Neural Computing and Applications
https://doi.org/10.1007/s00521-018-3796-3

ORIGINAL ARTICLE

Binary whale optimization algorithm and its application to unit commitment problem

Vijay Kumar[1] · Dinesh Kumar[2]

Received: 7 February 2017 / Accepted: 3 October 2018

Abstract

Whale optimization algorithm is a novel metaheuristic algorithm that imitates the social behavior of humpback whales. In this algorithm, the bubble-net hunting strategy of humpback whales is exploited. However, this algorithm, in its present form, is appropriate for continuous problems. To make it applicable to discrete problems, a binary version of this algorithm is being proposed in this paper. In the proposed approach, the solutions are binarized and sigmoidal transfer function is utilized to update the position of whales. The performance of the proposed algorithm is evaluated on 29 benchmark functions. Furthermore, unpaired *t* test is carried out to illustrate its statistical significance. The experimental results depict that the proposed algorithm outperforms others in respect of benchmark test functions. The proposed approach is applied on electrical engineering problem, a real-life application, named as "unit commitment." The proposed approach uses the priority list to handle spinning reserve constraints and search mechanism to handle minimum up/down time constraints. It is tested on standard IEEE systems consisting of 4, 10, 20, 40, 80, and 100 units and on IEEE 118-bus system and Taiwan 38-bus system as well. Experimental results reveal that the proposed approach is superior to other algorithms in terms of lower production cost.

Keywords Whale optimization · Binary whale optimization · Metaheuristics · Benchmark functions · Economic load dispatch · Unit commitment

1 Introduction

In the recent years, it has been observed that more attention is being paid to metaheuristic algorithms which are inspired from the natural phenomena. The well-known nature-inspired metaheuristic algorithms are genetic algorithm (GA) [1], particle swarm optimization (PSO) [2], ant colony optimization [3], bat algorithm (BA) [4], gravitational search algorithm (GSA) [5], etc. These approaches are able to solve real-life computational problems. However, these are developed to solve some specific problems not all types of problems [6]. Hence, novel algorithms are developed every year to solve complex real-life computational problems.

Whale optimization algorithm (WOA) [7] is a recently developed metaheuristic algorithm, which mimics the hunting behavior of whales. It simulates bubble-net attacking mechanism of whales. The WOA was applied to solve many engineering design problems successfully [7]. The original version of WOA was developed for continuous search space. However, several real-life optimization problems require binary search spaces such as feature selection [8], data mining [9], cell formation [10], and unit commitment [11]. The binary algorithms are required to solve these problems. This fact motivates us to develop binary version of WOA. In this paper, binary version of WOA (BWOA) is used to solve the unit commitment problem.

The unit commitment problem (UCP) is a well-known optimization problem in electrical engineering domain.

✉ Vijay Kumar
vijaykumarchahar@gmail.com

Dinesh Kumar
dinesh_chutani@yahoo.com

1 Computer Science and Engineering Department, Thapar Institute of Engineering and Technology, Patiala, Punjab, India

2 Electronics and Communication Engineering Department, Delhi Technological University, Delhi, India

Published online: 16 October 2018

UCP deals with determining the on/off states of power generating units to minimize the operating cost for a given time [12]. The main goal of UCP is to find the optimized schedule to on/off the power generating units to meet the electric power demand while satisfying constraints such as spinning reserve, load balance, minimum up/down time, etc. [13]. Over the last few decades, a number of techniques have been developed to solve UCP.

The techniques used for UCP are broadly classified into three categories. These are classical, metaheuristics and hybrid techniques [13]. The first one has classical optimization techniques such as priority list (PL) [14], Lagrangian relaxation (LR) [15], mixed integer linear programming (MILP) [16], branch and bound approaches (BB) [17], dynamic programming (DP) [18]. The main advantages of these techniques are fast convergence and the simplest form of representation. However, these techniques suffer from poor solution quality (in case of PL), computational time increasing exponentially with dimensionality of system (in case of BB), and problems with dimensionality of system (in case of DP) [13]. These problems have directed to use metaheuristic techniques.

The metaheuristic techniques are successfully used to solve the UCP. These are genetic algorithm (GA) [19], enhanced simulated annealing (SA) [20], binary particle swarm optimization (BPSO) [21], evolutionary programming (EP) [22], binary grey wolf algorithm (BGWA) [13], imperialistic competition algorithm (ICA) [23], etc. The hybrid techniques are developed by integrating both classical and metaheuristic techniques to solve the UCP. The well-known hybrid techniques are Lagrangian relaxation particle swarm optimization (LRPSO) [24], hybrid genetic-ICA (HGICA) [25], hybrid harmony search random search (HHSRSA) [26], local convergence-averse BPSO (LCA-PSO) [27], and hybrid particle swarm optimization grey wolf (PSO-GWO) [28]. A large number of nature-inspired metaheuristic techniques for solving UCP are reported in the literature [29].

The main contribution of this paper is to develop a binary version of whale optimization algorithm (BWOA) for finding optimal regions of the search space. In the original version of WOA, bubble-net mechanism changes the position of search points in continuous search space. On the contrary, in BWOA, the outcome of bubble-net mechanism is converted into a probability value for each component of binary vector that dictates whether that component will take 0 or 1 value. For solving unit commitment problem, the proposed approach uses priority list to handle spinning reserve constraints and search mechanism to handle minimum up/down time constraints efficiently. The proposed approach is applied on 29 standard benchmark test functions. It is compared with seven recently developed metaheuristic techniques. It is also applied on the unit commitment problem with varying number of units from 10 to 100 and complex power systems.

The rest of this paper is organized as follows. Section 2 presents the basic concepts of whale optimization algorithm. The proposed binary version of whale optimization algorithm is described in Sect. 3. In Sect. 4, the proposed approach is investigated on a set of benchmark test functions and compared with existing metaheuristic techniques. Section 5 describes a binary version of whale optimization algorithm for unit commitment problem. The experimental results are discussed in Sect. 4. Section 6 outlines the conclusions.

2 Whale optimization algorithm

Mirjalili and Lewis proposed a whale optimization algorithm (WOA), which is inspired from bubble-net hunting strategy of humpback whales. The humpback whales have a preference to hunt small fishes near the surface. Therefore, they swim near the prey by generating distinctive bubbles along a circle. WOA mimics two main phases. The first phase is exploitation, i.e., encircling a prey and spiral bubble-net attacking method. The second phase is exploration, i.e., searching the prey. The mathematical modeling of whale optimization algorithm (WOA) is given below [7].

2.1 Encircling prey

Humpback whales encircle the prey (i.e., small fishes). They update their position toward the best search agent during the course of iteration. The following equations are used to model their behavior.

$$\mathrm{Dist} = |C \times Y_B(t_{\mathrm{cur}}) - Y(t_{\mathrm{cur}})| \tag{1}$$

$$Y(t_{\mathrm{cur}} + 1) = Y_B(t_{\mathrm{cur}}) - A \times \mathrm{Dist} \tag{2}$$

where Dist represents the distance between the position of best solution (Y_B) and current solution (Y). t_{cur} is the current iteration. $Y_B(t_{\mathrm{cur}})$ represents the position of best solution. A and C are the coefficient vectors that are computed as follows [7]:

$$A = 2 \times au \times rand_1 - au \tag{3}$$

$$C = 2 \times rand_2 \tag{4}$$

The value of au is linearly decreased from 2 to 0 during the course of iterations. The $rand_1$ and $rand_2$ are random variables. The value of these variables lies in the range [0, 1].

2.2 Spiral bubble-net attacking method

The two approaches are used to model the bubble-net behavior of humpback whales. These are shrinking and encircling, and spiral updating mechanism. The shrinking and encircling mechanism is achieved by decreasing the value of *au* linearly making A to have values in the range $[-1, 1]$.

The spiral equation between the positions of humpback whale and prey to mimic the helix-shaped movement of humpback whales is given below:

$$Y(t_{\mathrm{cur}}+1) = \mathrm{Dist}' \times e^{bl} \times cos(2\pi l) + Y(t_{\mathrm{cur}}) \tag{5}$$

Here,

$$Dist' = Y_B(t_{\mathrm{cur}}) - Y(t_{\mathrm{cur}}) \tag{6}$$

where b is a constant parameter for defining the shape of logarithmic spiral, l is a random number that lies in range of $[0, 1]$.

To simulate these approaches simultaneously, a suitable assumption is made, that is, to update the position of whales, and it is equally likely to choose either the shrinking and encircling or spiral path. The mathematical modeling of these mechanisms is given below:

$$Y(t_{\mathrm{cur}}+1) = \begin{cases} Y(t_{\mathrm{cur}}) - A \times \mathrm{Dist} & if\ Rand < 0.5 \\ \mathrm{Dist}' \times e^{bl} \times \cos(2\pi l) + Y(t_{\mathrm{cur}}) & if\ Rand \geq 0.5 \end{cases} \tag{7}$$

where *Rand* is a random number in the range $[0, 1]$.

