine operator because it used sine function to update the current ELD solution as shown in Eq. (8).

$$P'^i_j = P^i_j + r_1 \times \sin(r_2) \times \left| r_3 P^{gbest} - P^i_j \right| \tag{8}$$

Step 3.2:Cosine Operator This operator is called Cosine operator because it uses cosine function to update the current ELD solution as shown in Eq. (9).

$$P'^i_j = P^i_j + r_1 \times \cos(r_2) \times \left| r_3 P^{gbest} - P^i_j \right| \tag{9}$$

In accordance with the values r_4 where $r_4 \in [0, 1]$, the value of the current generation unit is either updated based on sine function in Eq. (8) or cosine function in Eq. (9). This can be summarized in Eq. (10)

$$P'^i_j = \begin{cases} P^i_j + r_1 \times \sin(r_2) \times \left| r_3 P^{gbest} - P^i_j \right|, & r_4 < 0.5 \\ P^i_j + r_1 \times \mathrm{Cos}(r_2) \times \left| r_3 P^{gbest} - P^i_j \right|, & r_4 \geq 0.5 \end{cases} \tag{10}$$

where r_1, r_2, r_3 are random numbers. r_1 is the parameter represented the position of the next region which indicates the bandwidth between the solution and the best solution P^{gbest}. r_2 is the parameter which represents how far the movement shall be outwards or towards global-best solution (or the destination). r_3 is the parameter which represents the drawing of the random weight for the destination. This is to de-emphasize ($r_3 < 1$) or emphasize ($r_3 > 1$) the impact of the destination on the bandwidth distance defined.

The success of any optimization algorithm depends on how the exploration and exploitation processes of the search spaces are balanced. Therefore, the range between sine and cosine functions should be adaptively updated using Eq. (11).

$$r_1 = a - t\frac{a}{It_{Max}} \tag{11}$$

Step 3.3: maintaining process The maintaining process is responsible for modifying the infeasible solutions to be feasible. After the Sine and Cosine operators are invoked, the constructed solution may be infeasible. This is because either equality or inequality constraints are violated. In this case, the maintaining process will check the value of each generated unit according to the allowable value range. Furthermore, the power output shall be equal to the power generated by the solution.Therefore, the solution is maintained as shown in Algorithm 2.

Algorithm 2 Maintaining(***P***)

```
for $i = 1, \cdots, N$ do
    $Tot = Tot + P_i$
end for
$dis = Tot - P_D$
while $dis \neq 0$ do
    $tmp = rand(0, N)$
    $i \in (1, b)$ {$i$ is any generating unit}
    if $((P_i \pm tmp) \geq P_i^{(min)}$ and $(P_i \pm tmp) \leq Pi^{(max)})$ then
        $P_i = P_i \pm tmp$
        $dis = dis \pm tmp$
    end if
end while
```

Step 3.4: β-hill climbing optimizer. The β-hill climbing optimizer behaves as a local search operator in SCA. With the range of R_β where $R_\beta \in [0, 1]$, the β-hill climbing is invoked as shown in Algorithm 1. The constructed and maintained solution from Sine and Cosine operators (i.e., $\boldsymbol{P}^i = (P^i_1, P^i_2, \dots, P^i_N)$) that occurs within the range of R_β rate is treated as an initial solution to the β-hill climbing optimizer. The solution is also evaluated using the same objective function $F(\boldsymbol{P}^i)$ as given in Eq. (1). In β -hill climbing, three operators are iteratively executed to improve $\boldsymbol{P}^i$:

1. $\mathcal{N}$-operator: The current solution $\boldsymbol{P}^i$ is affected by its neighbouring solution to yield a new solution $\boldsymbol{P}'^i$ as follows: $P'^i_j = P^i_j \pm U(0, 1) \times bw$ where bw is bandwidth between the current and neighboring solution set normally with a small value.
2. β-operator: Within the probability of β parameter where $\beta \in [0, 1]$ as shown in Eq. (12), the β-operator is used.

$$P'_i \leftarrow \begin{cases} P'_i & rnd \leq \beta \\ P_i & otherwise. \end{cases} \quad (12)$$

Where rnd is a function generates a uniform value between 0 and 1. Any generation unit P^i_j will be regenerated from its feasible range: $P'^i_j \in [P^{(min)}_i, P^{(max)}_i] \quad \forall i \in (1, 2, \dots, N)$.

In case the feasibility of the generated solution is broken, the maintenance strategy used in SCA is again invoked to maintain the constructed solution obtained by βHC optimizer as pseudo-coded in Algorithm 2.

3. $\mathcal{S}$-operator: The current solution $\boldsymbol{P}^i$ is replaced by new solution $\boldsymbol{P}'^i$, in case its objective function value is better. The pseudo-code of the β hill climbing optimizer is shown in Algorithm 3.

Algorithm 3 βHC(***P***) pseudo-code

```
{——— Step 1: Initialize the $\beta$HC parameters ———}
Initialize the parameters ($bw$, $\beta$ and $It_\beta$).
$t = 0$
while $(t \leq It_\beta)$ do
    $\boldsymbol{P}' = \boldsymbol{P}$
    Select $j \in [1, N]$
    $P'_j = P'_j \pm U(0, 1) \times bw$        {$\mathcal{N}$-operator}
    $\boldsymbol{P}'' = \boldsymbol{P}'$
    for $j = 1, \cdots, N$ do
        if $(rnd \leq \beta)$ then
            $P''_j \in [P_j^{(min)}, P_j^{(max)}]$
        end if     {$rnd \in [0, 1]$}
    end for     {$\beta$-operator}
    Maintaining ($\boldsymbol{P}''$)     {See Algorithm 2}
    if $(f(\boldsymbol{P}'') \leq f(\boldsymbol{P}))$ then
        $\boldsymbol{P} = \boldsymbol{P}''$
    end if
    $t = t + 1$
end while
```

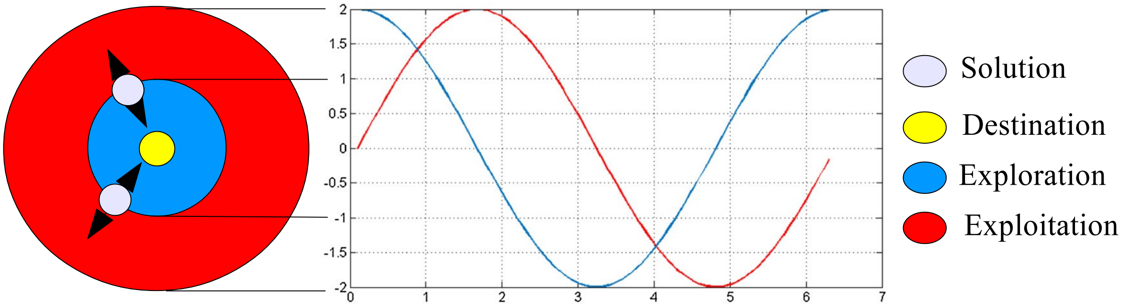


Fig. 2 Convergence behaviour of sine consing operators in SCA

Step 4: Update SBM The proposed SCA-βHC updates its memory (SBM) when an improvement is achieved. In other words, if the new solution $\boldsymbol{P}'^i$ is better than the current solution $\boldsymbol{P}^i$, the new solution $\boldsymbol{P}'^i$ will be included in SBM while the $\boldsymbol{P}^i$ will be excluded. Furthermore, the best solution P^{gbest} is updated in case the improvement is achieved.

Step 5: Termination Condition The proposed SCA-βHC is terminated when the stop condition which normally depends on the maximum number of iterations (It_{Max}) is reached. The best solution thus far reached is then delivered.

3.1 Convergence analysis of SCA-βHC

In order to study the convergence behaviour, the operators of standard SCA shall be initially analysed. The operators of standard SCA are the sine and cosine operators. Normally, the convergence analysis is studied in terms of exploration and exploitation process. The exploration refers to the ability of SCA to navigate several search space regions at the same time while exploitation refers to the ability of SCA to converge to local minima in each search space region.

The search behaviour of sine and cosine operators is visualized in Fig. 2. During the SCA search, the positions of the current solutions are updated by either sine or cosine operator. The sine cosine curve move in a regular way within the value range of $[-2, 2]$. Mirjalili (2016b) determines the range $[-2, -1)$ or $(1, 2]$ where the solutions are moved to explore the search space region while the range of $[-1, 1]$ is used to locally exploit the search space region of each solution. It should be noticed that the range of the sine and cosine operators is gradually decreased during the search process. Therefore, the SCA start with high exploration, the search moved toward exploitation stage at the final stage of runs. In Fig. 2, the exploration and exploitation is visualised using the equation $r_1 = a - \frac{t}{It_{Max}} a$ where a is supposed to be constant and t is the current iteration while It_{Max} is the maximum number of iterations.

In Fig. 3, the range of the sine and cosine equations is decreased when the iteration number is increased. Initially, the value of a is chosen to be 3 which refer to the range of moving the current solution from exploration to exploitation stage. As can be recognized, the search process with SCA initiated with high exploration power, the range of the search space is gradually degraded during the search to recognize the exploitation process in the final stage of run. Therefore, with increasing the value of a, the SCA convergence behaviour will be initiated with a wider search space where the exploration is much concentrated. The search will move slowly from the exploration stage to exploitation stage leading to slow convergence. Oppositely, when the value of a is small, the range of the search space will be narrow where the search will move quickly from exploration stage to exploitation stage with fast convergence.

3.2 Time complexity of the SCA-βHC

In order to analyse the time complexity of the proposed method, the pseudo-code of SCA-βHC shown in Algorithm 1 is analyzed. Initially, the generation of the initial population required a time complexity of $\mathcal{O}(NM)$ where N is the number of generation units while M is the population size. In the improvement loop where the position of each search agent is updated, the "while" loop in line 9 required $\mathcal{O}(It_{max})$. The inner two "for" loops also required a time complexity of $\mathcal{O}(NM)$. The maintaining($\boldsymbol{P}$) procedure shown in Algorithm 2 required time complexity of $\max\{\mathcal{O}(N), \mathcal{O}(dis)\}$ where dis is the distance bandwidth between the total load demand and the actual load demand (P_D). The time complexity of the βHC algorithm is equal to $\mathcal{O}(It_\beta \times N)$ where It_β is the iteration number required for βHC algorithm. Note that this is the extra time complexity related to the SCA-βHC in comparison with original version of SCA.

In addition to the time complexity related to the operations of the proposed SCA-βHC method, the ELD objective

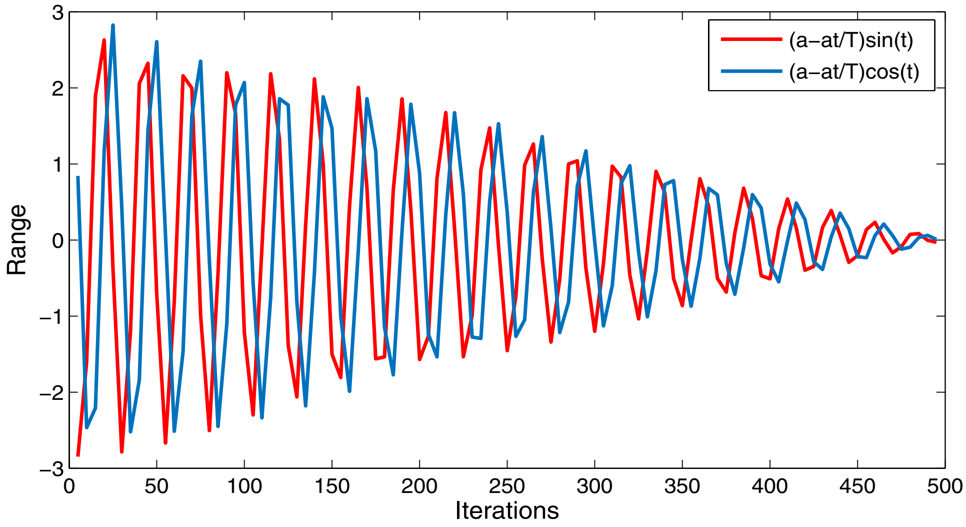


Fig. 3 Convergence analysis of sine cosine algorithm

function is also used heavily at each iteration of SCA and at each iteration of βHC method.The time complexity required to evaluate any ELD solution using the objective function is $\mathcal{O}(N)$.

Overall, the time complexity of the proposed SCA-βHC method is $\mathcal{O}(NM) + \mathcal{O}(It_{max} \times N \times M \times \max\{\mathcal{O}(N), \mathcal{O}(dis)\} \times It_{\beta} \times N \times N^2)$. Apparently, the time complexity of the proposed SCA-βHC method is higher than that required for original SCA.

4 Experiments and results

In this section, the effectiveness and robustness of the proposed SCA-βHC on solving ELD problems through six case studies with different numbers of generation units and load demand (LD) is tested. These cases are:

(I) Case 1: 3 generation units with LD= 850 MW, (II) Case 2: 6 generation units with LD= 1263 MW, (III) Case 3: 13 generation units with LD= 2520 MW, (IV) Case 4: 13 generation units with LD= 1800 MW, (V) Case 5: 15 generation units with LD= 2630 MW, and (VI) Case 6: 40 generation units with LD= 10500 MW.

It should be noted that in the two cases (Case 2 and Case 5), the ramp rate limits and prohibited zones of the units were respected in the practical application. While the valve point loading (VPL) effect is considered in the other four cases. Furthermore, the transmission loss is neglected in these cases.