2.3 Prey search

The prey search is done through the variation of A. For the exploration process, the value of A should be > 1. The mathematical modeling for prey search is given below:

$$\mathrm{Dist}'' = |C \times Y_{rand} - Y| \tag{8}$$

$$Y(t_{\mathrm{cur}}+1) = Y_{rand} - A \times \mathrm{Dist}'' \tag{9}$$

where Y_{rand} is a random whale chosen from the population. The pseudocode of WOA is depicted in Fig. 1 [7].

3 Binary whale optimization algorithm

3.1 Motivation

There are many real-life engineering problems that require discrete binary search space. The well-known problems are feature selection [8], dimensionality reduction [8], data mining [9], and unit commitment [11]. For these problems, the solutions should be encoded in binary vectors. Besides this, the search space should also be considered as binary space. The binary search space can be considered as a hypercube in which the search agents (whales) may go to nearer and farther corners of the hypercube by flipping different number of bits [30].

Hussien et al. [31] and Eid [32] proposed two binary versions of WOA. They proposed change in only one phase, i.e., prey search thus leaving the other two phases, namely shrinking and encircling phases, untouched. The bubble-net behavior of whales also needed modification. These facts motivated us to develop a novel binary version of WOA that incorporates all the modifications mentioned as above. These phases are to be adapted in the discrete search space thus making us move one step further so far as the position updating mechanism of whales in proposed BWOA is concerned.

3.2 Mathematically foundation

For designing the binary version of WOA, it is an imperative to make changes in some basic concepts of original WOA. Due to the binary search space, each dimension can take either 0 or 1 value. Therefore, the movement of search agents (whales) through this dimension may change from 0 to 1 or vice versa.

The foremost difference between original and binary version of WOA is that of position updating mechanism. In BWOA, the toggling between the values 0 and 1 indicates position updating. Furthermore, the value of current bit is changed with a probability that is computed in accordance with helix-shaped movement of whale.

To achieve this, an appropriate transfer function is required to map the helix-shaped movement values into probability values of position updating. The transfer function forces whales to travel in a binary space. Based on the above-mentioned concept, an appropriate probability function can be formulated as:

$$Cstep = \frac{1}{1 + e^{-10(A \times \mathrm{Dist} - 0.5)}} \tag{10}$$

where *Cstep* is the step size that can be computed using sigmoidal function. Dist is the distance between the position of prey and humpback whale.

3.3 Binary whale optimization algorithm (BWOA)

Based on the concept mentioned above in Sect. 3.2, the three major changes in original whale optimization algorithm are proposed.

First, the shrinking and encircling prey phase is modified. The position of whale is modified according to the equation mentioned below:

Fig. 1 Pseudocode of whale optimization algorithm (WOA) [7]

```
Whale Optimization Algorithm (WOA)
Input:
    Number of whales in the pack, n;
    Control Coefficient, au;
    Maximum number of iterations, MAX_ITER;
Output:
    Global best whale position, Y_best
    Best fitness value, fit(Y_best)
begin
    Generate initial population of n whales Y_i (i = 1,2,....,n)
    Set iteration counter t_cur = 0
    Compute the fitness of each whale
    Identify the best whale based on fitness, i.e., Y_best
    while (t_cur < MAX_ITER)
       for each whale do
          Compute control coefficients A and C by Eqs. 3 and 4
        if (Rand < 0.5)
          if (|A| < 1)
            Update the position of the current whale by Eq.2
          else if (|A| ≥ 1)
            Select a random whale, Y_rand
            Update the position of the current whale by Eq. 9
          end if
        else if (Rand ≥ 0.5)
            Update the position of the current whale by Eq. 5
       end if
     end for
       Compute the fitness of all whales
       Update the value of Y_best based on fitness
       Increment the current iteration t_cur by 1.
    end while
    return the best solution, Y_best
end
```

$$Y(t_{\mathrm{cur}}+1)=\begin{cases}\mathrm{complement}(Y(t_{\mathrm{cur}})), & \textit{if } rand < C_{\mathrm{step}}\\ Y(t_{\mathrm{cur}}), & \text{otherwise}\end{cases} \tag{11}$$

where *Cstep* is computed as is given by Eq. (10). A and Dist are computed using Eqs. (1) and (3), respectively.

The second modification is done in the bubble-net behavior of whales that makes use of $Cstep'$ computed as follows:

$$Cstep' = \frac{1}{1+e^{-10(A\times \mathrm{Dist}'-0.5)}} \tag{12}$$

where, A and $Dist'$ are computed using Eqs. (3) and (6), respectively.

The position of helix-shaped movement of humpback whales is updated according to Eq. (5). Thereafter, the modification in position updating process is done as follows:

$$Y(t_{\mathrm{cur}}+1)=\begin{cases}\mathrm{complement}(Y(t_{\mathrm{cur}})), & \textit{if } rand < Cstep'\\ Y(t_{\mathrm{cur}}), & \text{otherwise}\end{cases} \tag{13}$$

The third modification is done in the searching of prey. The mathematical formulation of $Cstep''$ is given below:

$$Cstep'' = \frac{1}{1 + e^{-10(A \times Dist'' - 0.5)}} \tag{14}$$

where, A and $Dist''$ are computed using Eqs. (3) and (8), respectively. Hence, the position of whale is updated according to Eq. (15).

$$Y(t_{\mathrm{cur}} + 1) = \begin{cases} \mathrm{complement}(Y(t_{\mathrm{cur}})), & if\ rand < Cstep'' \\ Y(t_{\mathrm{cur}}), & \text{otherwise} \end{cases} \tag{15}$$

The general steps of Binary version of WOA are depicted in Fig. 2. The changes are highlighted (in red color).

3.4 Computational analysis

In this section, the complexity analysis of BWOA is presented. The time and space complexities of BWOA are described (Fig. 3).

3.4.1 Time complexity

The time complexity of intermediate states of the proposed BWOA is given below.

1. Initialization of BWGO requires $O(n \times D)$ time where n denotes the number of whales and D is the dimension of the benchmark test function.

Fig. 2 Pseudocode of the binary whale optimization algorithm (BWOA)

```
Binary Whale Optimization Algorithm (BWOA)
Input:
    Number of whales in the pack, n;
    Control Coefficient, au;
    Maximum number of iterations, MAX_ITER;
Output:
    Global best binary whale position, Y_best
    Best fitness value, fit(Y_best)
begin
    Generate initial population of n whales Y_i (i = 1,2,....,n)
    Set iteration counter t_cur = 0
    Compute the fitness of each whale
    Identify the best whale based on fitness, i.e., Y_best
    while (t_cur < MAX_ITER)
      for each whale do
        Compute control coefficients A, and C by Eqs. 3 and 4
       if (Rand < 0.5)
         if (|A| < 1)
           Update the position of the current whale by Eq.11
         else if (|A| ≥ 1)
           Select a random whale, Y_rand
           Update the position of the current whale by Eq. 15
         end if
      else if (Rand ≥ 0.5)
           Update the position of the current whale by Eq. 13
      end if
     end for
      Compute the fitness of all whales
      Update the value of Y_best based on fitness
      Increment the current iteration t_cur by 1.
    end while
    return the best solution, Y_best
end
```

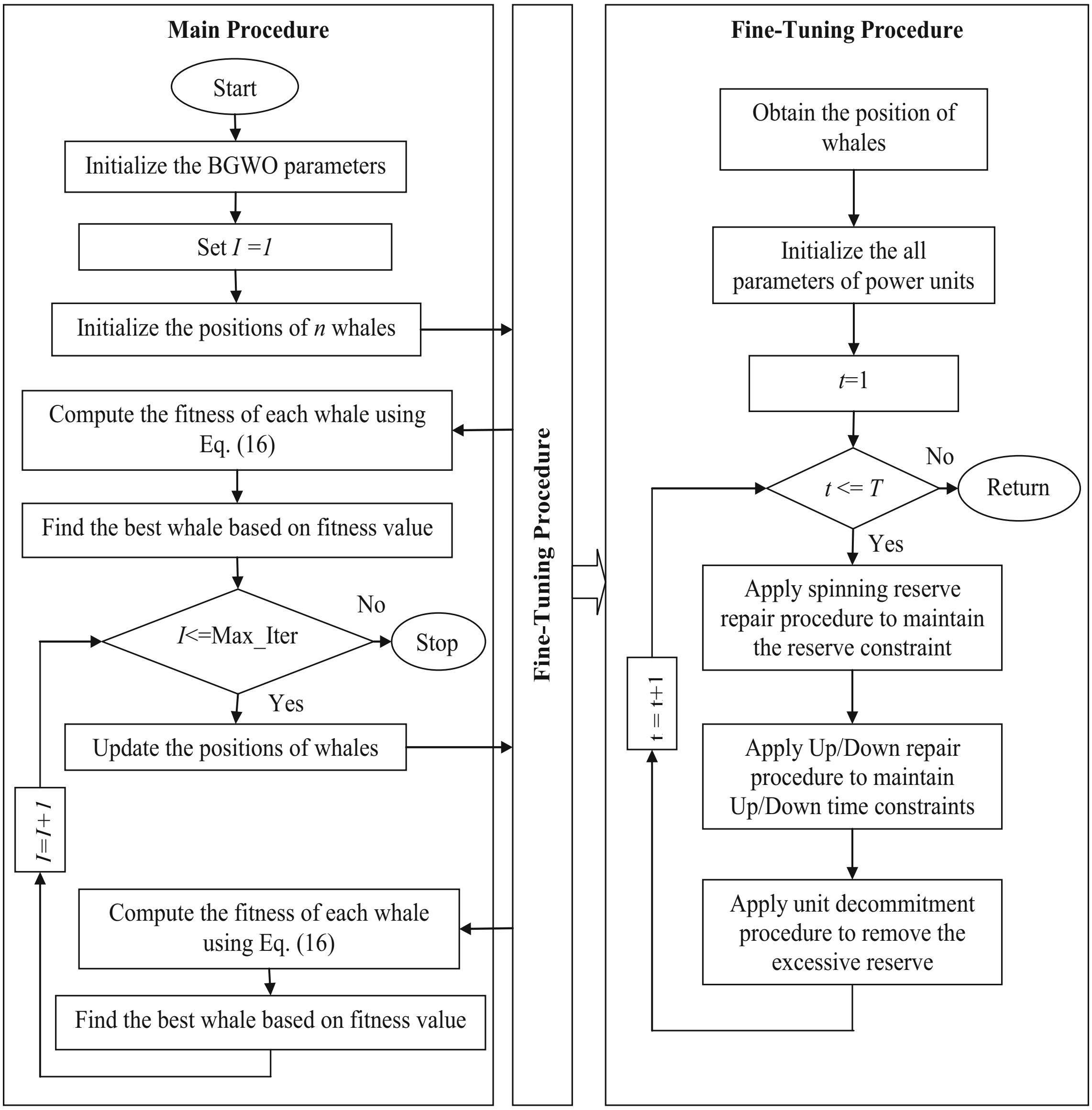

Fig. 3 Flowchart for unit commitment problem using binary whale optimization algorithm

2. Computation of fitness function requires $O(n \times D)$ time.
3. Computation of control parameters and position update steps of BWOA requires $O(n \times D)$ each.

Therefore, summing up the complexities of the above steps and the total complexity of BWOA becomes $O(n \times D)$ per iteration/generation. The total time complexity of BWOA for maximum number of iterations is $O(n \times D \times \text{MaxIteration})$. Here, MaxIteration denotes the maximum number of iterations/generations.

3.4.2 Space complexity

The space requirement of BWOA is due to its initialization process. Thus, the total space complexity of BWOA is $O(n \times D)$.

4 Experimental results

To demonstrate the performance of proposed binary whale optimization algorithm, a set of 29 well-known benchmark test functions [33, 34] are utilized, and the results are reported along with seven well-known metaheuristic algorithms. The simulations are carried out in MATLAB

R2016b environment operating on Windows 7 and 2.2 GHz processor with 8 GB RAM.

4.1 Benchmark test functions used

The benchmark test functions are divided into four categories. These are unimodal, multimodal having many local minima, multimodal having no local minima, and composite functions, which are described in Tables 22, 23, 24, and 25 in Appendix, respectively [33, 34]. In these tables, D indicates dimension of the search space. The first category includes seven test functions $(F_1 - F_7)$. The second category comprises six test functions, i.e., $(F_8 - F_{13})$. The third category consists of 10 test functions, i.e., $(F_{14} - F_{23})$. The fourth category comprises six composite test functions, i.e., $(F_{24} - F_{29})$. These functions are minimization problems.

4.2 Algorithms for comparisons and parameter setting

To assess the performance of proposed algorithm, it is compared with seven recently developed metaheuristic techniques such as genetic algorithm (GA) [1], Grey wolf algorithm (GWA) [35], binary particle swarm optimization (BPSO) [36], binary gravitational search algorithm (BGSA) [30], binary bat algorithm (BBA) [37], binary dragonfly algorithm (BDA) [38], and binary grey wolf algorithm (BGWA) [39]. The parameter settings of above-mentioned algorithms used in this paper are mentioned in Table 1. All the parameters are set according to trial and error on small simulations and are commonly reported in literature.

In addition to these parameters, the maximum number of iterations and population size for the above-mentioned algorithms are set to 500 and 30, respectively. Due to stochastic nature of these algorithms, the results are averaged over 30 independent runs under 30 different random seeds. The mean best-of-run solution and standard deviation of best solution in the last iteration are reported in tables.

Table 1 Parameter setting for algorithms used

Algorithm	Parameters	Values
GA	Selection procedure	Roulette wheel
	Crossover (probability)	Single-point (0.90)
	Mutation (probability)	Uniform (0.005)
GWA	Control parameter (au)	Decrease from 2 to 0
BPSO	Constant parameters (c_1, c_2)	2, 2
	Inertia weight	Decrease from 0.9 to 0.4
	Maximum velocity	6
BGSA	Gravitational constant (G_0)	100
BBA	Frequency range	[0, 2]
	Loudness	0.25
	Pulse emission rate	0.5
	Constant parameter (γ)	0.9
BDA	Cohesion weight	0.7
	Separation weight	0.1
	Alignment weight	0.1
	Inertia weight	Decrease from 0.9 to 0.2
	Enemy factor	1
BGWA	Control parameter (au)	Decrease from 2 to 0
BWOA	au	Decrease from 2 to 0
	b	1

4.3 Results and discussion

The performance of proposed approach is compared with above-mentioned algorithms on benchmark test functions mentioned in Sect. 4.1.

4.3.1 Estimation of exploitation ability

The unimodal benchmark test functions $(F_1 - F_7)$ have only one global optimum. These functions are used to test the exploitation capability of examined metaheuristic algorithms. Table 2 shows the results for unimodal test functions. The table demonstrates that BWOA offers better results than the other metaheuristic algorithms on all unimodal test functions in terms of mean and standard deviation. The results also reveal that the BWOA is able to find global optimum in unimodal test functions. Hence, the proposed algorithm can provide good exploitation.

4.3.2 Estimation of exploration ability

The multimodal benchmark test functions $(F_8 - F_{23})$ have many local optima whose number increases exponentially with increase in problem size. The purpose of these functions is to test the exploration capability of examined metaheuristic algorithms. The results for multimodal test functions $(F_8 - F_{13})$ and fixed dimension multimodal test functions $(F_{14} - F_{23})$ are reported in Tables 3 and 4, respectively. These tables reveal that the proposed BWOA gives better results than the other metaheuristic algorithms on most of multimodal test functions in terms of mean and standard deviation. The results reveal that the BWOA is able to locate global optimum in multimodal test functions. Hence, the proposed algorithm can provide very good exploration capability.

Table 2 Mean and standard deviation of the best-of-run solution on unimodal test functions

Fun	Mean best-of-run solution (SD)							
	GA	GWA	BPSO	BGSA	BBA	BDA	BGWA	BWOA
F_1	1.01E+01	5.29E+00	5.29E+00	2.05E+03	1.85E+00	2.82E−01	1.38E−01	**0.00E+00**
	(2.49E+01)	(9.34E−02)	(2.76E+00)	(4.14E+01)	(2.49E+00)	(4.17E−01)	(1.50E−01)	**(0.00E+00)**
F_2	2.69E−01	1.56E−01	2.22E−01	1.32E+00	9.65E−02	5.89E−02	3.42E−02	**0.00E+00**
	(2.37E−01)	(1.94E−02)	(9.38E−02)	(6.73E−01)	(6.46E−02)	(6.93E−02)	(4.69E−02)	**(0.00E+00)**
F_3	5.55E+02	7.01E+01	2.24E+01	5.09E+02	7.81E+00	1.42E+01	5.10E+00	**0.00E+00**
	(2.51E+02)	(2.19E+01)	(1.02E+03)	(2.66E+02)	(9.79E+00)	(2.26E+01)	(1.84E+00)	**(0.00E+00)**
F_4	1.59E+00	1.31E+00	2.60E+00	7.99E+00	1.15E+00	2.48E−01	1.27E−01	**0.00E+00**
	(1.21E+00)	(3.17E−01)	(8.38E−01)	(3.45E+00)	(6.14E−01)	(2.31E−01)	(3.55E−02)	**(0.00E+00)**
F_5	3.69E+02	9.75E+01	1.48E+02	2.62E+03	25.07E+00	23.55E+00	17.68E+00	**0.00E+00**
	(3.42E+02)	(6.01E+01)	(1.37E+02)	(1.73E+03)	(2.84E+01)	(3.46E+01)	(2.19E+01)	**(0.00E+00)**
F_6	6.98E+00	8.16E−01	8.49E+00	1.26E+02	2.69E+00	9.53E−02	4.45E−02	**0.00E+00**
	(7.01E+00)	(1.26E−02)	(6.14E+00)	(8.80E+01)	(2.74E+00)	(1.29E−01)	(1.06E−01)	**(0.00E+00)**
F_7	4.71E−02	2.21E−02	1.55E−02	2.28E−02	6.00E−03	1.22E−02	3.16E−03	**9.32E−04**
	(4.35E−02)	(1.03E−03)	(7.47E−03)	(2.68E−02)	(4.40E−03)	(1.46E−02)	(2.72E−03)	**(8.73E−04)**