Our experiments are carried out on MATLAB (R2014a) using a laptop with Intel(R) Core(TM) i7 2.8GHz, 16 GB of RAM, and operating system managing Window 10. The performance of the proposed SCA-βHC algorithm are compared with the state-of-the-art algorithms. The comparative algorithms for all tested cases are shown in Table 1.

Fourteen experimental scenarios are designed in order to study the influence of the various parameter settings on the performance of the proposed SCA-βHC algorithm. These scenarios are divided into three groups as shown in Table 2. SenExp1 to SenExp4 are designed to study the impact of the parameter β on the performance of SCA-βHC algorithm using four different values (β= 0, β= 0.01, β= 0.1, and β= 0.5), The higher value of β leads to higher rate of exploration. The next five scenarios (SenExp5 to SenExp9) are designed to study the effect of the parameter *bw* on the behaviour of the proposed algorithm using various values (*bw*=0.001, *bw*=0.01, *bw*=0.1, *bw*=0.5, and *bw*=0.95). The higher values of *bw* leads to higher rate of exploitation. Finally, the last five scenarios (SenExp10 to SenExp14) are designed to study the influence of the parameter R_{β}. This parameter is studied using five different values as follows: R_{β}=0, R_{β}=0.001, R_{β}=0.01, R_{β}=0.1, and R_{β}=0.3. The higher value of R_{β} leads to higher rate of invoking the β-hill climbing optimizer and thus higher rate of exploitation. In instance, when the value of the parameter R_{β} is equal to zero as shown in SenExp10, its meaning the calling of the β-hill climbing optimizer is neglected and thus the original version of SCA is used in this scenario. These settings are introduced using an *ad hoc* method in order to find the best configurations of the proposed SCA-βHC algorithm. On other hand, the settings of the other parameters as follows: the size of population is set to 30, and the number of replicated runs is also set to 30 for each experimental scenarios and the maximum number of iterations used is 100,000.

4.1 Case 1: 3-unit generator system with demand of 850 MW

This case study comprises of three generating units with the expected load demand to be meet by all of the three units is

Table 1 Comparative methods

#	Algorithm	Test cases						References
		1	2	3	4	5	6	
1	ABC	–	–	–	–	–	✓	Hemamalini and Simon (2010)
2	ABOMDE	–	–	–	✓	–	✓	Lohokare et al. (2012)
3	ACO	–	–	✓	–	–	✓	Pothiya et al. (2010)
4	ARCGA	–	–	–	–	–	✓	Sayah and Hamouda (2013)
5	βHC	✓	–	✓	✓	–	✓	Al-Betar et al. (2018a)
6	β-GWO	✓	–	✓	✓	–	✓	Al-Betar et al. (2020b)
7	BBO	–	✓	–	–	–	–	Bhattacharya and Chattopadhyay (2010c)
8	BGO	–	–	–	–	–	✓	Bhattacharya and Chattopadhyay (2010c)
9	CBPSO-RVM	–	–	–	–	–	✓	Lu et al. (2010)
10	CLCS-CLM	–	–	–	–	✓	✓	Huang et al. (2020)
11	CPSO	–	✓	–	–	✓	–	Safari and Shayeghi (2011)
12	CS	–	–	–	–	✓	✓	Huang et al. (2020)
13	CS-CLM	–	–	–	–	✓	✓	Huang et al. (2020)
14	CSOMA	–	–	–	–	–	✓	dos Santos Coelho and Mariani (2010)
15	DE-BBO	–	–	–	–	–	✓	Bhattacharya and Chattopadhyay (2010b)
16	DHS	–	–	–	✓	–	–	Wang and Li (2013)
17	DVL-MILP	–	–	–	–	–	✓	Fraga et al. (2012)
18	FAPSO	–	–	–	–	–	✓	Niknam et al. (2011)
19	FAPSO-NM	–	–	–	–	–	✓	Niknam et al. (2011)
20	FCASO-SQP	✓	–	✓	✓	–	✓	Cai et al. (2012b)
21	FFA	–	–	–	–	–	✓	Sinha et al. (2003)
22	GA	–	✓	–	–	–	–	Safari and Shayeghi (2011)
23	GAPSO	–	–	–	–	✓	–	Sudhakaran et al. (2007)
24	GA-PS-SQP	✓	–	–	✓	–	✓	Alsumait et al. (2010)
25	GSO	–	–	–	–	–	✓	Moradi-Dalvand et al. (2012)
26	GWO	–	–	–	✓	–	✓	Suleiman et al. (2015)
27	HBF	–	✓	–	–	✓	–	Panigrahi and Pandi (2008)
28	HCA	–	–	–	–	–	✓	Coelho and Mariani (2010)
29	HCASO	✓	–	✓	✓	–	✓	Cai et al. (2012b)
30	HCPSO	✓	–	✓	–	–	✓	Cai et al. (2012a)
31	HCPSO-SQP	✓	–	✓	–	–	✓	Cai et al. (2012a)
32	HGWO	–	–	–	–	–	✓	Jayabarathi et al. (2016)
33	HHS	–	–	–	✓	–	✓	Pandi et al. (2011)
34	HMAPSO	–	–	–	✓	–	✓	Kumar et al. (2011)
35	HQIPSO	–	–	–	✓	–	✓	Chakraborty et al. (2011)
36	HS1	–	✓	–	–	✓	–	Panigrahi et al. (2012)
37	HS2	–	–	–	✓	–	–	Coelho and Mariani (2009)
38	IGWO	–	✓	–	–	–	–	Mehmood and Ahmad (2018)
39	IHS1	–	✓	–	–	✓	–	Panigrahi et al. (2012)
40	IHS2	–	–	–	✓	–	–	Coelho and Mariani (2009)
41	IPSO	–	✓	–	–	✓	–	Safari and Shayeghi (2011)
42	JAYA	–	–	–	–	✓	–	Yu et al. (2019)
43	MDE	✓	–	–	–	–	✓	Amjady and Sharifzadeh (2010)
44	MPDE	–	–	–	✓	–	✓	Li et al. (2019)
45	MSL	–	–	–	✓	–	–	Mohammadi-Ivatloo et al. (2012)
46	NDS	✓	–	–	✓	–	✓	Lin et al. (2011)
47	NPSO	–	✓	–	–	–	–	Selvakumar and Thanushkodi (2007)
48	NPSO-LRS	–	✓	–	–	–	–	Selvakumar and Thanushkodi (2007)
49	NSABC	–	–	–	✓	–	✓	Awadallah et al. (2019)

Table 1 (continued)

#	Algorithm	Test cases						References
		1	2	3	4	5	6	
50	NSS	✓	–	–	✓	–	✓	Tsai et al. (2011)
51	NUHS	✓	–	✓	✓	–	✓	Al-Betar et al. (2018b).
52	PSO	–	✓	–	–	✓	–	Safari and Shayeghi (2011)
53	PSO-LRS	–	✓	–	–	–	–	Selvakumar and Thanushkodi (2007)
54	PSO-MSAF	–	–	–	–	–	✓	Subbaraj et al. (2010)
55	PVHS	–	✓	–	–	✓	–	Panigrahi et al. (2012)
56	QGSO	–	–	–	–	–	✓	Moradi-Dalvand et al. (2012)
57	QIPSO	✓	–	–	✓	–	✓	Chakraborty et al. (2011)
58	RCGA	–	–	–	✓	–	✓	Chakraborty et al. (2011)
59	SDE	✓	–	–	–	–	–	Srinivasa Reddy and Vaisakh (2013)
60	SDP	–	–	✓	✓	–	✓	Alawode et al. (2018)
61	SOH-PSO	–	✓	–	–	✓	–	Chaturvedi et al. (2008)
62	SOMA	–	–	–	–	–	✓	Coelho and Mariani (2010)
63	SSA	–	–	–	–	–	✓	Sakthivel et al. (2020)
64	THS	✓	–	✓	✓	–	✓	Al-Betar et al. (2016)
65	TLA	–	–	–	–	–	✓	Azizipanah-Abarghooee et al. (2012)
66	TS1	–	–	✓	–	–	✓	Pothiya et al. (2010)
67	TS2	–	–	✓	–	–	–	Khamsawang and Jiriwibhakorn (2010)
68	TSARGA	–	–	–	–	–	✓	Subbaraj et al. (2011)

Table 2 Experimental scenarios

Experimental	β	*bw*	R_β	Iterations					
Scenario				Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
SenExp1	0	0.5	0.01	1000	2000	5000	5000	5000	10000
SenExp2	0.01								
SenExp3	0.1								
SenExp4	0.5								
SenExp5		0.001	0.01	1000	2000	5000	5000	5000	10000
SenExp6		0.01							
SenExp7		0.1							
SenExp8		0.5							
SenExp9		0.95							
SenExp10			0	20000	40000	50000	50000	50000	200000
SenExp11			0.001	2000	5000	10000	10000	10000	50000
SenExp12			0.01	1000	2000	5000	5000	5000	10000
SenExp13			0.1	100	300	500	500	500	5000
SenExp14			0.3	50	100	200	200	200	2000

850 MW. The unit data is taken from (Walters and Sheble 1993). Table 3 illustrates the results of studying the influence of different parameter settings on the performance of the proposed SCA-βHC algorithm. In this table, the cost of each generating unit in the best solution obtained for each experiential scenario is recorded. Furthermore, the total cost of the best solution, as well as, the mean and the standard derivation of the results are also summarized in this table.

SenExp1 to SenExp4 are designed to study the effect the β parameter (i.e., SenExp1 (β=0), SenExp2 (β=0.01), SenExp3 (β=0.1), and SenExp4 (β=0.5)) on the behaviour of the SCA-βHC algorithm. The configurations of these scenarios are presented in Table 2. From Table 3, it can be seen that the four scenarios are able to obtain the same results in terms of the best fuel cost (i.e., 8,234.07 \$/h). On another hand, the standard derivation values in Table 3 reflect the robustness of the proposed SCA-βHC algorithm. It can be observed from the results provided in Table 3 that the proposed SCA-βHC in SenExp3 is more robust than those of the other three scenarios (i.e., SenExp1, SenExp2, and

Table 3 The results of SCA-βHC on 3-unit generator system with demand of 850 MW using different parameter settings

Unit	Studying β				Studying *bw*					Studying R_β				
	SenExp1	SenExp2	SenExp3	SenExp4	SenExp5	SenExp6	SenExp7	SenExp8	SenExp9	SenExp10	SenExp11	SenExp12	SenExp13	SenExp14
u1	300.2669	300.2669	300.2669	300.2669	300.2669	300.2669	300.2669	300.2669	300.2669	300.2669	300.2669	300.2669	300.2669	300.2669
u2	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000	400.0000
u3	149.7331	149.7331	149.7331	149.7331	149.7331	149.7331	149.7331	149.7331	149.7331	149.7331	149.7331	149.7331	149.7331	149.7331
Total cost	**8234.07**	**8234.07**	**8234.07**	**8234.07**	**8234.07**	**8234.07**	**8234.07**	**8234.07**	**8234.07**	**8234.07**	**8234.07**	**8234.07**	**8234.07**	**8234.07**
Mean cost	8234.07	8234.07	8234.07	8234.07	8234.07	8234.07	8234.07	8234.07	8234.07	8234.07	8234.07	8234.07	8234.07	8234.07
Stdv	1.27E-06	1.40E-07	1.14E-07	4.15E-07	2.22E-10	3.15E-09	2.88E-08	1.14E-07	4.96E-07	6.98E-04	1.26E-09	2.22E-10	3.01E-10	1.48E-10

Table 4 Comparison results on 3-unit generator system with demand of 850 MW

Algorithm	Best cost	Mean cost
SCA-βHC	**8234.07**	8234.07
SCA	**8234.07**	8234.07
βHC	**8234.07**	8234.07
β-GWO	**8234.07**	8234.07
FCASO-SQP	**8234.07**	8234.07
GA-PS-SQP	8234.10	8234.10
HCASO	**8234.07**	8234.07
HCPSO	**8234.07**	NA
HCPSO-SQP	**8234.07**	8234.07
MDE	**8234.07**	NA
NDS	**8234.07**	8234.07
NSS	8,234.08	8234.08
NUHS	**8234.07**	8234.07
QIPSO	**8234.07**	8234.10
SDE	**8234.07**	NA
THS	**8234.07**	8234.07

SenExp4). Therefore, the value of the β parameter will be set to 0.01 in the following experiments.

Similarly, the impact of the *bw* parameter on the performance of the proposed algorithm is studied using five different scenarios with different configurations (i.e., SenExp5 (*bw*=0.001), SenExp6 (*bw*=0.01), SenExp7 (*bw*=0.1), SenExp8 (*bw*=0.5), and SenExp9 (*bw*=0.95)). From Table 3, it can be seen that the performance of the proposed SCA-βHC in the five scenarios are almost similar, by obtaining the same results in terms of total cost, and mean of the results. However, the performance of SenExp5 is more robust than the other four scenarios based on the standard derivation results. Therefore, the value of the *bw* parameter will be set to 0.001 in the following experiments.

The last five scenarios (SenExp10 to SenExp14) are designed to study the effect of the parameter R_β on the performance of the proposed SCA-βHC algorithm. Each experimental scenario is introduced with specific value of R_β as follows: SenExp10 (R_β=0), SenExp11 (R_β=0.001), SenExp12 (R_β=0.01), SenExp13 (R_β=0.1), and SenExp14 (R_β =0.3). The results of running these scenarios are recorded in Table 3. Apparently, the five scenarios obtained the same best results. This is due to the simplicity of the problem studied in this case. On the other hand, the performance of the proposed SCA-βHC algorithm on SenExp14 is more robust than those of the other four scenarios based on the standard derivation results.