Bold values indicate the best results

Table 3 Mean and standard deviation of the best-of-run solution on multimodal test functions

Fun	Mean best-of-run solution (SD)							
	GA	GWA	BPSO	BGSA	BBA	BDA	BGWA	BWOA
F_8	− 9.29E+02	− 6.12E+02	**− 9.88E+02**	− 8.28E+02	− 9.85E+02	− 9.24E+02	− 9.27E+02	− 9.21E+02
	(2.79E+01)	(1.15E+01)	**(1.42E+01)**	(4.26E+01)	(2.76E+01)	(6.57E+01)	(4.15E+01)	(1.00E−31)
F_9	2.18E+00	1.59E+00	4.97E+00	5.99E+00	1.58E+00	1.80E+00	1.40E+00	**0.00E+00**
	(1.83E+00)	(1.33E+00)	(1.59E+00)	(2.96E+00)	(1.33E+00)	(1.05E+00)	(1.21E+00)	**(0.00E+00)**
F_{10}	1.39E+00	1.89E+00	2.72E+00	2.94E+00	1.15E+00	3.88E−01	2.72E−01	**8.88E−16**
	(1.34E+00)	(5.06E−01)	(4.72E−01)	(1.48E+00)	(7.28E−01)	(5.71E−01)	(3.00E−01)	**(0.00E+00)**
F_{11}	7.06E−01	3.21E−01	3.87E−01	6.47E−01	2.46E−01	1.93E−01	6.59E−02	**0.00E+00**
	(3.22E−01)	(1.78E−02)	(1.32E−01)	(2.28E−01)	(8.39E−02)	(1.13E−01)	(1.76E−02)	**(0.00E+00)**
F_{12}	1.91E−01	2.76E−01	6.21E−01	1.26E+01	2.71E−01	1.49E−01	1.82E−01	**1.12E−01**
	(2.44E−01)	(1.14E−01)	(3.88E−01)	(8.77E+00)	(3.28E−01)	(4.51E−01)	(2.97E−01)	**(0.00E+00)**
F_{13}	1.93E−01	1.67E−01	4.44E−01	9.22E−01	1.29E−01	3.52E−02	2.64E−02	**1.35E−32**
	(2.54E−01)	(3.10E−02)	(2.11E−01)	(2.45E−01)	(7.36E−02)	(5.65E−02)	(6.79E−02)	**(5.56E−48)**

Bold values indicate the best results

4.3.3 Ability to escape from local minima

The composite benchmark test functions $(F_{24} - F_{29})$ are very difficult to optimize as proper balance between exploration and exploitation is required. The optimization results of these functions are reported in Table 5. As seen from this table, the proposed BWOA outperforms other competitive algorithms on all composite functions in terms of mean and standard deviation. The results reveal that the proposed approach provides better balance between exploration and exploitation. The proposed approach uses the concept of $|A| > 1$ to explore the search space and $|A| < 1$ is responsible for exploitation.

4.3.4 Test for statistical significance

Besides the basic statistical analysis (i.e., mean and standard deviation), the unpaired t test is used to test whether the proposed approach differs or not from the rest of the competitor algorithms in a statistically significant way. The sample size for this test is taken as 30. Table 6 illustrates the statistical results based on the optimization result

Table 4 Mean and standard deviation of the best-of-run solution on fixed multimodal test functions

Fun	Mean best-of-run solution (SD)							
	GA	GWA	BPSO	BGSA	BBA	BDA	BGWA	BWOA
F_{14}	4.30E+00	4.04E+00	9.98E−01	1.00E+00	1.20E+00	9.80E−01	9.27E−01	**8.70E−01**
	(2.48E+01)	(4.25E+00)	(6.37E−07)	(8.88E−05)	(6.70E−05)	(7.54E−06)	(5.65E−08)	**(2.11E−09)**
F_{15}	1.29E−02	3.37E−03	9.00E−04	2.10E−03	1.48E−03	2.04E−03	1.34E−03	**4.36E−04**
	(8.29E−05)	(6.25E−04)	(2.65E−08)	(2.99E−06)	(2.71E−06)	(1.58E−06)	(1.42E−06)	**(1.92E−08)**
F_{16}	− 8.93E−01	− 1.03E+00	− 1.02E+00	− 1.02E+00	− 7.68E−01	− 9.26E−01	− 1.05E+00	**− 1.03E+00**
	(1.70E−02)	(5.88E−03)	(9.95E−05)	(4.90E−05)	(2.04E−02)	(1.94E−02)	(3.94E−05)	**(5.01E−05)**
F_{17}	4.39E−01	3.97E−01	3.99E−01	3.99E−01	2.80E−01	2.36E−01	2.39E−01	**2.28E−01**
	(1.12E−02)	(1.51E−04)	(5.23E−07)	(2.99E−10)	(1.47E−02)	(1.27E−03)	(1.62E−05)	**(1.11E−09)**
F_{18}	8.30E+00	3.00E+00	3.00E+00	3.03E+00	2.95E+00	2.91E+00	2.92E+00	**2.90E+00**
	(1.25E+02)	(2.29E−03)	(2.95E−05)	(2.00E−03)	(1.20E−02)	(1.00E−03)	(1.21E−04)	**(1.00E−06)**
F_{19}	− 3.41E+00	− 3.32E+00	− 3.85E+00	− 3.85E+00	− 3.56E+00	− 3.30E+00	− 3.85E+00	**− 3.86E+00**
	(5.76E−02)	(2.30E−02)	(2.98E−06)	(2.94E−07)	(1.19E−02)	(2.24E−03)	(2.87E−04)	**(8.89E−09)**
F_{20}	− 1.75E+00	− 3.28E+00	− 3.11E+00	− 3.30E+00	− 1.35E+00	− 3.05E+00	− 3.16E+00	**− 3.31E+00**
	(2.68E−01)	(6.05E−03)	(2.30E−03)	(1.40E−03)	(3.70E−02)	(2.69E−02)	(1.95E−02)	**(1.06E−05)**
F_{21}	− 1.08E+00	− 6.86E+00	− 6.35E+00	− 3.56E+00	− 3.23E+00	− 3.99E+00	− 4.55E+00	**− 7.89E+00**
	(8.14E−01)	(3.01E+00)	(7.70E+00)	(1.06E+00)	(2.11E+00)	(1.10E+00)	(2.09E+00)	**(2.28E−01)**
F_{22}	− 9.69E−01	− 8.45E+00	− 7.50E+00	− 5.16E+00	− 4.11E+00	− 5.65E+00	− 6.82E+00	**− 9.89E+00**
	(1.20E−01)	(3.08E+00)	(5.63E+00)	(8.81E+00)	(1.80E+00)	(2.75E+00)	(2.31E+00)	**(6.35E−01)**
F_{23}	− 1.44E+00	− 6.57E+00	− 5.75E+00	− 3.57E+00	− 3.18E+00	− 4.02E+00	− 4.73E+00	**− 8.28E+00**
	(7.90E−01)	(2.14E+00)	(2.60E+00)	(1.29E+00)	(2.12E+00)	(1.90E+00)	(2.43E+00)	**(3.60E+00)**

Bold values indicate the best results

Table 5 Mean and standard deviation of the best-of-run solution on composite test functions

Fun	Mean best-of-run solution (SD)							
	GA	GWA	BPSO	BGSA	BBA	BDA	BGWA	BWOA
F_{24}	1.93E+02	1.00E+02	1.94E+02	2.52E+02	9.32E+01	9.27E+01	9.10E+01	**7.09E+01**
	(1.21E+02)	(8.16E+01)	(6.00E+01)	(4.36E+01)	(6.43E+01)	(4.20E+01)	(4.09E+01)	**(3.98E+01)**
F_{25}	2.05E+02	1.55E+02	1.46E+02	2.54E+02	1.56E+02	1.28E+02	1.31E+02	**1.12E+02**
	(1.60E+02)	(2.31E+01)	(2.90E+01)	(3.62E+01)	(3.18E+01)	(2.05E+01)	(2.79E+01)	**(1.58E+01)**
F_{26}	3.84E+02	1.72E+02	4.45E+02	2.13E+02	1.49E+02	1.03E+02	1.19E+02	**9.62E+01**
	(1.18E+02)	(3.27E+01)	(4.93E+01)	(6.36E+01)	(3.87E+01)	(2.76E+01)	(1.02E+01)	**(9.03E+00)**
F_{27}	5.88E+02	3.14E+02	4.79E+02	2.55E+02	1.46E+02	1.16E+02	1.22E+02	**9.98E+01**
	(1.02E+02)	(2.06E+01)	(3.01E+01)	(5.01E+01)	(2.29E+01)	(1.98E+01)	(1.56E+01)	**(1.03E+01)**
F_{28}	2.46E+02	1.83E+02	1.72E+02	2.45E+02	1.66E+02	1.46E+02	1.37E+02	**1.12E+02**
	(1.83E+02)	(1.01E+01)	(6.42E+01)	(5.28E+01)	(4.98E+01)	(3.55E+01)	(2.93E+01)	**(2.44E+01)**
F_{29}	9.14E+02	8.61E+02	6.91E+02	2.32E+02	1.52E+02	1.34E+02	1.28E+02	**9.94E+01**
	(1.23E+01)	(1.25E+02)	(1.49E+02)	(2.76E+01)	(3.36E+01)	(2.77E+01)	(2.09E+01)	**(1.88E+01)**

Bold values indicate the best results

obtained from benchmark test functions. According to the p values in Table 6, BWOA is significantly different from other competitive algorithms for all unimodal benchmark functions ($F_1 - F_7$). The p values reported in Table 6 show that BWOA do not only have a significant improvement in F_{12} and F_{21}, but this attains significant improvement in all the remaining multimodal benchmark test functions ($F_8 - F_{11},\ F_{13} - F_{20},\ and\ F_{22} - F_{23}$) also. BWOA is significantly better in five out of six composite functions based on the p values reported in Table 6. Therefore,

Table 6 Unpaired t test among BWOA and second-best algorithm

	SE	t	95% CI	Two-tailed p	Significance
F_1	0.076	3.7040	− 0.43771042 to − 0.12628958	0.0009	Extremely significant
F_2	0.013	4.6552	− 0.08477706 to − 0.03302294	< 0.0001	Extremely significant
F_3	1.787	4.3695	− 11.38787195 to − 4.23212805	< 0.0001	Extremely Significant
F_4	0.042	5.8803	− 0.33242170 to − 0.16357830	< 0.0001	Extremely significant
F_5	6.317	3.7280	− 36.19498157 to − 10.90501843	0.0004	Extremely significant
F_6	0.024	4.0464	− 0.14244458 to − 0.04815542	0.0002	Extremely significant
F_7	0.001	6.1881	0.00342862 to 0.00670738	< 0.0001	Extremely significant
F_8	3.042	22.2100	− 73.65441381 to − 61.47558619	< 0.0001	Extremely significant
F_9	0.243	6.5068	− 2.06606432 to − 1.09393568	< 0.0001	Extremely significant
F_{10}	0.104	3.7218	− 0.59667874 to − 0.17932126	0.0004	Extremely significant
F_{11}	0.021	9.3549	− 0.23429719 to − 0.15170281	< 0.0001	Extremely significant
F_{12}	0.082	0.4494	− 0.12782331 to 0.20182331	0.0004	Not significant
F_{13}	0.010	3.4124	0.01455140 to 0.05584860	0.0012	Statistically significant
F_{14}	0.000	7906.47	− 0.11000276 to − 0.10999724	< 0.0001	Extremely significant
F_{15}	0.000	2109.99	− 0.00104499 to − 0.00104301	< 0.0001	Extremely significant
F_{16}	0.000	7253.21	− 9.28002561 to − 9.27997439	< 0.0001	Extremely significant
F_{17}	0.000	34.5022	0.00753586 to 0.00846414	< 0.0001	Extremely significant
F_{18}	0.000	54.7722	0.00963454 to 0.01036546	< 0.0001	Extremely significant
F_{19}	0.000	1862.07	0.00999989 to 0.01000011	< 0.0001	Extremely significant
F_{20}	0.000	39.1219	− 0.01051166 to − 0.00948834	< 0.0001	Extremely significant
F_{21}	1.406	1.0950	− 1.27528997 to 4.35528997	0.2781	Not significant
F_{22}	1.034	2.3105	0.31940356 to 4.46059644	0.0244	Statistically significant
F_{23}	0.811	3.1205	− 4.15291419 to − 0.90708581	0.0028	Statistically significant
F_{24}	10.564	2.0636	0.65355238 to 42.94644762	0.0435	Statistically significant
F_{25}	4.725	3.3859	6.54102997 to 25.45897003	0.0013	Statistically significant
F_{26}	5.302	1.2826	− 3.81288318 to 17.41288318	0.2047	Not significant
F_{27}	4.075	3.9756	8.04331950 to 24.35668050	0.0002	Extremely significant
F_{28}	7.865	4.3231	18.25707400 to 49.74292600	0.0001	Extremely significant
F_{29}	6.112	5.6609	22.36532040 to 46.83467960	< 0.0001	Extremely significant

BWOA is statistically significant over other competitive algorithms in most of benchmark functions.

5 Application to unit commitment problem

In this section, the performance of BWOA is evaluated by solving a real-life problem in electrical engineering. The unit commitment problem (UCP) is one of the most important problems in electrical engineering. Hence, the proposed BWOA is being applied to UCP.

5.1 Mathematical formulation of UCP

5.1.1 Objective function

The foremost objective of UCP is to find the optimal schedule for operating the available generation units such that the total generation cost should be minimized. The total power generation cost consists of fuel cost, start-up and shutdown costs. The fuel cost comprises fuel price, heat rate of generating units, turn-on, turn-off, and initial status of units. The total fuel cost (T_{fc}) over time t is given below [40, 41]:

$$T_{\mathrm{fc}} = \sum_{t=1}^{T} \sum_{i=1}^{N} \left[f_{\mathrm{cost}}\left(P_i^t\right) + SUC_i^t\left(1 - u_i^{t-1}\right)\right] u_i^t \tag{16}$$

where N is the number of generating units; T is total scheduling time; $f_{\mathrm{cost}}\left(P_i^t\right)$ is the fuel cost of i^{th} unit at time t; u_i^t is the on/off status of ith unit at time t; P_i^t is power generation of ith unit at time t; SUC_i^t is the start-up cost of ith unit at time t.

The mathematical representation of fuel cost function is given below [42, 43]:

$$f_{\text{cost}}(P_i^t) = \alpha_i + \beta_i P_i^t + \gamma_i (P_i^t)^2 \tag{17}$$

where α_i, β_i and γ_i are the fuel cost coefficients of ith unit. The start-up cost can be defined as [28]:

$$\begin{aligned}&\text{SUC}_i^t\\&= \begin{cases} \text{HSC}_i^t, & \forall\,\text{MDT}_i^t \le \text{MDT}_{i,on}^t \le (\text{MDT}_i^t + \text{CSH}_i^t) \\ \text{CSC}_i^t, & \forall\,\text{MDT}_{i,on}^t > (\text{MDT}_i^t + \text{CSH}_i^t) \end{cases} (i \in N;\;\; t \in T)\end{aligned} \tag{18}$$

where CSC_i^t and HSC_i^t are cold and hot start-up cost of ith unit at time t, respectively. CSH_i^t and MDT_i^t are cold start hour and minimum down time of ith unit at time t, respectively. $\text{MDT}_{i,\text{on}}^t$ is the continuously on time of ith unit up to time t. The constraints are associated with UCP that are stated below.

5.1.2 Power balance constraint

The total generated power must be equal to power demand in each hour. The constraint is stated below [43]:

$$\sum_{i=1}^{N} P_i^t u_i^t - P_D^t = 0, \quad t = 1, 2, \ldots, T \tag{19}$$

where P_D^t is the power demand at time t.

5.1.3 Spinning reserve constraint

The important feature of reliability is a condition of overload capacity of generation unit. The overload capacity of generation is known as spinning reserve constraint. The spinning reserve constraint is defined as follows:

$$\sum_{i=1}^{N} P_{i(\max)}\, u_i^t \ge P_D^t + P_R^t \tag{20}$$

where P_R^t is the spinning reserve at time t. $P_{i(\max)}$ is the maximum power limit of unit i.

5.1.4 Minimum up and down time constraints

Minimum up/down time constraints means that a generation unit should be on or off for a minimum number of hours before commitment (i.e., off to on state) or decommitment (i.e., on to off state), respectively. These constraints are defined as follow:

$$u_i^t = \begin{cases} \text{off} \rightarrow \text{on}, & \textit{if } T_{i,\text{off}}^t \ge \text{MDT}_i \\ \text{on} \rightarrow \text{off}, & \textit{if } T_{i,\text{on}}^t \ge \text{MUT}_i \end{cases} \tag{21}$$

where $T_{i,\text{on}}^t$ represents continuously on time of unit i up to time t, MUT_i is minimum up time of unit i, $T_{i,off}^t$ represents continuously off time of unit i up to time t and MDT_i is unit i minimum down time.