The results of the proposed SCA-βHC against those of the other comparative methods are recorded in Table 4. It can be seen from the results illustrated in Table 4 that the performance of the proposed algorithm is similar to 13 out of 15 other comparative methods, by obtained the optimal total fuel cost (i.e., 8,234.07 \$/h).

4.2 Case 2: 6-unit generator system with demand of 1263 MW

In this section, the ELD problem includes six generating units with the expected load demand of all units is 1263 MW. The characteristics of the generating units are provided in Das and Suganthan Das and Suganthan (2010). In this case of ELD problem, the ramp rate limits and prohibited zones of the units were considered. Table 5 shows the results of studying the three parameters (i.e., β, bw, and R_β) of the proposed SCA-βHC algorithm using this case of ELD problem (i.e., case 2). These parameters are studied using fourteen experimental scenarios. The configurations of these scenarios are illustrated in Table 2. The best solution with the minimum total fuel cost is recorded in Table 5. Furthermore, the mean costs, the standard derivation (Stdv.), the total power output of the besat solution, and the transmission loss (P_L) of the best solution are also provided in Table 5.

Table 5 The results of SCA-βHC on 6-unit generator system with demand of 1263 MW using different parameter settings

Unit	Studying β				Studying bw					Studying R_β				
	SenExp1	SenExp2	SenExp3	SenExp4	SenExp5	SenExp6	SenExp7	SenExp8	SenExp9	SenExp10	SenExp11	SenExp12	SenExp13	SenExp14
u1	461.3571	445.2667	443.2907	446.1885	443.7381	451.1399	443.7343	446.1885	445.0888	445.4002	447.6929	443.7381	445.8100	443.5177
u2	166.7088	171.5940	167.6555	173.3997	171.5929	172.5519	173.7467	173.3997	174.8443	186.2030	171.6989	171.5929	175.5422	174.7330
u3	264.8249	257.8924	264.9224	260.5301	263.2908	262.0418	264.1692	260.5301	259.3219	257.3371	263.1559	263.2908	261.7730	264.1966
u4	139.6412	145.6300	147.6303	146.5351	144.1608	141.5791	146.8146	146.5351	147.5050	139.8582	139.0385	144.1608	141.4126	143.9844
u5	155.8851	169.5972	164.2202	163.1581	165.5969	162.3922	160.9229	163.1581	162.1932	135.1230	165.1707	165.5969	164.4745	163.7919
u6	87.0505	85.4459	87.6190	85.5226	87.0454	85.7496	85.9234	85.5226	86.3404	111.4298	88.7881	87.0454	86.4566	85.1836
Total cost	15446.95	15444.82	15444.81	15444.43	15444.34	15444.39	15444.46	15444.43	15444.53	15461.47	15444.53	15444.34	15444.31	**15444.30**
Mean cost	15452.67	15447.21	15446.12	15445.18	15445.02	15445.14	15445.16	15445.18	15445.41	15495.10	15445.70	15445.02	15444.79	15444.77
Stdv.	4.01E+00	1.51E+00	8.07E-01	4.54E-01	3.66E-01	4.77E-01	4.57E-01	4.54E-01	4.85E-01	1.97E+01	6.83E-01	3.66E-01	3.23E-01	2.69E-01
Output	1275.47	1275.43	1275.34	1275.33	1275.42	1275.45	1275.31	1275.33	1275.29	1275.35	1275.55	1275.42	1275.47	1275.41
P_L	12.4676	12.4262	12.3381	12.3342	12.4249	12.4545	12.3111	12.3342	12.2936	12.3519	12.5450	12.4249	12.4689	12.4072

The first four experimental scenarios (SenExp1–SenExp4) are designed to study the effect of the parameter β on the performance of the proposed algorithm using four various values (0, 0.01, 0.1, and 0.5). The results of running these scenarios are summarized in Table 5. From Table 5, it can be shown that the results of the proposed SCA-βHC algorithm are gradually enhanced by increasing the value of β, where the last scenario (i.e., SenExp4) achieved the minimum total fuel cost (i.e., 15444.43 \$/h). As a result, the β parameter will be set to 0.5 in the following experiments of this case study.

Similarly, SenExp5–SenExp9 are designed to study the influence of the parameter bw on the behaviour of the proposed algorithm using this case of ELD problem. It should be noted that the values of bw studied in these scenarios are: SenExp5 (bw=0.001), SenExp6 (bw=0.01), SenExp7 (bw=0.1), SenExp8 (bw=0.5), and SenExp9 (bw=0.95). The results of running these experimental scenarios are provided in Table 5. Remarkably, the performance of the proposed algorithm in SenExp5 perform is better than the other four scenarios in terms the total fuel cost and the mean costs. Notably, the differences among the results of these scenarios are not exceeds 0.19 \$/h. This can prove the influence of the parameter bw on the behaviour of the proposed SCA-βHC algorithm used for this ELD case is not significant. Therefore, the bw parameter will be set to 0.001 in the remaining experiments for Case 2.

The last five experimental scenarios (SenExp10–SenExp14) are designed to study the impact of the parameter R_β on the performance of the proposed SCA-βHC algorithm. The R_β is investigated using the following values: 0, 0.001, 0.01, 0.1, and 0.3. It should be noted that, the higher

Table 6 Comparison results on 6-unit generator system with demand of 1263 MW

Algorithm	Best cost	Mean cost	P_L	Output
SCA-βHC	15444.30	15444.77	12.4072	1275.41
SCA	15461.47	15495.10	12.3519	1275.35
BBO	15443.10	15443.10	12.4460	1275.45
CPSO	15447.00	15449.00	12.9583	1276.00
GA	15459.00	15469.00	13.0217	1276.03
HBF	15443.82	15446.95	12.4437	1275.40
HS1	15447.50	15455.80	12.3606	1275.37
IGWO	**15442.20**	15442.60	12.3123	1275.30
IHS1	15445.64	15453.50	12.5886	1275.56
IPSO	15444.00	15446.30	12.5480	1275.51
NPSO	15450.00	NA	12.9470	1275.95
NPSO-LRS	15450.00	NA	12.9361	1275.94
PSO	15450.00	15454.00	12.9584	1276.01
PSO-LRS	15450.00	NA	12.9571	1275.95
PVHS	15444.96	15451.40	12.3672	1275.36
SOH-PSO	15446.00	NA	12.5500	1275.55

the value of R_β, the higher probability of calling the β-hill climbing optimizer, and thus higher rate of the exploitation will be. From Table 5, it can be observed that the results of the proposed algorithm are gradually enhanced by increasing the value of R_β. Apparently, the performance of the proposed algorithm in SenExp14 outperforms the performance of the other four scenarios (i.e., SenExp10–SenExp13), by obtaining the minimum total fuel cost (15444.30 $/h). On the other hand, the difference between the results of SenExp13 and SenExp14 is equal to 0.01 $/h, and the difference between the results of SenExp12 and SenExp14 is equal to 0.04 $/h. This advice the research to escape the bigger values of R_β parameter.

Finally, the performance of the proposed SCA-βHC is compared with the other fifteen comparative methods as shown in Table 6. It can be see from Table 6 that the IGWO algorithm outperforms the proposed SCA-βHC algorithm, as well as the other comparative methods by obtaining the minimum fuel cost (15442.20 $/h). While the proposed SCA-βHC algorithm perform better than 11 out of 16 competitive methods by getting the fifth best results (15444.30 $/h).

4.3 Case 3: 13-unit generator system with demand of 2520 MW

The 13-unit generator system comprises of 13 generating units. This system considers the valve-point effect, while the transmission loss is neglected. The required load demand is 2520 MW. The data of the units in this case study is taken from (Walters and Sheble 1993). Table 7 summarizes the results of studying the various parameter settings on the performance of the proposed SCA-βHC algorithm using fourteen experimental scenarios. In this table, the best solution obtained by running each of the experimental scenarios is recorded. In addition, the total cost of the best solution, the mean cost, and the standard derivation are also provided. It should be noted that the best solution is highlighted using **bold** font.

Table 7 shows the results produced to study the impact of the parameter β on the performance of the proposed SCA-βHC algorithm using four experimental scenarios with different values (i.e., SenExp1 (β=0), SenExp2 (β=0.01), SenExp3 (β=0.1), and SenExp4 (β=0.5)). From Table 7, it can be shown that the performance of the proposed SCA-βHC algorithm in SenExp3 is better than other three scenarios by obtaining the best fuel cost (24164.12 $/h). This guide us to set the parameter β to be 0.1 for the next set of experiments.

SenExp5 to SenExp9 are designed to study the influence of the various settings of the parameter *bw* on the behaviour of the SCA-βHC algorithm (SenExp5 (*bw*=0.001), SenExp6 (*bw*=0.01), SenExp7 (*bw*=0.1), SenExp8 (*bw*=0.5), and SenExp9 (*bw*=0.95)). Table 7 summarizes the results of running these five scenarios. Clearly, it can be seen that the performance of SenExp7 is better than other four scenarios by getting the best solution with the minimum fuel cost (i.e., 27164.09$/h). The parameter *bw* will be set to 0.1 in the following experimental scenarios.

Similarly, the results of studying the impact of varying values of parameter R_β on the performance of the proposed SCA-βHC algorithm are provided in Table 7. It should be noticed that the parameter R_β is studied using five different values in five experimental scenarios (i.e., SenExp10 (R_β=0), SenExp11 (R_β=0.001), SenExp12 (R_β=0.01), SenExp13 (R_β=0.1), and SenExp14 (R_β=0.3)). From 7, it can be observed that the performance of the proposed SCA-βHC algorithm in SenExp11 to SenExp14 is better than SenExp10. This is because the value of the parameter R_β is set to zero. However, the last three scenarios (i.e., SenExp12–SenExp14) are able to obtain the same best results (27164.09$/h). In summary, the performance of the proposed SCA-βHC algorithm on SenExp13 is more robust than the other four scenarios as suggested by standard derivation results.

Finally, Table 8 shows the results of the proposed SCA-βHC algorithm against those of the other comparative methods. The numbers in this table represents the best results achieved, as well as the mean of the results. It should be noted that the results of the proposed SCA-βHC algorithm represents the best results recorded in Table 7. Apparently, the performance of the NUHS and THS algorithms are better than the other competitive methods by obtaining the minimum fuel cost (24164.06$/h). The proposed SCA-$\beta$HC algorithm obtained the second minimum fuel cost (24164.09$/h), where the difference between the results of the SCA-βHC

Table 7 The results of SCA-βHC on 13-unit generator system with demand of 2520 MW using different parameter settings

Unit	Studying β				Studying *bw*					Studying R_β				
	SenExp1	SenExp2	SenExp3	SenExp4	SenExp5	SenExp6	SenExp7	SenExp8	SenExp9	SenExp10	SenExp11	SenExp12	SenExp13	SenExp14
u1	679.7617	628.3185	628.3185	628.3509	628.3187	628.3186	628.3185	628.3185	628.3185	630.0390	628.3186	628.3185	628.3185	628.3185
u2	359.9006	299.1336	299.1963	298.6862	298.7798	299.1844	299.1942	299.1963	299.1898	360.0000	299.1992	299.1942	299.1934	299.1916
u3	294.6811	296.3404	294.5266	299.1168	295.5738	295.2485	294.5033	294.5266	294.5386	360.0000	294.5325	294.5033	294.5071	294.5046
u4	159.7330	159.7331	159.7325	159.5346	159.7331	159.7330	159.7328	159.7325	159.7306	160.6614	159.7330	159.7328	159.7330	159.7328
u5	159.7331	159.7331	159.7326	159.1360	159.7331	159.7338	159.7331	159.7326	159.7326	157.7379	159.7328	159.7331	159.7331	159.7331
u6	159.7331	159.7328	159.7330	159.6290	159.7331	159.7334	159.7330	159.7330	159.7330	60.0000	159.7328	159.7330	159.7331	159.7331
u7	159.7330	159.7331	159.7329	159.8542	159.7331	159.7323	159.7328	159.7329	159.7332	158.5796	159.7325	159.7328	159.7330	159.7331
u8	159.7331	159.7330	159.7321	158.9161	159.7331	159.7316	159.7331	159.7321	159.7332	174.0237	159.7331	159.7331	159.7330	159.7329
u9	159.7331	159.7331	159.7330	158.8837	159.7331	159.7330	159.7330	159.7330	159.7320	151.7919	159.7331	159.7330	159.7330	159.7331
u10	77.2581	75.8847	77.3934	77.3235	77.4340	77.0100	77.3973	77.3934	77.3760	77.1666	77.4001	77.3973	77.3948	77.3978
u11	40.0000	77.3330	77.3939	76.1237	77.0886	77.3947	77.3943	77.3939	77.3900	120.0000	77.4001	77.3943	77.3936	77.3986
u12	55.0000	92.2206	92.3952	92.2593	92.1534	92.1687	92.3947	92.3952	92.3983	55.0000	92.3755	92.3947	92.3944	92.3952
u13	55.0000	92.3710	92.3806	92.1858	92.2531	92.2779	92.4000	92.3806	92.3944	55.0000	92.3768	92.4000	92.3999	92.3957
Total cost	24509.07	24166.96	24164.12	24177.52	24166.58	24165.27	**24164.09**	24164.12	24164.15	24610.82	24164.13	**24164.09**	**24164.09**	**24164.09**
Mean cost	24530.34	24188.73	24164.42	24199.03	24168.14	24166.83	24164.60	24164.42	24164.53	24681.65	24165.79	24164.60	24164.43	24164.38
Stdv	2.08E+01	1.96E+01	2.68E-01	7.75E+00	6.73E-01	7.59E-01	4.89E-01	2.68E-01	2.15E-01	3.68E+01	1.33E+00	4.89E-01	2.69E-01	2.84E-01

Table 8 Comparison results on 13-unit generator system with demand of 2520 MW

Algorithm	Best cost	Mean cost
SCA-βHC	24164.09	24164.38
SCA	24610.82	24681.65
ACO	24174.39	24211.09
βHC	24164.18	24204.90
β-GWO	24164.10	24164.20
FCASO-SQP	24190.63	NA
HCASO	24212.93	NA
HCPSO	24211.56	NA
HCPSO-SQP	24190.97	NA
NUHS	**24164.06**	24185.61
SDP	24169.92	NA
THS	**24164.06**	24195.21
TS1	24180.31	24243.37
TS2	24171.21	24184.06

algorithm and the results of the NUHS or THS algorithms is (0.03\$/h). Furthermore, the performance of the proposed SCA-βHC algorithm is better than the original version of since cosine algorithm (SCA).