5.1.5 Ramping constraints

Due to the unit's mechanical characteristics, the output power of each unit is restricted by their corresponding ramp rate constraints. The ramp-up and ramp-down constraints are given below [43]:

$$P_i^t - P_i^{t-1} \le \text{RU}_i \quad \text{as generation increases} \tag{22}$$

$$P_i^{t-1} - P_i^t \le \text{RD}_i \quad \text{as generation decreases} \tag{23}$$

where RD_i and RU_i are the ramp-down and ramp-up limits of unit i, respectively.

5.1.6 Unit generation limit constraint

The power generation of each unit must be within their specified limits, which are given below:

$$u_i^t \cdot \Gamma_{i(min)}^t \le P_i^t \le u_i^t \cdot \Gamma_{i(max)}^t \tag{24}$$

where $\Gamma_{i(min)}^t$ and $\Gamma_{i(max)}^t$ are minimum and maximum generating power limit of unit i at time t respectively. These are computed as follows:

$$\Gamma_{i(min)}^t = \begin{cases} \max(P_{i(min)}, P_i^{t-1} - RD_i), & \textit{if } u_i^{t-1} = u_i^t = 1 \text{ and } \psi = 1 \\ P_{i(min)}, & \textit{if } u_i^{t-1} = 0 \text{ and } u_i^t = 1 \\ 0, & \textit{if } u_i^{t-1} = 0 \text{ or } 1 \text{ and } u_i^t = 0 \end{cases} \tag{25}$$

$$\Gamma_{i(max)}^t = \begin{cases} \min(P_{i(max)}, P_i^{t-1} + RU_i), & \textit{if } u_i^{t-1} = u_i^t = 1 \text{ and } \psi = 1 \\ P_{i(max)}, & \textit{if } u_i^{t-1} = 0 \text{ and } u_i^t = 1 \\ 0, & \textit{if } u_i^{t-1} = 0 \text{ or } 1 \text{ and } u_i^t = 0 \end{cases} \tag{26}$$

where ψ represents the indicator for ramp rate constraints. $P_{i(\min)}$ and $P_{i(\max)}$ are the minimum and maximum power limit of unit i respectively. The ramp rate constraints are to be considered when the value of ψ is 1; otherwise, these constraints are not to be considered.

5.2 Binary whale optimization algorithm for unit commitment

Figure 3 shows the flowchart of unit commitment problem using binary whale optimization algorithm. The basic steps of BWOA for unit commitment are mentioned below:

Step 1. Initialization of population The status of each unit on/off is distinguished as a chromosome. The status of available unit at each hour constitutes a sub-chromosome. Therefore, an agent consists of T sub-chromosomes. An agent denotes unit commitment schedule over time horizon T. It is represented in terms of integer matrix U having dimensions $N \times T$

$$U = \begin{bmatrix} u_1^1 & u_1^2 & u_1^3 & \cdots & u_1^T \\ u_2^1 & u_2^2 & u_2^3 & \cdots & u_2^T \\ u_3^1 & u_3^2 & u_3^3 & \cdots & u_3^T \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ u_N^1 & u_N^2 & u_N^3 & \cdots & u_N^T \end{bmatrix} \quad (27)$$

where u_i^t is the on/off status of unit i at a time t.

A set of agents are formed randomly in the initialization process. For NP agents, the candidate solution of each agent $U_k (k = 1, 2, \ldots, NP)$ is randomly initialized. The element u_i^t of each agent U_k is generated either 0 or 1 using random function.

Step 2. Evaluation of priority list for unit commitment The priority list is based on fuel cost obtained from the average fuel cost of each unit operating at its maximum output power. The average full load cost (fc) of a unit is defined as the cost per unit of power when power unit is at its full capacity. The fuel cost of unit i is obtained from Eq. 17, fc_i can be stated as:

$$\mathrm{fc}_i = \frac{\alpha_i}{P_{i(\max)}} + \beta_i + \gamma_i \cdot P_{i(\max)} \quad (28)$$

The priority list of power units will be formulated based on the order of fc. The power unit having lowest fc will be considered as highest priority.

Step 3. Modification of agents to assure spinning reserve constraints The primary unit commitment may not satisfy the spinning reserve constraint, which is given by Eq. (20). The spinning reserve violations in primary unit commitment are revamped using following procedure.

Step 3.1 *Set $t = 1$*

Step 3.2 *Compute fc_i for all the uncommitted units (u_i) at hour t. Prepare a sorted list say, $LU(\mathrm{fc}_i)$ based on the value of fuel cost*

Step 3.3 *Compute excessive spinning reserve at each hour (R^t)*

$$R^t = \sum_{i=1}^{N} P_{i(\max)} u_i^t - P_D^t - P_R^t. \quad (29)$$

Step 3.4 *If $R^t \geq 0$ then goto Step 3.6.*

Step 3.5 *Generate a random number $\zeta \in [0, 1]$. If $\zeta < P_R$, then commit an uncommitted unit in $LU(\mathrm{fc}_i)$ that having lowest fc_i and return to Step 3.3; Otherwise, randomly commit an uncommitted unit in $LU(\mathrm{fc}_i)$ and return to Step 3.3*

Step 3.6 *If $t < T$ then $t = t + 1$ and return to Step 3.2. Otherwise, Stop.*

Step 4. Modification of agents to assure minimum up/down time constraints The unit schedule obtained from previous step, may be defy the minimum up and down time constraints. The on and off times of units should be determined for violation of up/down time constraints. The continuously on/off times of unit i up to hour t is computed as follows:

$$T_{i,\mathrm{on}}^{t} = \begin{cases} T_{i,\mathrm{on}}^{t-1} + 1 & \text{if } u_i^t = 1 \\ 0 & \text{if } u_i^t = 0 \end{cases}, \quad T_{i,\mathrm{off}}^{t} = \begin{cases} T_{i,\mathrm{off}}^{t-1} + 1 & \text{if } u_i^t = 0 \\ 0 & \text{if } u_i^t = 1 \end{cases} \tag{30}$$

The procedure for repair violations of minimum up and down times constraints is mentioned below.

Step 4.1 *Compute on and off times of all units for whole schedule time horizon using Eq.* (30)

Step 4.2 *Set* $t = 1$.

Step 4.3 *Set* $i = 1$.

Step 4.4 *If* $u_i^t = 0$ and $u_i^{t-1} = 1$ and $T_{i,\mathrm{on}}^{t-1} < \mathrm{MUT}_i$, *then set* $u_i^t = 1$.

Step 4.5 *If* $u_i^t = 0$ and $u_i^{t-1} = 1$ and $t + \mathrm{MDT}_i - 1 \le T$ and $T_{i,\mathrm{off}}^{t+\mathrm{MDT}_i-1} < \mathrm{MDT}_i$, *then set* $u_i^t = 1$.

Step 4.6 *If* $u_i^t = 0$ and $u_i^{t-1} = 1$ and $t + \mathrm{MDT}_i - 1 > T$ and $\sum_{l=t}^{T} u_i^l > 0$, *then set* $u_i^t = 1$.

Step 4.7 *Update the on and off times for unit i using Eq.* (30).

Step 4.8 *If* $i < N$ *then* $i = i + 1$ *and goto Step 4.4.*

Step 4.9 *If* $t < T$ *then* $t = t + 1$ *and goto Step 4.3. Otherwise, Stop.*

*Step 5. Decommitment of excess units*The previous procedure for repairing minimum up and down time constraints may lead to excessive spinning reserves. The procedure for decommitment of excessive units is described as below

Step 5.1 *Set* $t = 1$.

Step 5.2 *Compute* fc_i *for each committed unit at hour t. Prepare a sorted list* $LU(\mathrm{fc}_i)$ *based on* fc_i. *Suppose the first unit in* $LU(\mathrm{fc}_i)$ *be* CU^t

Step 5.3 *Compute excessive spinning reserve at each hour*

$$R^t = \sum_{i=1}^{N} P_{i(\max)} u_i^t - P_D^t - P_R^t \tag{31}$$

Step 5.4 *If* $R^t \ge P_{\max}(CU^t)$ *then goto Step 5.6. Here* $P_{\max}$ *denotes maximum generating power.*

Step 5.5 *If decommitting* CU^t *does not violate their minimum up/down time constraint, then decommit* CU^t *and update on/off status of all units.*

Step 5.6 *Delete* CU^t *from* $LU(fc_i)$

Step 5.7 *If* $LU(fc_i)$ *is not empty. Let the first unit in* $LU(fc_i)$ *be* CU^t *and goto Step 5.3.*

Step 5.8 *If* $t < T$ *then* $t = t + 1$ *and goto Step 5.2. Otherwise, Stop.*

*Step 6. Evaluation of fitness function of all agents*The fitness function for each agent is computed using Eq. (16). The agent are sorted according to the fitness values of agents*Step 7. Update the position of agents*The position of each agent is updated using the BWOA which is mentioned in Sect. 2.2*Step 8. Termination Criterion*If maximum number of iteration is reached, then optimal UCP schedule and solutions are obtained. Otherwise, the iteration counter is incremented and go to Step 3

5.3 Performance evaluation

In order to demonstrate the effectiveness of BWOA for solving UCP, it is tested on three test systems such as small-scale, medium-scale and large-scale power systems. For these test systems, the spinning reserve requirement is considered to 10% of the load demand. The load demand and characteristics of generating power systems are taken from [12, 44] and described in Appendix 1. The load demand pattern of generating power systems is shown in Fig. 4. The population size and maximum number of iterations for BWOA are set to 50 and 500, respectively. The other parameters remain same as mentioned in Sect. 4.2. Due to stochastic nature of BWOA, 30 test trails are constructed for each test set. Each trail starts with different initial populations. The best, average, and worst costs are reported in tables. The optimality gap for small (10 units) and medium-scale power system (20 units) is also mentioned as the optimal solution values for these instances are obtained thorough CPLEX software [45].

5.3.1 Test system 1: small-scale power system

For small-scale power system, two different cases are considered. The 4-unit and 10-unit test systems are taken into consideration for this power system.

Case-I: 4-unit test system

The first test system consists of 4-unit test system. The scheduling periods are 8 h for 4-unit test system. The load demand and characteristics of generating unit test are shown in Tables 26 and 27, respectively. Table 7 shows optimal commitment and generation schedule obtained from the proposed approach. The total operational and start-up costs obtained from proposed approach are \$74644.07 and \$0, respectively. The results were reported from literature when same problem was solved using improved Lagrangian relaxation (ILR) [15], A.SMP [44], B.SMP [44], Lagrangian relaxation and PSO (LRPSO) [41], binary differential evolution (BDE) [40], genetic algorithm (GA) [19], binary fireworks algorithm (BFWA) [42], binary grey wolf algorithm (BGWA) [13], and hybrid genetic imperialist competitive algorithm (HGICA) [25]. The performance comparison between BWOA and the above-mentioned metaheuristic techniques is depicted in Table 8. The experimental results reveal that the proposed BWOA outperforms the other techniques in terms of total cost. The superiority of BWOA is evident that it satisfies all the constraints except the ramp rate constraints that are not considered in this case. Among the existing techniques, HGICA is the second-best algorithm.

Case-II: 10-unit test system

The second test system consists of 10-generating unit systems. The scheduling periods are 24 h. The load demand data and characteristics of generating systems are shown in Tables 28 and 29 respectively. The optimal commitment and generation schedules obtained from BWOA are depicted in Tables 9 and 10 respectively. From these tables, it is observed that the total operational and start-up costs, obtained from BWOA, are \$559846.02 and \$4090, respectively. The performance of BWOA is compared with eleven well-known techniques and comparison results are reported in Table 11. These are local convergence-averse binary particle swarm optimization (LCA-PSO) [27], evolutionary programming (EP) [22], Lagrangian relaxation (LR) [15], enhanced simulated annealing (ESA) [20], binary PSO (BPSO) [21], improved binary PSO (IBPSO) [11], BDE [40], GA [19], BFWA [42], BGWA [13], and HGICA [25]. The results obtained from

Table 8 Comparison of results obtained for 4-unit system without ramp rate constraints

Method	Best cost (\$)	Average cost (\$)	Worst cost (\$)
ILR [15]	75,231	NA	NA
A.SMP [44]	74,812	74,877	75,166
B. SMP [44]	74,812	74,877	75,166
LRPSO [24]	74,808	NA	NA
BDE [40]	74,676	NA	NA
GA [19]	74,675	NA	NA
BFWA [42]	74,723	74,723	74,723
BGWA [13]	74,666	74,666	74,666
HGICA [25]	74,653	74,653	74,653
BWOA	**74,644.07**	**74,644.07**	**74,644.07**

Bold values indicate the best results

Table 7 Optimal commitment schedule and unit output power of 4-unit system using BWOA

Commitment schedule					Output power				Run cost (\$/h)	Start-up cost (\$/h)
Hours	PU1	PU2	PU3	PU4	PU1	PU2	PU3	PU4		
1	1	1	0	0	300	150	0	0	9145.36	0
2	1	1	1	0	300	205	25	0	10,892.24	0
3	1	1	1	1	300	250	30	20	12,570.54	0
4	1	1	1	0	300	215	25	0	11,079.38	0
5	1	0	1	1	300	0	80	0	8523.53	0
6	1	0	1	0	255	0	25	0	6103.14	0
7	1	0	1	0	265	0	25	0	6279.82	0
8	1	1	0	0	300	200	0	0	10,050.06	0
	Total operational cost = 74,644.07\$				Total start-up cost = 0\$				Total cost = 74,644.07\$	

Table 9 Optimal commitment schedule of 10-unit power system using BWOA

Hour	PU1	PU2	PU3	PU4	PU5	PU6	PU7	PU8	PU9	PU10
1	1	1	0	0	0	0	0	0	0	0
2	1	1	0	0	0	0	0	0	0	0
3	1	1	0	0	1	0	0	0	0	0
4	1	1	0	0	1	0	0	0	0	0
5	1	1	0	1	1	0	0	0	0	0
6	1	1	1	1	1	0	0	0	0	0
7	1	1	1	1	1	0	0	0	0	0
8	1	1	1	1	1	0	0	0	0	0
9	1	1	1	1	1	1	1	0	0	0
10	1	1	1	1	1	1	1	1	0	0
11	1	1	1	1	1	1	1	1	1	0
12	1	1	1	1	1	1	1	1	1	1
13	1	1	1	1	1	1	1	1	0	0
14	1	1	1	1	1	1	1	0	0	0
15	1	1	1	1	1	0	0	0	0	0
16	1	1	1	1	1	0	0	0	0	0
17	1	1	1	1	1	0	0	0	0	0
18	1	1	1	1	1	1	0	0	0	0
19	1	1	1	1	1	1	0	0	0	0
20	1	1	1	1	1	1	1	1	0	0
21	1	1	1	1	1	1	1	0	0	0
22	1	1	1	0	1	0	1	0	0	0
23	1	1	0	0	0	1	0	0	0	0
24	1	1	0	0	0	0	0	0	0	0

Table 11 reveal that the BWOA gives comparable results with HGICA. The minimum generation cost obtained from BWOA and HGICA are \$563,936.02 and \$563,935.31, respectively. Table 11 also shows the optimality gap for this test instance as the optimal solution value is \$563,938, which is obtained from CPLEX. Table 11 shows that the proposed approach is able to find optimal solution. The proposed approach performs better than other techniques in terms of generation cost.

5.3.2 Test system 2: medium-scale power system

Medium-scale power systems are tested for two different cases. The 20-unit and 40-unit test systems are taken into consideration for this power system. The scheduling periods are 24 h for these power systems. The ramp rate constraints are not considered for this test system. The proposed approach is compared with eleven well-established techniques such as local convergence-averse binary particle swarm optimization (LCA-PSO) [27], evolutionary programming (EP) [22], Lagrangian relaxation (LR) [15], enhanced simulated annealing (ESA) [20], binary differential evolution (BDE) [40], binary PSO (BPSO) [21], improved binary PSO (IBPSO) [11], genetic algorithm (GA) [19], binary fireworks algorithm (BFWA) [42], binary grey wolf algorithm (BGWA) [13], and hybrid genetic imperialist competitive algorithm (HGICA) [25].

Case-I: 20-unit test system

The first system consists of 20-unit test system. For 20-unit test system, the characteristics of 10-unit test system are replicated and load demand is taken as two times of 10-unit test system's load. Tables 12 and 13 show the optimal commitment and generation schedule obtained from BWOA. The operational and start-up costs obtained from BWOA are \$1,115,120 and \$8440, respectively.