4.4 Case 4: 13-unit generator system with demand of 1800 MW

The proposed SCA-βHC algorithm is tested using another test system consists of thirteen generating units. The expected load power demand to be meet by all of the generating units is 1800 MW. The data of generating units of this system were extracted from (Walters and Sheble 1993). The results obtained for studying the influence of the various parameter settings on the performance of the proposed SCA-βHC algorithm are summarized in Table 9. Fourteen experimental scenarios are also designed for this purpose, where the parameter settings of these scenarios are provided in Table 2. It should be noted that the best solution obtained by each of the experimental scenarios with the minimum total fuel cost is recorded in Table 9. Furthermore, the mean of the costs obtained by running each scenario 30 times, as well as the standard derivation are summarized in Table 9.

The results of studying the impact of the parameter β on the behaviour of the proposed SCA-βHC algorithm using various settings are recorded in Table 9. The first four experimental scenarios (SenExp1 (β=0), SenExp2 (β=0.01), SenExp3 (β=0.1), and SenExp4 (β=0.5)) are designed for this purpose. The best results achieved by running these scenarios are illustrated. From Table 9, it can be observed that the performance of SCA-βHC algorithm in SenExp3 is better than the other three scenarios (i.e., SenExp1, SenExp2, and SenExp4) by getting the minimum total fuel cost (17960.39 \$/h). This is because that the value of 0.1 is the threshold value of the parameter β where it assists to strike a right balance between exploration and exploitation and thus guide SCA-βHC algorithm to achieve better results.

The next five experimental scenarios (SenExp5–SenExp9) are designed to study the effect of the parameter *bw* on the behaviour of the proposed SCA-βHC algorithm using five different values (i.e., 0.001, 0.01, 0.1, 0.5, and 0.95). The results of running these scenarios are summarized in Table 9. In this table, the best solution obtained by each scenario, the total fuel cost of the best solution, the mean costs, and the standard derivation are recorded. From Table 9, it can be seen that the performance of the proposed SCA-βHC algorithm in SenExp8 is better than the other four scenarios, by getting the minimum fuel cost (17960.39 \$/h). The lower value of *bw* leads to worst results as seen in SenExp5 and SenExp6. Furthermore, the higher value of *bw* leads to worst results as seen in SenExp9. This is prove that the value 0.5 is the threshold value of the parameter *bw* where it is able to yield the right balance between the exploration and exploitation. Thus the parameter *bw* is set to be 0.5 in the following experiments.

The last experimental scenarios (i.e., SenExp10–SenExp14) are introduced to study the impact of the parameter R_β on the performance of the proposed SCA-βHC algorithm using five different values (i.e., 0, 0.001, 0.01, 0.1, and 0.3). From Table 9, it can be observed that the performance of the proposed SCA-βHC algorithm in SenExp12 is better than the other four scenarios by getting the minimum total fuel cost (17960.39 \$/h). However, the performance of the proposed SCA-βHC algorithm in SenExp10 is the worst. This is due to the fact that the value of the parameter R_β is equal to zero, and thus the β-hill climbing optimizer is neglected. Based on the above discussions, the following parameter settings: β=0.1, *bw*=0.5, and R_β=0.01, are guided the proposed SCA-βHC algorithm to achieve the best results for this case of the ELD problem.

Table 10 shows the comparison of SCA-βHC performance against the twenty-four other comparative methods. Clearly, the performance of the proposed SCA-βHC algorithm outperforms the original version of sine cosine algorithm (i.e., SCA). On the other hand, five of the competitive methods (i.e., DHS, IHS, MPDE, NUHS, and THS) are able to obtain the minimum fuel cost (17960.37 \$/h), while the proposed SCA-βHC algorithm are able to achieve the third minimum fuel cost (17960.39 \$/h). It should be noticed that the difference between the results of the proposed SCA-βHC algorithm and the best results published by others is (0.02\$/h).

Table 9 The results of SCA-βHC on 13-unit generator system with demand of 1800 MW using different parameter settings

Unit	Studying β				Studying *bw*					Studying R_β				
	SenExp1	SenExp2	SenExp3	SenExp4	SenExp5	SenExp6	SenExp7	SenExp8	SenExp9	SenExp10	SenExp11	SenExp12	SenExp13	SenExp14
u1	538.5587	628.3185	628.3186	448.9935	628.3185	628.3185	628.3185	628.3186	628.3184	538.2903	628.3186	628.3186	628.3182	628.3184
u2	299.2097	224.0583	149.5971	298.3129	224.6419	149.5996	149.5980	149.5971	149.5765	360.0000	224.3748	149.5971	149.6006	149.5996
u3	224.4900	149.0518	222.7559	299.8060	149.3148	222.8132	222.8006	222.7559	222.7934	301.3149	147.9813	222.7559	222.7578	222.7624
u4	60.0001	109.5513	60.0000	110.4166	109.5231	109.8644	109.8499	60.0000	109.8616	60.0000	109.8659	60.0000	109.8686	109.8658
u5	60.2851	109.7412	109.8664	109.4716	109.7848	109.8653	109.8601	109.8664	109.8664	60.0000	109.8680	109.8664	109.8658	109.8663
u6	159.7332	109.6630	109.8649	109.9188	109.6774	109.8660	60.0000	109.8649	60.0000	60.0000	109.8630	109.8649	109.8598	109.8658
u7	60.0067	60.0000	109.8652	61.0476	108.9989	109.8657	109.8483	109.8652	109.8614	60.0000	109.8652	109.8652	60.0000	60.0000
u8	109.9847	109.7695	109.8656	60.5422	109.6173	109.8074	109.8656	109.8656	109.8559	110.3949	109.8633	109.8656	109.8670	109.8658
u9	60.0000	109.8465	109.8663	111.3500	60.0000	60.0000	109.8589	109.8663	109.8665	60.0000	60.0000	109.8663	109.8622	109.8558
u10	77.4053	40.0000	40.0000	40.0000	40.1232	40.0000	40.0000	40.0000	40.0000	40.0000	40.0000	40.0000	40.0000	40.0000
u11	40.3264	40.0000	40.0000	40.0687	40.0000	40.0000	40.0000	40.0000	40.0000	40.0000	40.0000	40.0000	40.0000	40.0000
u12	55.0000	55.0000	55.0000	55.0000	55.0000	55.0000	55.0000	55.0000	55.0000	55.0000	55.0000	55.0000	55.0000	55.0000
u13	55.0000	55.0000	55.0000	55.0722	55.0000	55.0000	55.0000	55.0000	55.0000	55.0000	55.0000	55.0000	55.0000	55.0000
Total cost	17994.60	17963.46	**17960.39**	17999.93	17971.11	17960.56	17960.52	**17960.39**	17960.48	18097.94	17960.60	**17960.39**	17960.46	17960.41
Mean cost	18064.47	17973.92	17960.96	18017.58	17977.40	17967.44	17961.58	17960.96	17961.28	18163.24	17963.86	17960.96	17961.06	17960.69
Stdv.	3.11E+01	4.91E+00	5.45E-01	8.48E+00	3.26E+00	3.23E+00	9.95E-01	5.45E-01	6.40E-01	3.05E+01	2.55E+00	5.45E-01	5.05E-01	1.95E-01

Table 10 Comparison results on 13-unit generator system with demand of 1800 MW

Algorithm	Best cost	Mean cost
SCA-βHC	17960.39	17960.96
SCA	18097.94	18163.24
ABOMDE	17963.85	17967.36
βHC	17960.41	17988.60
β-GWO	17960.38	17960.63
DHS	**17960.37**	17961.12
FCASO-SQP	17964.08	18001.96
GA-PS-SQP	17964.00	18199.00
GWO	17972.94	NA
HCASO	17965.15	18022.04
HHS	17963.83	17972.48
HMAPSO	17969.31	17969.31
HQIPSO	17966.37	18081.05
HS2	17965.62	17986.56
IHS2	**17960.37**	17965.42
MPDE	**17960.37**	17960.37
MSL	18158.68	NA
NDS	17976.95	17976.95
NSABC	17972.96	18022.86
NSS	17976.95	17976.95
NUHS	**17960.37**	17987.10
QIPSO	17969.01	18,075.11
RCGA	17963.86	NA
SDP	17963.98	NA
THS	**17960.37**	17977.60

4.5 Case 5: 15-unit generator system with demand of 2630 MW

The ELD problem considered in this case study includes 15 generating units. The expected load demand of all units is 2630 MW, and the data of all generating units are taken from (Das and Suganthan 2010). The ramp rate limits and prohibited zones of the units were considered in this case study. Table 11 shows the results of using different parameter settings on the performance of proposed SCA-βHC algorithm. The parameters of the proposed SCA-βHC algorithm are studied using fourteen experimental scenarios, where the configurations of these scenarios are given in Table 2. The best solution with the minimum total fuel cost for each experimental scenario is highlighted in Table 11. In addition, the mean costs, the standard derivation (Stdv.), the total power output of the best solution, and the transmission loss (P_L) of the best solution are also provided in Table 11. The best results are highlighted using **bold** font.

The results of running SenExp1–SenExp4 are recorded in Table 11. These scenarios are designed to study the influence of the parameter β on the performance of the proposed SCA-βHC algorithm. From Table 11, it can be seen that the performance of the proposed SCA-βHC algorithm in SenExp3 performs is better than the other scenarios (SenExp1, SenExp2, and SenExp4) by achieving the minimum total fuel cost (32766.33 \$/h). Furthermore, the performance of SenExp1 is worse than the other scenarios. This is because the value of the parameter β is set to zero, and thus the source of the exploration in the proposed algorithm is neglected. Therefore, the parameter $\beta = 0.1$ is used in the following experiments.

Similarly, the results of studying the *bw* impact on the behaviour of the proposed SCA-βHC algorithm are recorded in Table 11. The five experimental scenarios SenExp5 to SenExp9 are designed for this purposes. From Table 11, it can be observed that the performance of the proposed SCA-βHC algorithm in SenExp6 is better than the other four scenarios (SenExp5, SenExp7–SenExp9) by getting the minimum total fuel cost (32761.56 \$/h). This is to prove that the 0.01 is the suitable value for the parameter *bw* which is able to provide the correct trade-off between the global and local search capabilities. In a nutshell, $bw = 0.01$ will be used in the following experiments in this case study.

The results of studying the effect of the parameter R_β on the behaviour of the proposed SCA-βHC algorithm are summarized in Table 11. The last five experimental scenarios (SenExp9–SenExp14) are designed for this purpose. From Table 11, it can be seen that the performance of the proposed SCA-βHC algorithm in SenExp12 perform better than the other four scenarios by achieving the minimum total fuel cost (32761.56 \$/h). On the other hand, the performance of SenExp14 is better than the other four scenarios by getting the minimum mean cost (32794.94 \$/h). However, the minimum total fuel cost will be used for comparison with the other methods in the literature, and thus leads us to use the results obtained by SenExp12. This is to prove that β=0.1, *bw*=0.01, and R_β=0.01 are the best parameter settings of the proposed algorithm which is able to strike the right balance between the exploration and exploitation, and thus navigate the search space of this version of the ELD problem to get better results.

Finally, the best results obtained by the proposed SCA-βHC algorithm against those of the other fourteen algorithm are recorded in Table 12. The results of the proposed SCA-βHC algorithm are taken from Table 11. It can be seen from the results provided in Table 12 that the CS and CLCS-CLM algorithms are able to obtain the same best results (32704.45\$/h). On the other hand, the performance of proposed SCA-βHC algorithm performs is better than the original version of the SCA algorithm, where the difference between the two algorithms is (474.85 \$/h). Furthermore, the performance of the proposed SCA-βHC algorithm is better than the performance of CPSO, HBF, HS, IHS, PSO, and PVHS algorithms. While the performance of the proposed

Table 11 The results of SCA-βHC on 15-unit generator system with demand of 2630 MW using different parameter settings

Unit	Studying β				Studying *bw*					Studying R_β				
	SenExp1	SenExp2	SenExp3	SenExp4	SenExp5	SenExp6	SenExp7	SenExp8	SenExp9	SenExp10	SenExp11	SenExp12	SenExp13	SenExp14
u1	445.1826	449.9004	455.0000	446.4519	447.4097	454.9992	455.0000	433.2206	455.0000	455.0000	455.0000	454.9992	454.6593	454.6675
u2	375.4861	379.9546	379.7787	375.8743	378.2698	379.9906	379.7787	377.8055	366.5078	365.2799	372.5244	379.9906	362.2041	356.5866
u3	127.7277	127.6655	127.7214	123.7210	128.2483	129.9949	127.7214	130.0000	130.0000	62.8301	128.3183	129.9949	130.0000	129.5685
u4	130.0000	130.0000	129.0245	115.3872	127.2547	129.9893	129.0245	129.4492	130.0000	111.1942	129.9600	129.9893	127.1825	129.9816
u5	167.4468	162.7793	160.0095	166.9796	169.5005	151.3573	160.0095	169.6129	168.2064	166.5309	150.0134	151.3573	167.2991	165.9637
u6	460.0000	458.4060	459.9993	457.8660	460.0000	455.7416	459.9993	459.9978	459.4582	460.0000	459.9899	455.7416	459.9812	459.9835
u7	426.0590	425.4470	423.1932	424.5730	426.6417	429.8619	423.1932	423.2444	417.5728	426.9211	427.4937	429.8619	427.1598	427.3491
u8	116.3425	113.3859	66.0569	105.8798	119.4139	126.1459	66.0569	96.3082	94.2809	136.4840	72.9246	126.1459	67.1613	65.2922
u9	60.0253	48.3453	79.9165	74.6181	42.8528	68.6827	79.9165	53.8901	58.9989	99.3562	71.8336	68.6827	78.4836	108.9350
u10	120.0490	120.0337	155.5481	157.2921	130.0218	110.4942	155.5481	139.8758	153.2265	160.0000	142.5523	110.4942	143.7631	112.9739
u11	77.4275	78.2169	77.2448	70.2027	70.0735	72.5522	77.2448	77.8587	76.0520	31.5722	71.1089	72.5522	68.2131	79.9728
u12	50.5781	79.3480	69.4126	77.5825	79.6873	79.7710	69.4126	79.5200	72.8365	80.0000	77.5664	79.7710	80.0000	79.4798
u13	26.7817	44.3706	41.4438	29.7888	26.0035	31.8292	41.4438	46.6143	27.2363	42.3020	36.6584	31.8292	31.6839	29.2915
u14	43.7330	15.8054	19.4431	19.9372	21.6764	17.1081	19.4431	15.5516	26.4756	49.1337	32.4223	17.1081	30.6809	39.0515
u15	34.3415	25.5329	16.2934	15.8340	33.0094	21.9895	16.2934	26.4062	24.6180	20.0995	30.9564	21.9895	30.9021	20.0958
Total cost	32820.90	32770.70	32766.33	32815.46	32763.92	32761.56	32766.33	32781.62	32765.17	33236.41	32787.96	**32761.56**	32777.04	32769.42
Mean cost	32871.64	32798.10	32854.53	32861.37	32800.63	32799.42	32854.53	32804.91	32797.57	34690.21	32824.13	32799.42	32795.50	32794.94
Stdv.	2.72E+01	1.27E+01	3.97E+01	1.72E+01	1.25E+01	1.57E+01	3.97E+01	1.11E+01	1.70E+01	1.61E+03	1.86E+01	1.57E+01	1.31E+01	1.28E+01
Output	2661.18	2659.19	2660.09	2661.99	2660.06	2660.51	2660.09	2659.36	2660.47	2666.70	2659.32	2660.51	2659.37	2659.19
P_L	31.1808	29.1914	30.0858	31.9948	30.0633	30.5077	30.0858	29.3552	30.4699	36.5587	29.3226	30.5077	29.3740	29.1930

Table 12 Comparison results on 15-unit generator system with demand of 2630 MW

Algorithm	Best cost	Mean cost	P_L	Output
SCA-βHC	32761.56	32799.42	30.5077	2660.51
SCA	33236.41	34690.21	36.5587	2666.70
CLCS-CLM	**32704.45**	32704.45	30.6614	2660.66
CPSO	32835.00	33021.00	32.1302	2662.10
CS	**32704.45**	32704.75	30.6614	2660.66
CS-CLM	**32704.45**	32704.45	30.6614	2660.66
GAPSO	32724.00	32984.00	31.7500	2661.75
HBF	32784.50	32976.81	28.9470	2658.95
HS1	32813.34	32910.65	30.6852	2660.66
IHS1	32830.34	32925.26	30.5336	2660.55
IPSO	32709.00	32784.50	30.8580	2660.80
JAYA	32712.65	32743.46	31.1936	2661.20
PSO	32858.00	33039.00	32.4306	2662.40
PVHS	32780.00	32892.46	32.1953	2662.28
SOH-PSO	32751.39	32878.00	32.2800	2662.29

SCA-βHC algorithm are competitive with other comparative algorithms.

4.6 Case 6: 40-unit generator system with demand of 10,500 MW

The system considered in this case study consist of 40 generating units with valve point effects. Furthermore, the transmission loss is neglected, while the expected power load demand is 10,500 MW. The data of generating units of this system were taken from (Sinha et al. 2003). Table 13 illustrates the results obtained by studying the impact of the different parameters settings on the performance of the proposed SCA-βHC algorithm. In this table, the best solution achieved by running each experimental scenario over 30 replicated runs is recorded. The total fuel cost of the best solution, the mean costs, and the standard derivations are also provided in Table 13. The parameter settings of each experimental scenario is provided in Table 2. The best fuel cost is highlighted using **bold** font.

The parameter β is studied using four various values in four experimental scenarios (SenExp1 (β=0), SenExp2 (β=0.01), SenExp3 (β=0.1), and SenExp4 (β=0.5)). The results of running these scenarios over 30 times are summarized in Table 13. Clearly, it can be seen that the higher value of β leads to worse results as seen in SenExp4, as well as ignoring the parameter β leads to worse results as illustrates in SenExp1. However, the performance of the proposed SCA-βHC algorithm in SenExp2 is better than the performance of the other three scenarios (SenExp1, SenExp3, and SenExp4) by obtaining the minimum total fuel cost (121414.70 $/h). On the other hand, it can be seen from the results in Table 13 that the SenExp2 is able to obtain the minimum mean of costs in comparison with other three scenarios. This is prove that the value of 0.01 of the parameter β is the threshold value able to make a balance between exploration and exploitation for this case of ELD problem. Therefore, value of the β = 0.01 will be used in the following experiments.

The five experimental scenarios (SenExp5 (*bw*=0.001), SenExp6 (*bw*=0.01), SenExp7 (*bw*=0.1), SenExp8 (*bw*=0.5), and SenExp9 (*bw*=0.95)) are designed to study the effect of the parameter *bw* on the performance of the proposed SCA-βHC algorithm. Table 13 shows the results of running these scenarios. It can be observed from the results in this table that the higher values of the parameter *bw* leads to better results, while SenExp8 obtained the minimum total fuel cost (121414.70 $/h) in comparison with other four scenarios (SenExp5, SenExp6, SenExp7, and SenExp9). On the other hand, the SenExp9 produce the best results in terms of the mean cost (121439.12 $/h). In the following experiments, *bw* = 0.5 is used since it achieves the minimum fuel cost as can be recognized from SenExp8.

Table 13 illustrates the results of running the five experimental scenarios (i.e., SenExp10–SenExp14) which are designed to study the impact of the parameter R_β on the behaviour of the proposed SCA-βHC algorithm. From Table 13, it can be shown that the results of the proposed SCA-βHC algorithm are enhanced gradually by increasing the value of R_β based on the mean of costs. Furthermore, the performance of the proposed SCA-βHC algorithm in SenExp10 is the worse among the other five scenarios (SenExp10–SenExp14) by getting the worst results. This is because that the value of the R_β is equal to zero, and thus the β-hill climbing optimizer is not functioned. On the other hand, SenExp12 is better than other scenarios by obtaining the minimum total fuel cost for this case of ELD problem. Based on above discussions, the settings of the three parameters of the proposed SCA-βHC are β=0.01, *bw*=0.5, and R_β =0.01 which are yielded the best results.

Finally, the best results of the proposed SCA-βHC algorithm against those of the other forty seven comparative methods are recorded in Table 14. From Table 14, it can be seen that the performance of the HGWO algorithm outperforms the proposed SCA-βHC algorithm, as well as other competitive methods by obtaining the minimum fuel cost (121412.00 $/h). Furthermore, the proposed SCA-βHC algorithm is ranked in the ninth position based on the results obtained (121414.70 $/h), where the difference between the results of the proposed algorithms and the results of HGWO is equal to 2.70 $/h. On the other hand, the performance of the proposed algorithm outperforms the original sine cosine algorithm (SCA), where the difference between the two algorithms is equal to 3870.79 $/h.

Table 13 The results of SCA-βHC on 40-unit generator system with demand of 10,500 MW using different parameter settings

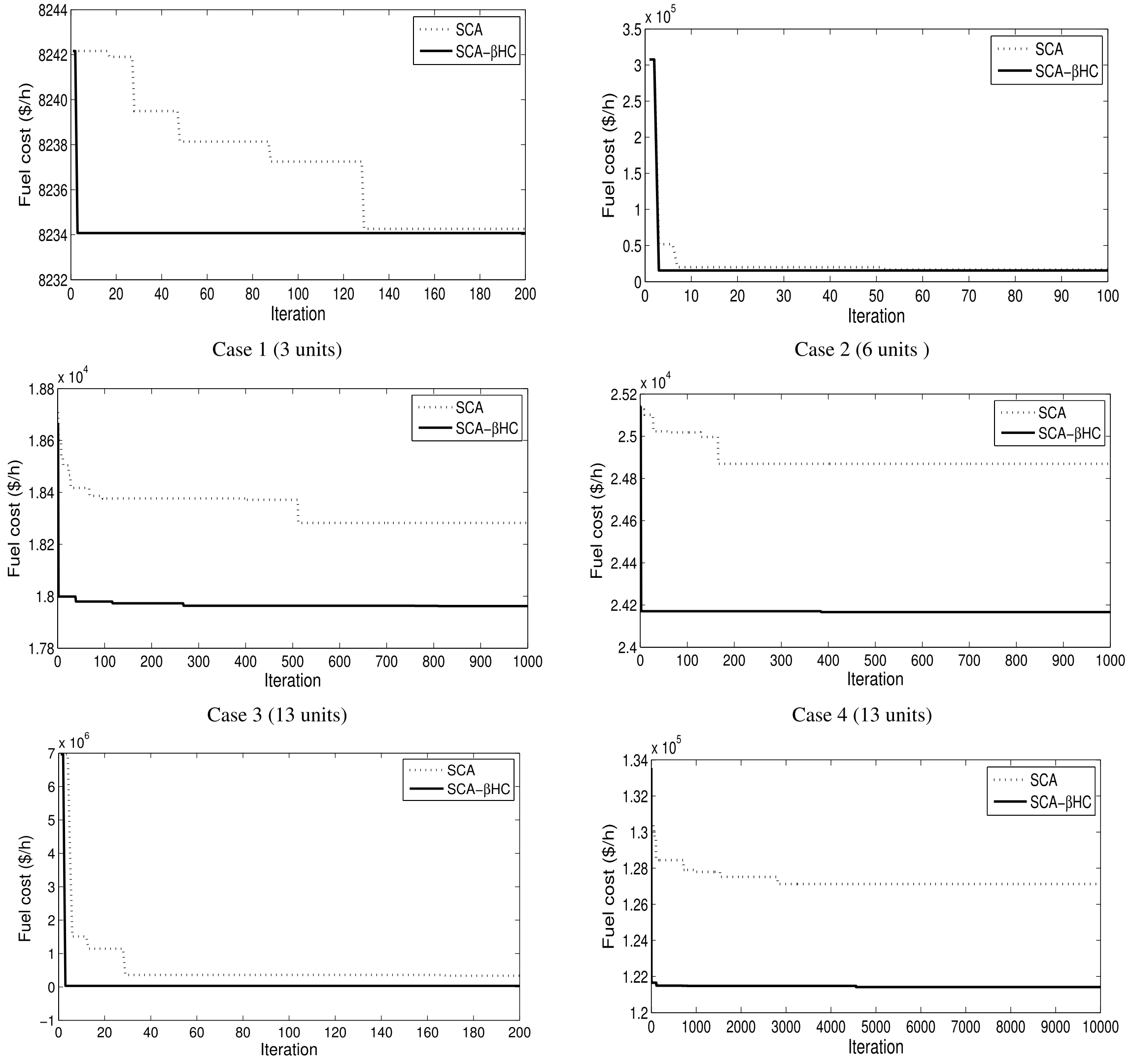
Unit	Studying β				Studying *bw*					Studying R_β				
	SenExp1	SenExp2	SenExp3	SenExp4	SenExp5	SenExp6	SenExp7	SenExp8	SenExp9	SenExp10	SenExp11	SenExp12	SenExp13	SenExp14
u1	110.8039	110.8000	110.8185	114.0000	110.7998	110.7998	110.8062	110.8000	110.8024	114.0000	110.8080	110.8000	110.8007	110.8059
u2	110.8096	110.8048	110.8146	114.0000	110.9822	113.8418	110.8051	110.8048	110.8021	114.0000	110.8137	110.8048	110.8043	110.8064
u3	120.0000	97.4000	97.3975	120.0000	97.3999	97.3999	97.4000	97.4000	97.4000	120.0000	97.4043	97.4000	97.4001	97.4000
u4	179.7332	179.7337	179.7337	190.0000	179.7331	179.7331	179.7330	179.7337	179.7334	190.0000	179.7331	179.7337	179.7331	179.7332
u5	95.0069	92.7210	88.3510	97.0000	90.7034	94.9229	91.8350	92.7210	87.8848	97.0000	92.3983	92.7210	92.6166	92.5946
u6	140.0000	140.0000	140.0000	140.0000	140.0000	140.0000	140.0000	140.0000	140.0000	140.0000	140.0000	140.0000	140.0000	140.0000
u7	259.5996	259.6010	259.6159	300.0000	259.5997	259.5997	259.5997	259.6010	259.6005	300.0000	259.5998	259.6010	259.5997	259.5999
u8	284.5999	284.5992	284.6239	300.0000	284.5996	284.5996	284.5996	284.5992	284.5997	300.0000	284.5997	284.5992	284.5998	284.5996
u9	284.5995	284.5998	284.6259	300.0000	284.5997	284.5997	284.5996	284.5998	284.6008	300.0000	284.5997	284.5998	284.6002	284.6003
u10	130.0000	130.0000	130.0000	241.3571	130.0000	130.0000	130.0000	130.0000	130.0000	232.7067	130.0000	130.0000	130.0000	130.0000
u11	168.7979	168.8007	168.7891	95.7957	168.7998	94.0000	168.8002	168.8007	168.7986	171.4390	168.8001	168.8007	168.7997	168.7997
u12	94.0000	168.7994	168.8314	94.1297	168.7998	168.7997	168.7996	168.7994	168.8000	94.0000	168.8000	168.7994	168.7998	168.7999
u13	484.0387	214.7597	214.7544	134.4110	214.7598	214.7598	214.7600	214.7597	214.7592	125.0000	214.7598	214.7597	214.7602	214.7606
u14	484.0356	394.2797	304.5213	385.0533	394.2794	303.9888	304.5193	394.2797	394.2792	125.0000	394.2796	394.2797	394.2788	394.2798
u15	484.0392	304.5196	394.2735	375.2145	394.2794	394.2788	394.2795	304.5196	304.5188	500.0000	394.2794	304.5196	394.2794	304.5196
u16	214.7584	394.2791	394.2749	284.3197	304.5196	394.2794	394.2793	394.2791	394.2794	125.0000	304.5195	394.2791	304.5198	394.2795
u17	489.2792	489.2799	489.2805	498.2863	489.2794	489.2793	489.2795	489.2799	489.2793	500.0000	489.2782	489.2799	489.2797	489.2791
u18	489.2824	489.2792	489.2847	495.2856	489.2794	489.2794	489.2793	489.2792	489.2802	500.0000	489.2793	489.2792	489.2793	489.2811
u19	511.2815	511.2809	511.2834	515.7878	511.2794	511.2793	511.2796	511.2809	511.2793	550.0000	511.2795	511.2809	511.2797	511.2793
u20	242.0000	511.2797	511.2784	514.0600	511.2792	511.2793	511.2795	511.2797	511.2797	550.0000	511.2798	511.2797	511.2795	511.2797
u21	523.2797	523.2807	523.2861	539.7477	523.3139	523.2794	523.2796	523.2807	523.2795	550.0000	523.2798	523.2807	523.2796	523.2794
u22	523.2795	523.2794	523.3169	537.1717	523.2794	523.2794	523.2793	523.2794	523.2801	550.0000	523.2801	523.2794	523.2792	523.2795
u23	523.2792	523.2795	523.2876	523.5460	523.2794	523.2794	523.2794	523.2795	523.2793	550.0000	523.2793	523.2795	523.2792	523.2796
u24	523.2791	523.2798	523.3054	522.3354	523.2794	523.2794	523.2793	523.2798	523.2809	550.0000	523.2794	523.2798	523.2795	523.2795
u25	523.2794	523.2797	523.2810	535.8016	523.2794	523.2793	523.2796	523.2797	523.2801	550.0000	523.2794	523.2797	523.2794	523.2800
u26	523.2797	523.2796	523.2949	531.2837	523.2794	523.2794	523.2796	523.2796	523.2794	550.0000	523.2795	523.2796	523.2795	523.2793
u27	10.0000	10.0000	10.0000	11.7667	10.0000	10.0000	10.0000	10.0000	10.0000	10.0000	10.0000	10.0000	10.0000	10.0000
u28	10.0000	10.0000	10.0000	15.7150	10.0000	10.0000	10.0000	10.0000	10.0000	13.3467	10.0000	10.0000	10.0000	10.0000
u29	10.0000	10.0000	10.0000	19.9302	10.0000	10.0000	10.0000	10.0000	10.0000	10.0000	10.0000	10.0000	10.0000	10.0000
u30	94.1046	87.8206	91.9519	88.8402	89.7990	87.8000	88.7064	87.8206	92.6632	50.0437	88.1309	87.8206	87.9303	87.9444
u31	162.4511	190.0000	190.0000	188.6722	189.8664	190.0000	190.0000	190.0000	190.0000	142.9124	190.0000	190.0000	190.0000	190.0000
u32	116.2677	190.0000	190.0000	166.7910	189.9726	189.7245	190.0000	190.0000	190.0000	190.0000	190.0000	190.0000	190.0000	190.0000
u33	190.0000	190.0000	190.0000	179.3273	190.0000	190.0000	190.0000	190.0000	190.0000	190.0000	190.0000	190.0000	190.0000	190.0000
u34	164.8026	164.8012	164.8202	191.5209	164.7998	200.0000	164.8015	164.8012	164.8001	200.0000	164.7998	164.8012	164.8022	164.8002
u35	200.0000	164.8017	164.8211	179.1873	164.7998	200.0000	164.8001	164.8017	164.7999	200.0000	164.8003	164.8017	164.8005	164.8002
u36	200.0000	164.8001	164.8012	185.6193	164.7999	164.7998	164.8011	164.8001	164.8002	200.0000	164.8002	164.8001	164.8003	164.8001
u37	110.0000	110.0000	110.0000	93.6139	110.0000	110.0000	110.0000	110.0000	110.0000	98.4056	110.0000	110.0000	110.0000	110.0000
u38	110.0000	110.0000	110.0000	58.9672	110.0000	110.0000	110.0000	110.0000	110.0000	87.1460	110.0000	110.0000	110.0000	110.0000
u39	94.7514	110.0000	110.0000	102.8143	110.0000	110.0000	110.0000	110.0000	110.0000	110.0000	110.0000	110.0000	110.0000	110.0000
u40	511.2804	511.2803	511.2812	518.6479	511.2794	511.2792	511.2801	511.2803	511.2799	550.0000	511.2795	511.2803	511.2799	511.2795

Table 13 (continued)

Unit	Studying β				Studying *bw*					Studying R_β				
	SenExp1	SenExp2	SenExp3	SenExp4	SenExp5	SenExp6	SenExp7	SenExp8	SenExp9	SenExp10	SenExp11	SenExp12	SenExp13	SenExp14
Total cost	123766.60	121414.70	121416.35	123929.90	121417.47	121449.90	121415.29	**121414.70**	121414.71	125285.48	121414.85	**121414.70**	121414.73	121414.75
Mean cost	124799.98	121444.94	121421.85	124247.81	121466.61	121469.97	121448.68	121444.94	121439.12	126144.35	121456.61	121444.94	121423.17	121423.01
Stdv.	4.31E+02	1.92E+01	5.81E+00	1.49E+02	1.79E+01	7.80E+00	1.86E+01	1.92E+01	2.26E+01	3.21E+02	2.20E+01	1.92E+01	1.07E+01	9.51E+00

Table 14 Comparison results on 40-unit generator system with demand of 10,500 MW

Algorithm	Best cost	Mean cost
SCA-βHC	121414.70	121444.94
SCA	125285.48	126144.35
ABC	121441.03	121995.82
ABOMDE	121414.87	121487.85
ACO	121811.37	121930.58
ARCGA	121415.50	121462.15
BGO	121479.50	121512.06
βHC	121414.68	121496.84
β-GWO	121415.09	121417.79
CBPSO-RVM	121555.32	122281.14
CLCS-CLM	121412.54	121412.99
CS	121457.16	121512.30
CS-CLM	121450.41	121503.55
CSOMA	121414.70	121415.05
DE-BBO	121420.89	121420.90
DVL-MILP	121413.00	NA
FAPSO	121712.40	121778.25
FAPSO-NM	121418.30	121418.80
FCASO-SQP	121456.98	122026.21
FFA	121415.05	121416.57
GA-PS-SQP	121458.00	122039.00
GSO	124265.40	124609.18
GWO	121488.40	NA
HCA	121414.70	121415.05
HCASO	121865.63	122100.74
HCPSO	121865.23	122100.87
HCPSO-SQP	121458.54	122028.16
HGWO	**121412.00**	121419.00
HHS	121415.59	121615.85
HMAPSO	121586.90	121586.90
HQIPSO	121418.60	121427.47
MDE	121414.79	121418.44
MPDE	121412.54	121412.62
NDS	121647.40	121647.40
NSABC	122270.91	123675.40
NSS	122186.90	0.00
NUHS	121412.74	121549.95
PSO-MSAF	121423.23	NA
QGSO	124265.40	124609.18
QIPSO	121448.21	122225.07
RCGA	121418.72	NA
SDP	121412.54	NA
SOMA	121418.79	121449.88
SSA	121412.35	121412.97
THS	121425.15	121528.65
TLA	122009.77	122074.90
TS1	122288.38	122424.81
TSARGA	121463.07	122928.31



Case 1 (3 units)

Case 2 (6 units)

Case 3 (13 units)

Case 4 (13 units)

Case 5 (15 units)

Case 6 (40 units)

Fig. 4 The convergence plots of the proposed SCA-βHC algorithm for different ELD problems

4.7 Convergence analysis

In order to study the hit rate of using heuristic search (i.e., β HC) in SCA, the hit rate parameter can be considered as R_β. Note that this parameter determines the percentage of hitting the βHC by SCA. This parameter is studied with different values as shown in Table 2. The results obtained by the different values of R_β can be shown in Tables 3, 5, 7, 9, 11, and 13. As can be aware from the results, when the value of R_β is small, the best results will be better. Therefore, when the number of hit rate of βHC from SCA is small, the behavior of the proposed method tends to be better due to its ability in achieving the right balance in exploration and exploitation during the search.

Figure 4 plots the convergence behaviour of the proposed SCA-βHC algorithm against the original version of sine cosine algorithm (SCA) on the different cases of ELD problems. It should be noted that the *x-axis* represent the number of iterations, while the *y-axis* represents the total fuel cost ($/h). The number of iterations are chosen carefully in order to visualize the differences among the results of the two algorithms. The parameter settings of the SCA-βHC algorithm are the setting of the experimental scenario obtained the minimum fuel cost for each ELD problem. From Fig. 4, it can be seen that the SCA-βHC algorithm obtains near results to the optimal in the early stages of the search process. Furthermore, the figure reveals big differences between the results of the SCA-βHC algorithm and those of the SCA algorithm especially for Case 3, Case 4, Case 5 and Case 6.

5 Conclusion and future work

In this paper, the economic load dispatch (ELD) problems are tackled using a hybridized version of Sine Cosine Algorithm with β-hill climbing (βHS) optimizer using six real-wold problems of different number of generating units. ELD is tackled by achieving the total load demand generated by power units with minimum fuel costs with respect to several types of constraints such as equality constraints, inequality constraints, ramp rate limits, and prohibited operating zones. Although this problem is solved using different optimization algorithm, population-based algorithm has revealed very successful outcomes. Sine cosine algorithm (SCA) is the recent population-based optimization algorithm utilized the survival-of-the-fittest principle using Sine and Cosine trigonometric functions. In order to efficiently navigate the ELD search space and achieve the right balance between the exploration and exploitation, βHC, the recent local search algorithm, is hybridized with SCA as a new operator to help in exploiting the search space regions to which SCA explored.

The proposed algorithm is abbreviated as SCA-βHC which is evaluated using six real-world cases: 3-unit generator with 850 MW, 6-unit generator with 1263 MW, 13-unit generator with 2520 MW, 13-unit generator with 1800 MW, 15-unit generator with 2630 MW, and 40-unit generator with 10500 MW. The effect of the control parameters (i.e., β, bw, R_β) on the convergence behaviour of the proposed SCA-βHC is studied through 14 convergence scenarios with different values. In conclusion, SCA-βHC has the best performance when the β, bw, and R_β have small values. For comparative evaluation, the results produced by each ELD case is compared against several state-of-art methods. In a nutshell, the proposed SCA-βHC is able to yield the best overall results for 3-unit generator with 850 MW. Furthermore, it is able to produce the second-best results for 13-unit generator with 2520 MW, and the third-best results for 13-unit generator with 1800 MW. It occupies the 6th position out of 15 comparative methods for 6-unit generation case. It is also able to achieve the seventh position out of 15 comparative methods for 15-unit generator with 850 MW and the ninth position out of 48 comparative methods for 40-unit generator with 10500 MW. This reveal the viability of the proposed SCA-βHC for ELD domain and power systems in general.

As the proposed SCA-βHC reveals very promising results for ELD problem, in the future, it can be utilized for other developments such as:

1. Tackling other Engineering optimization Problems with multi-objective features such as truss structure.
2. Investigating other local search algorithms in SCA such as Simulated Annealing, Tabu Search, VNS, Iterated Local Search, and others
3. Adapting SCA-βHC for other versions of ELD problems with more challenging constraints to be more realistic such as the transmission losses and multiple fuels constraints .

References

Abed-alguni BH, Alkhateeb F (2020) Intelligent hybrid cuckoo search and $\beta $-hill climbing algorithm. J King Saud Univ Comput Inf Sci 32(2):159–173

Abo-Elnaga Y, El-Shorbagy M (2020) Multi-sine cosine algorithm for solving nonlinear bilevel programming problems. Int J Comput Intell Syst 13(1):421–432

Abualigah LM, Khader AT, Al-Betar MA (2017a) $\beta $-hill climbing technique for the text document clustering. New trends in information technology NTIT2017 conference. Amman, Jordan, IEEE, pp. 60–66

Abualigah LM, Khadery AT, Al-Betar MA, Alyasseri ZAA, Alomari OA, Hanandehk ES (2017b) Feature selection with $\beta $-hill climbing search for text clustering application. Second Palestinian International conference on information and communication technology (PICICT 2017). Gaza, Palestine, IEEE, pp. 22–27

Alawode K, Jubril A, Kehinde L, Ogunbona P (2018) Semidefinite programming solution of economic dispatch problem with non-smooth, non-convex cost functions. Electr Power Syst Res 164:178–187

Al-Betar MA (2017) $\beta $-hill climbing: an exploratory local search. Neural Comput Appl 28(1):153–168

Al-Betar MA (2021) A $\beta $-hill climbing optimizer for examination timetabling problem. J Ambient Intell Hum Comput 12:653–666

Al-Betar MA, Khader AT, Doush IA (2014) Memetic techniques for examination timetabling. Ann Oper Res 218(1):23–50

Al-Betar MA, Awadallah MA, Khader AT, Bolaji AL (2016) Tournament-based harmony search algorithm for non-convex economic load dispatch problem. Appl Soft Comput 47:449–459

Al-Betar MA, Awadallah MA, Bolaji AL, Alijla BO (2017) $\beta $-hill climbing algorithm for sudoku game. Second Palestinian international conference on information and communication technology (PICICT 2017). Gaza, Palestine, IEEE, pp. 84–88

Al-Betar MA, Awadallah MA, Doush IA, Alsukhni E, ALkhraisat H (2018a) A non-convex economic dispatch problem with valve loading effect using a new modified $\beta $-hill climbing local search algorithm. Arab J Sci Eng 43(12):7439–7456

Al-Betar MA, Awadallah MA, Khader AT, Bolaji AL, Almomani A (2018b) Economic load dispatch problems with valve-point loading using natural updated harmony search. Neural Comput Appl 29(10):767–781

Al-Betar MA, Alyasseri ZAA, Awadallah MA, Doush IA (2020a) Coronavirus herd immunity optimizer (chio). Neural Comput Appl. https://doi.org/10.1007/s00521-020-05296-6

Al-Betar MA, Awadallah MA, Krishan MM (2020b) A non-convex economic load dispatch problem with valve loading effect using a hybrid grey wolf optimizer. Neural Comput Appl 32:12127–12154

Alomari OA, Khader AT, Al-Betar MA, Alyasseri ZAA (2018a) A hybrid filter-wrapper gene selection method for cancer classification. 2018 2nd International conference on biosignal analysis. processing and systems (ICBAPS), IEEE, pp. 113–118

Alomari OA, Khader AT, Al-Betar MA, Awadallah MA (2018b) A novel gene selection method using modified MRMR and hybrid bat-inspired algorithm with $\beta $-hill climbing. Appl Intell 48(11):4429–4447

Alsukni E, Arabeyyat OS, Awadallah MA, Alsamarraie L, Abu-Doush I, Al-Betar MA (2019) Multiple-reservoir scheduling using $\beta $-hill climbing algorithm. J Intell Syst 28(4):559–570

Alsumait J, Sykulski J, Al-Othman A (2010) A hybrid ga-ps-sqp method to solve power system valve-point economic dispatch problems. Appl Energy 87(5):1773–1781

Alweshah M, Al-Daradkeh A, Al-Betar MA, Almomani A, Oqeili S (2020) $\beta $-hill climbing algorithm with probabilistic neural network for classification problems. J Ambient Intell Hum Comput 11:3405–3416

Alyasseri ZAA, Khader AT, Al-Betar MA (2017) Optimal EEG signals denoising using hybrid $\beta $-hill climbing algorithm and wavelet transform. ICISPC '17. Penang, Malaysia, ACM, pp. 5–11

Alyasseri ZAA, Khader AT, Al-Betar MA, Awadallah MA (2018) Hybridizing $\beta $-hill climbing with wavelet transform for denoising ECG signals. Inf Sci 429:229–246

Alzaidi AA, Ahmad M, Doja MN, Al Solami E, Beg MS (2018) A new 1d chaotic map and $beta$-hill climbing for generating substitution-boxes. IEEE Access 6:55405–55418

Amjady N, Sharifzadeh H (2010) Solution of non-convex economic dispatch problem considering valve loading effect by a new modified differential evolution algorithm. Int J Electr Power Energy Syst 32(8):893–903

Askarzadeh A (2016) A novel metaheuristic method for solving constrained engineering optimization problems: crow search algorithm. Comput Struct 169:1–12

Attia AF, El Sehiemy RA, Hasanien HM (2018) Optimal power flow solution in power systems using a novel sine-cosine algorithm. Int J Electr Power Energy Syst 99:331–343

Awadallah MA, Al-Betar MA, Bolaji AL, Alsukhni EM, Al-Zoubi H (2019) Natural selection methods for artificial bee colony with new versions of onlooker bee. Soft Comput 23(15):6455–6494

Azizipanah-Abarghooee R, Niknam T, Roosta A, Malekpour AR, Zare M (2012) Probabilistic multiobjective wind-thermal economic emission dispatch based on point estimated method. Energy 37(1):322–335

Babar MI, Ahmad A, Fayyaz S (2020) A hybrid sine cosine algorithm with sqp for solving convex and nonconvex economic dispatch problem. Mehran Univ Res J Eng Technol 39(1):31–46

Basu M, Chowdhury A (2013) Cuckoo search algorithm for economic dispatch. Energy 60:99–108

Belazzoug M, Touahria M, Nouioua F, Brahimi M (2019) An improved sine cosine algorithm to select features for text categorization. J King Saud Univ Comput Inf Sci 32(4):454–464

Bhattacharya A, Chattopadhyay PK (2010a) Biogeography-based optimization for different economic load dispatch problems. Power Syst IEEE Trans 25(2):1064–1077

Bhattacharya A, Chattopadhyay PK (2010b) Hybrid differential evolution with biogeography-based optimization for solution of economic load dispatch. Power Syst IEEE Trans 25(4):1955–1964

Bhattacharya A, Chattopadhyay PK (2010c) Solving complex economic load dispatch problems using biogeography-based optimization. Expert Syst Appl 37(5):3605–3615

Blum C, Roli A (2003) Metaheuristics in combinatorial optimization: overview and conceptual comparison. ACM Comput Surv (CSUR) 35(3):268–308

Blum C, Puchinger J, Raidl GR, Roli A (2011) Hybrid metaheuristics in combinatorial optimization: a survey. Appl Soft Comput 11(6):4135–4151

BoussaïD I, Lepagnot J, Siarry P (2013) A survey on optimization metaheuristics. Inf Sci 237:82–117

Cai J, Li Q, Li L, Peng H, Yang Y (2012a) A hybrid cpso-sqp method for economic dispatch considering the valve-point effects. Energy Convers Manag 53(1):175–181

Cai J, Li Q, Li L, Peng H, Yang Y (2012b) A hybrid fcaso-sqp method for solving the economic dispatch problems with valve-point effects. Energy 38(1):346–353

Chakraborty S, Senjyu T, Yona A, Saber A, Funabashi T (2011) Solving economic load dispatch problem with valve-point effects using a hybrid quantum mechanics inspired particle swarm optimisation. Gener Transm Distrib IET 5(10):1042–1052

Chaturvedi KT, Pandit M, Srivastava L (2008) Self-organizing hierarchical particle swarm optimization for nonconvex economic dispatch. IEEE Trans Power Syst 23(3):1079–1087

Chegini SN, Bagheri A, Najafi F (2018) Psoscalf: a new hybrid pso based on sine cosine algorithm and levy flight for solving optimization problems. Appl Soft Comput 73:697–726

Chen G, Ding X (2015) Optimal economic dispatch with valve loading effect using self-adaptive firefly algorithm. Appl Intell 42(2):276–288

Chen H, Jiao S, Heidari AA, Wang M, Chen X, Zhao X (2019) An opposition-based sine cosine approach with local search for parameter estimation of photovoltaic models. Energy Convers Manag 195:927–942

Chen H, Heidari AA, Zhao X, Zhang L, Chen H (2020) Advanced orthogonal learning-driven multi-swarm sine cosine optimization: framework and case studies. Expert Syst Appl 144:113113

Coelho LS, Mariani VC (2006) Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect. IEEE Trans Power Syst 21(2):989–996

Coelho LDS, Mariani VC (2009) An improved harmony search algorithm for power economic load dispatch. Energy Convers Manag 50(10):2522–2526

Coelho LDS, Mariani VC (2010) An efficient cultural self-organizing migrating strategy for economic dispatch optimization with valve-point effect. Energy Convers Manag 51(12):2580–2587

Dai C, Zhu Y, Chen W (2006) Seeker optimization algorithm. In: International conference on computational and information science, Springer, pp. 167–176

Das S, Suganthan PN (2010) Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems. Jadavpur University, Nanyang Technological University, Kolkata, pp 341–359

Das S, Bhattacharya A, Chakraborty AK (2018) Solution of short-term hydrothermal scheduling using sine cosine algorithm. Soft Comput 22(19):6409–6427

dos Santos CL, Mariani VC (2010) An efficient cultural self-organizing migrating strategy for economic dispatch optimization with valve-point effect. Energy Convers Manag 51(12):2580–2587

Elaiw A, Xia X, Shehata A (2013) Hybrid de-sqp and hybrid pso-sqp methods for solving dynamic economic emission dispatch problem with valve-point effects. Electr Power Syst Res 103:192–200

Elattar EE (2015) A hybrid genetic algorithm and bacterial foraging approach for dynamic economic dispatch problem. Int J Electr Power Energy Syst 69:18–26

El-Keib A, Ma H, Hart J (1994) Environmentally constrained economic dispatch using the Lagrangian relaxation method. Power Syst IEEE Trans 9(4):1723–1729

Elsakaan AA, El-Sehiemy RA, Kaddah SS, Elsaid MI (2018) An enhanced moth-flame optimizer for solving non-smooth economic dispatch problems with emissions. Energy 157:1063–1078

Fausto F, Reyna-Orta A, Cuevas E, Andrade AG, Perez-Cisneros M (2020) From ants to whales: metaheuristics for all tastes. Artif Intell Rev 53(1):753–810

Fraga ES, Yang L, Papageorgiou LG (2012) On the modelling of valve point loadings for power electricity dispatch. Appl Energy 91(1):301–303

Gholizadeh S, Sojoudizadeh R (2019) Modified sine-cosine algorithm for sizing optimization of truss structures with discrete design variables. Iran Univ Sci Technol 9(2):195–212

Ghosh KK, Ahmed S, Singh PK, Geem ZW, Sarkar R (2020) Improved binary sailfish optimizer based on adaptive $\beta $-hill climbing for feature selection. IEEE Access 8:83548–83560

Gupta S, Deep K (2019a) A hybrid self-adaptive sine cosine algorithm with opposition based learning. Expert Syst Appl 119:210–230

Gupta S, Deep K (2019b) Improved sine cosine algorithm with crossover scheme for global optimization. Knowl Based Syst 165:374–406

Gupta S, Deep K (2020) A novel hybrid sine cosine algorithm for global optimization and its application to train multilayer perceptrons. Appl Intell 50(4):993–1026

Hamouda E, Abohamama AS (2020) Wireless sensor nodes localiser based on sine-cosine algorithm. IET Wirel Sens Syst 10(4):145–153

Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H (2019) Harris hawks optimization: algorithm and applications. Future Gener Comput Syst 97:849–872

Hekimoğlu B (2019) Sine-cosine algorithm-based optimization for automatic voltage regulator system. Trans Inst Meas Control 41(6):1761–1771

Hemamalini S, Simon SP (2010) Artificial bee colony algorithm for economic load dispatch problem with non-smooth cost functions. Electr Power Compon Syst 38(7):786–803

Huang Z, Zhao J, Qi L, Gao Z, Duan H (2020) Comprehensive learning cuckoo search with chaos-lambda method for solving economic dispatch problems. Appl Intell 50:2779–2799

Jadoun VK, Pandey VC, Gupta N, Niazi KR, Swarnkar A (2018) Integration of renewable energy sources in dynamic economic load dispatch problem using an improved fireworks algorithm. IET Renew Power Gener 12(9):1004–1011

Jain N, Nangia U, Jain J (2018) Economic load dispatch using adaptive social acceleration constant based particle swarm optimization. J Inst Eng (India) Ser B 99(5):431–439

James J, Li VO (2016) A social spider algorithm for solving the non-convex economic load dispatch problem. Neurocomputing 171:955–965

Jayabarathi T, Raghunathan T, Adarsh B, Suganthan PN (2016) Economic dispatch using hybrid grey wolf optimizer. Energy 111:630–641

Kamboj VK, Bhadoria A, Bath S (2017) Solution of non-convex economic load dispatch problem for small-scale power systems using ant lion optimizer. Neural Comput Appl 28(8):2181–2192

Khamsawang S, Jiriwibhakorn S (2010) Dspso-tsa for economic dispatch problem with nonsmooth and noncontinuous cost functions. Energy Convers Manag 51(2):365–375

Kumar M, Dhillon J (2018) Hybrid artificial algae algorithm for economic load dispatch. Appl Soft Comput 71:89–109

Kumar R, Sharma D, Sadu A (2011) A hybrid multi-agent based particle swarm optimization algorithm for economic power dispatch. Int J Electr Power Energy Syst 33(1):115–123

Le Yang BL, He D (2020) A selection hyper-heuristic algorithm for multiobjective dynamic economic and environmental load dispatch. Complexity 2020:18

Li S, Fang H, Liu X (2018) Parameter optimization of support vector regression based on sine cosine algorithm. Expert Syst Appl 91:63–77

Li X, Zhang H, Lu Z (2019) A differential evolution algorithm based on multi-population for economic dispatch problems with valve-point effects. IEEE Access 7:95585–95609

Lin WM, Cheng FS, Tsay MT (2002) An improved tabu search for economic dispatch with multiple minima. IEEE Trans Power Syst 17(1):108–112

Lin WM, Gow HJ, Tsai MT (2011) Combining of direct search and signal-to-noise ratio for economic dispatch optimization. Energy Convers Manag 52(1):487–493

Liu S, Feng ZK, Niu WJ, Zhang HR, Song ZG (2019) Peak operation problem solving for hydropower reservoirs by elite-guide sine cosine algorithm with gaussian local search and random mutation. Energies 12(11):2189

Lohokare M, Panigrahi K, Pattnaik SS, Devi S, Mohapatra A (2012) Neighborhood search-driven accelerated biogeography-based optimization for optimal load dispatch. Syst Man Cybern Part C Appl Rev IEEE Trans 42(5):641–652

Long W, Wu T, Liang X, Xu S (2019) Solving high-dimensional global optimization problems using an improved sine cosine algorithm. Expert Syst Appl 123:108–126

Lu H, Sriyanyong P, Song YH, Dillon T (2010) Experimental study of a new hybrid pso with mutation for economic dispatch with non-smooth cost function. Int J Electr Power Energy Syst 32(9):921–935

Mahdad B, Srairi K (2018) A new interactive sine cosine algorithm for loading margin stability improvement under contingency. Electr Eng 100(2):913–933

Mandal B, Roy PK, Mandal S (2014) Economic load dispatch using krill herd algorithm. Int J Electr Power Energy Syst 57:1–10

Mehmood K, Ahmad A (2018) Improved grey wolf optimization for economic load dispatch problem considering valve point loading effect and prohibited operating zones. Nucleus 54(4):250–257

Mirjalili S (2015a) The ant lion optimizer. Adv Eng Softw 83:80–98

Mirjalili S (2015b) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl Based Syst 89:228–249

Mirjalili S (2016a) Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural Comput Appl 27(4):1053–1073

Mirjalili S (2016b) Sca: a sine cosine algorithm for solving optimization problems. Knowl Based Syst 96:120–133

Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67

Mirjalili S, Mirjalili SM, Hatamlou A (2016) Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Comput Appl 27(2):495–513

Mirjalili SM, Mirjalili SZ, Saremi S, Mirjalili S (2020) Sine cosine algorithm: theory, literature review, and application in designing bend photonic crystal waveguides. Nature-inspired optimizers. Springer, Berlin, pp 201–217

Mohammadi F, Abdi H (2018) A modified crow search algorithm (MCSA) for solving economic load dispatch problem. Appl Soft Comput 71:51–65

Mohammadi-Ivatloo B, Rabiee A, Soroudi A, Ehsan M (2012) Iteration pso with time varying acceleration coefficients for solving non-convex economic dispatch problems. Int J Electr Power Energy Syst 42(1):508–516

Moradi-Dalvand M, Mohammadi-Ivatloo B, Najafi A, Rabiee A (2012) Continuous quick group search optimizer for solving non-convex economic dispatch problems. Electr Power Syst Res 93:93–105

Nayak DR, Dash R, Majhi B, Wang S (2018) Combining extreme learning machine with modified sine cosine algorithm for detection of pathological brain. Comput Electr Eng 68:366–380

Nenavath H, Jatoth RK (2018) Hybridizing sine cosine algorithm with differential evolution for global optimization and object tracking. Appl Soft Comput 62:1019–1043

Nenavath H, Jatoth RK, Das S (2018) A synergy of the sine-cosine algorithm and particle swarm optimizer for improved global optimization and object tracking. Swarm Evol Comput 43:1–30

Neto JXV, Reynoso-Meza G, Ruppel TH, Mariani VC, dos Santos CL (2017) Solving non-smooth economic dispatch by a new combination of continuous grasp algorithm and differential evolution. Int J Electr Power Energy Syst 84:13–24

Niknam T, Mojarrad HD, Meymand HZ, Firouzi BB (2011) A new honey bee mating optimization algorithm for non-smooth economic dispatch. Energy 36(2):896–908

Oliva D, Hinojosa S, Abd Elaziz M, Ortega-Sánchez N (2018) Context based image segmentation using antlion optimization and sine cosine algorithm. Multimed Tools Appl 77(19):25761–25797

Ong YS, Keane AJ (2004) Meta-lamarckian learning in memetic algorithms. IEEE Trans Evol Comput 8(2):99–110

Ong YS, Lim MH, Zhu N, Wong KW (2006) Classification of adaptive memetic algorithms: a comparative study. IEEE Trans Syst Man Cybern Part B (Cybern) 36(1):141–152

Özyön S, Aydin D (2013) Incremental artificial bee colony with local search to economic dispatch problem with ramp rate limits and prohibited operating zones. Energy Convers Manag 65:397–407

Pandey VC, Jadoun VK, Gupta N, Niazi K, Swarnkar A (2018) Improved fireworks algorithm with chaotic sequence operator for large-scale non-convex economic load dispatch problem. Arab J Sci Eng 43(6):2919–2929

Pandi VR, Panigrahi BK, Mohapatra A, Mallick MK (2011) Economic load dispatch solution by improved harmony search with wavelet mutation. Int J Comput Sci Eng 6(1):122–131

Panigrahi B, Pandi VR (2008) Bacterial foraging optimisation: Nelder-mead hybrid algorithm for economic load dispatch. IET Gener Transm Distrib 2(4):556–565

Panigrahi B, Pandi VR, Das S, Cui Z, Sharma R (2012) Economic load dispatch using population-variance harmony search algorithm. Trans Inst Meas Control 34(6):746–754

Parouha RP, Das KN (2016) A novel hybrid optimizer for solving economic load dispatch problem. Int J Electr Power Energy Syst 78:108–126

Pawan Preet Singh GSNKVKK Rohit Bains (2017) Comparative analysis on economic load dispatch problem optimization using moth flame optimization and sine cosine algorithms. JARIIE 3:65–75

Pothiya S, Ngamroo I, Kongprawechnon W (2010) Ant colony optimisation for economic dispatch problem with non-smooth cost functions. Int J Electr Power Energy Syst 32(5):478–487

Qin Q, Cheng S, Chu X, Lei X, Shi Y (2017) Solving non-convex/non-smooth economic load dispatch problems via an enhanced particle swarm optimization. Appl Soft Comput 59:229–242

Raut U, Mishra S (2020) A new pareto multi-objective sine cosine algorithm for performance enhancement of radial distribution network by optimal allocation of distributed generators. Evol Intell. https://doi.org/10.1007/s12065-020-00428-2

Reddy KS, Panwar LK, Panigrahi B, Kumar R (2018) A new binary variant of sine-cosine algorithm: development and application to solve profit-based unit commitment problem. Arab J Sci Eng 43(8):4041–4056

Rizk-Allah RM, Hala RAESSHEAAAES, Abdel Mageed M (2020) A new sine cosine optimization algorithm for solving combined non-convex economic and emission power dispatch problems. Int J Energy Convers (IRECON) 5:180–192

Roy P, Roy P, Chakrabarti A (2013) Modified shuffled frog leaping algorithm with genetic algorithm crossover for solving economic load dispatch problem with valve-point effect. Appl Soft Comput 13(11):4244–4252

Roy PK, Bhui S, Paul C (2014) Solution of economic load dispatch using hybrid chemical reaction optimization approach. Appl Soft Comput 24:109–125

Safari A, Shayeghi H (2011) Iteration particle swarm optimization procedure for economic load dispatch with generator constraints. Expert Syst Appl 38(5):6043–6048

Sakthivel V, Suman M, Sathya P (2020) Large-scale economic load dispatch using squirrel search algorithm. Int J Energy Sect Manag 14(6):1351–1380

Saremi S, Mirjalili S, Lewis A (2017) Grasshopper optimisation algorithm: theory and application. Adv Eng Softw 105:30–47

Sayah S, Hamouda A (2013) A hybrid differential evolution algorithm based on particle swarm optimization for nonconvex economic dispatch problems. Appl Soft Comput 13(4):1608–1619

Selvakumar AI, Thanushkodi K (2007) A new particle swarm optimization solution to nonconvex economic dispatch problems. Power Syst IEEE Trans 22(1):42–51

Sharma B, Prakash R, Tiwari S, Mishra K (2017) A variant of environmental adaptation method with real parameter encoding and its application in economic load dispatch problem. Appl Intell 47(2):409–429

Shilaja C, Ravi K (2017) Optimization of emission/economic dispatch using euclidean affine flower pollination algorithm (efpa) and binary fpa (bfpa) in solar photo voltaic generation. Renew Energy 107:550–566

Sindhu R, Ngadiran R, Yacob YM, Zahri NAH, Hariharan M (2017) Sine-cosine algorithm for feature selection with elitism strategy and new updating mechanism. Neural Comput Appl 28(10):2947–2958

Sinha N, Chakrabarti R, Chattopadhyay P (2003) Evolutionary programming techniques for economic load dispatch. IEEE Trans Evol Comput 7(1):83–94

Srinivasa Reddy A, Vaisakh K (2013) Shuffled differential evolution for economic dispatch with valve point loading effects. Int J Electr Power Energy Syst 46:342–352

Subbaraj P, Rengaraj R, Salivahanan S, Senthilkumar T (2010) Parallel particle swarm optimization with modified stochastic acceleration factors for solving large scale economic dispatch problem. Int J Electr Power Energy Syst 32(9):1014–1023

Subbaraj P, Rengaraj R, Salivahanan S (2011) Enhancement of self-adaptive real-coded genetic algorithm using taguchi method for economic dispatch problem. Appl Soft Comput 11(1):83–92

Sudhakaran M, Ajay-D-Vimalraj P, Palanivelu T (2007) Ga and pso culled hybrid technique for economic dispatch problem with prohibited operating zones. J Zhejiang Univ Sci A 8(6):896–903

Suleiman M, Mustafa Z, Mohmed M (2015) Grey wolf optimizer for solving economic dispatch problem with valve-loading effects. APRN J Eng Appl Sci 10:1619–1628

Suresh V, Sreejith S (2017) Generation dispatch of combined solar thermal systems using dragonfly algorithm. Computing 99(1):59–80. https://doi.org/10.1007/s00607-016-0514-9

Tawhid MA, Savsani P (2019) Discrete sine-cosine algorithm (dsca) with local search for solving traveling salesman problem. Arab J Sci Eng 44(4):3669–3679

Tripati P, Tomar U, Singh VK, Bhoi AK (2018) Solution of economic load dispatch problems through moth flame optimization algorithm, devices and networking. Adv Commun. Springer, Berlin, pp 287–294

Tsai MT, Gow HJ, Lin WM (2011) A novel stochastic search method for the solution of economic dispatch problems with non-convex fuel cost functions. Int J Electr Power Energy Syst 33(4):1070–1076

Turgut OE (2017) Thermal and economical optimization of a shell and tube evaporator using hybrid backtracking search-sine-cosine algorithm. Arab J Sci Eng 42(5):2105–2123

Victoire T, Jeyakumar AE (2004) Hybrid pso-sqp for economic dispatch with valve-point effect. Electr Power Syst Res 71(1):51–59

Walters DC, Sheble GB (1993) Genetic algorithm solution of economic dispatch with valve point loading. IEEE Trans Power Syst 8(3):1325–1332

Wang L, Lp L (2013) An effective differential harmony search algorithm for the solving non-convex economic load dispatch problems. Int J Electr Power Energy Syst 44(1):832–843

Wong K, Fung C (1993) Simulated annealing based economic dispatch algorithm. In: IEE proceedings C (generation, transmission and distribution), IET, vol 140, pp. 509–515

Yu JT, Kim CH, Wadood A, Khurshaid T, Rhee SB (2019) Jaya algorithm with self-adaptive multi-population and lévy flights for solving economic load dispatch problems. IEEE Access 7:21372–21384

Zhang H, Gao Z, Zhang J, Liu J, Nie Z (2020) Zhang J (2020) Hybridizing extended ant lion optimizer with sine cosine algorithm approach for abrupt motion tracking. EURASIP J Image Video Process 1:4