The performance of proposed BWOA is compared with other techniques reported in the literature, which are reported in Table 14. In Table 14, the "NA" means the values are not available in the literature. It is observed from this table that the best cost obtained from BWOA is \$1,123,560. The exploration and exploitation capability of BWOA provide quality solution for UCP. After the proposed BWOA, HGICA provides better results than the other techniques. The best cost obtained from HGICA is \$1,124,565, which is \$1005 higher than the proposed approach. According to the results reported in Table 14, HGICA is the second-best algorithm among the above-

Table 10 Optimal unit output power for 10-unit system using BWOA

Hour	PU1	PU2	PU3	PU4	PU5	PU6	PU7	PU8	PU9	PU10	Run cost	Start-up cost
1	455	245	0	0	0	0	0	0	0	0	13,683.13	0
2	455	295	0	0	0	0	0	0	0	0	14,554.50	0
3	455	370	0	0	25	0	0	0	0	0	16,809.45	900
4	455	455	0	0	40	0	0	0	0	0	18,597.67	0
5	455	390	0	130	25	0	0	0	0	0	20,020.02	560
6	455	360	130	130	25	0	0	0	0	0	22,387.04	1100
7	455	410	130	130	25	0	0	0	0	0	23,261.98	0
8	455	455	130	130	30	0	0	0	0	0	24,150.34	0
9	455	455	130	130	85	20	25	0	0	0	27,251.05	860
10	455	455	130	130	162	33	25	10	0	0	30,057.55	60
11	455	455	130	130	162	73	25	10	10	0	31,916.06	60
12	455	455	130	130	162	80	25	43	10	10	33,890.16	60
13	455	455	130	130	162	33	25	10	0	0	30,057.55	0
14	455	455	130	130	85	20	25	0	0	0	27,251.05	0
15	455	455	130	130	30	0	0	0	0	0	24,150.34	0
16	455	310	130	130	25	0	0	0	0	0	21,513.66	0
17	455	260	130	130	25	0	0	0	0	0	20,641.82	0
18	455	340	130	130	25	20	0	0	0	0	23,037.12	0
19	455	440	130	130	25	20	0	0	0	0	23,897.57	0
20	455	455	130	130	162	33	25	10	0	0	30,057.60	490
21	455	455	130	130	85	20	25	0	0	0	27,251.05	0
22	455	455	130	0	35	0	25	0	0	0	22,336.53	0
23	455	425	0	0	0	20	0	0	0	0	17,645.36	0
24	455	345	0	0	0	0	0	0	0	0	15,427.42	0
	Total operational cost = 559,846.02$						Total start-up cost = 4090$				Total cost = 563,936.02 $	

Table 11 Comparison of results obtained for 10-unit system without ramp rate constraints

Method	Best cost ($) (optimality gap)	Average cost ($) (optimality gap)	Worst cost ($) (optimality gap)
LCA-PSO [27]	570,006 (1.0760)	NA	NA
EP [22]	564,551 (0.1087)	565,352 (0.2507)	566,231 (− 0.4066)
LR [15]	566,107 (0.3846)	566,493 (0.4530)	566,817 (− 0.5105)
ESA [20]	565,828 (0.3351)	565,988 (0.3635)	566,260 (− 0.4117)
BDE [40]	563,997 (0.0104)	563,997 (0.0104)	563,997 (− 0.0104)
BPSO [21]	563,977 (0.0069)	563,977 (0.0069)	563,977 (− 0.0069)
IBPSO [11]	563,977 (0.0069)	564,155 (0.0384)	565,312 (− 0.2436)
GA [19]	563,977 (0.0069)	564,275 (0.0597)	566,606 (− 0.4731)
BFWA [42]	563,977 (0.0069)	564,018 (0.0141)	564,855 (− 0.1626)
BGWA [13]	563,976 (0.0067)	564,378 (0.0780)	565,518 (− 0.2802)
HGICA [25]	563,935.31 (− 0.0004)	563,937 (− 0.0002)	563,938 (0.0000)
BWOA	563,936.02 (− 0.0003)	563,938.89 (0.0001)	563,939.01 (− 0.0001)

Table 12 Optimal commitment schedule of 20-unit power system using BWOA

	PU1	PU2	PU3	PU4	PU5	PU6	PU7	PU8	PU9	PU10	PU11	PU12	PU13	PU14	PU15	PU16	PU17	PU18	PU19	PU20
1	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
2	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
3	1	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
4	1	1	0	0	1	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0
5	1	1	1	0	1	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0
6	1	1	1	1	1	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0
7	1	1	1	1	1	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0
8	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
9	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	0	1	0	0	0
10	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	0	0
11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
13	1	1	1	1	1	1	1	1	0	0	1	1	1	1	1	1	1	0	1	0
14	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	1	0	0	0	0
15	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
16	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
17	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
18	1	1	1	1	1	0	1	0	0	0	1	1	1	1	1	1	0	0	0	0
19	1	1	1	1	1	0	1	0	0	0	1	1	1	1	1	1	0	0	0	0
20	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	0	0	0
21	1	1	1	1	1	1	1	0	0	0	1	1	1	1	1	0	1	0	0	0
22	1	1	1	0	1	1	0	0	0	0	1	1	0	0	1	0	1	0	0	0
23	1	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
24	1	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0

Table 13 Unit output power for 20-unit system using BWOA

	PU1	PU2	PU3	PU4	PU5	PU6	PU7	PU8	PU9	PU10	PU11	PU12	PU13	PU14	PU15	PU16	PU17	PU18	PU19	PU20	Run Cost	Start-up Cost
1	455	245	0	0	0	0	0	0	0	0	455	245	0	0	0	0	0	0	0	0	28,653.69	0
2	455	295	0	0	0	0	0	0	0	0	455	295	0	0	0	0	0	0	0	0	32,528.58	0
3	455	382	0	0	25	0	0	0	0	0	455	382	0	0	0	0	0	0	0	0	37,322.27	900
4	455	455	0	0	40	0	0	0	0	0	455	455	0	0	40	0	0	0	0	0	41,733.66	900
5	455	455	130	0	25	0	0	0	0	0	455	455	0	0	25	0	0	0	0	0	41,492.05	560
6	455	425	130	130	25	0	0	0	0	0	455	425	130	0	25	0	0	0	0	0	44,242.54	2220
7	455	455	130	130	45	0	0	0	0	0	455	455	130	0	45	0	0	0	0	0	47,235.77	0
8	455	455	130	130	30	0	0	0	0	0	455	455	130	130	30	0	0	0	0	0	50,227.23	1100
9	455	455	130	130	95	20	25	0	0	0	455	455	130	130	95	0	25	0	0	0	52,963.06	1200
10	455	455	130	130	162	33	25	10	10	0	455	455	130	130	162	33	25	0	0	0	58,143.56	640
11	455	455	130	130	162	73	25	10	10	10	455	455	130	130	162	73	25	10	0	0	59,835.27	120
12	455	455	130	130	162	80	25	43	10	10	455	455	130	130	162	80	25	43	10	10	61,780.33	120
13	455	455	130	130	162	33	25	10	0	0	455	455	130	130	162	33	25	0	10	0	58,161.73	0
14	455	455	130	130	98	20	25	0	0	0	455	455	130	130	98	20	0	0	0	0	52,417.18	0
15	455	455	130	130	30	0	0	0	0	0	455	455	130	130	30	0	0	0	0	0	47,525.06	0
16	455	310	130	130	25	0	0	0	0	0	455	310	130	130	25	0	0	0	0	0	41,182.77	0
17	455	260	130	130	25	0	0	0	0	0	455	260	130	130	25	0	0	0	0	0	49,532.78	0
18	455	338	130	130	25	0	25	0	0	0	455	338	130	130	25	20	0	0	0	0	42,839.79	0
19	455	438	130	130	25	0	25	0	0	0	455	438	130	130	25	20	0	0	0	0	48,711.38	0
20	455	455	130	130	162	33	25	10	10	0	455	455	130	130	162	33	25	0	0	0	56,712.77	680
21	455	455	130	130	95	20	25	0	0	0	455	455	130	130	95	0	25	0	0	0	52,091.11	0
22	455	455	130	0	103	20	0	0	0	0	455	455	0	0	103	0	25	0	0	0	43,718.41	0
23	455	433	0	0	25	0	0	0	0	0	455	433	0	0	0	0	0	0	0	0	33,447.13	0
24	455	345	0	0	0	0	0	0	0	0	455	345	0	0	0	0	0	0	0	0	32,621.89	0
Total operational cost = $1,115,120										Total start-up cost = $8440										Total cost = 1,123,560$		

Table 14 Comparison of results obtained for 20-unit system without ramp rate constraints

Method	Best cost ($) (optimality gap)	Average cost ($) (optimality gap)	Worst cost ($) (optimality gap)
LCA-PSO [27]	1,139,005 (1.3984)	NA	NA
EP [22]	1,126,494 (0.2846)	1,127,257 (0.3525)	1,129,793 (0.5782)
LR [15]	1,128,362 (0.4509)	1,128,395 (0.4538)	1,128,444 (0.4582)
ESA [20]	1,126,254 (0.2632)	1,127,955 (0.4147)	1,129,112 (0.5176)
BDE [40]	1,126,998 (0.3295)	1,127,374 (0.3629)	1,127,927 (0.4122)
BPSO [21]	1,128,192 (0.4358)	1,128,213 (0.4376)	1,128,403 (0.4546)
IBPSO [11]	1,196,029 (6.4748)	NA	NA
GA [19]	1,126,243 (0.2622)	1,128,790 (0.4890)	1,132,059 (0.7800)
BFWA [42]	1,124,658 (0.1211)	1,124,941 (0.1464)	1,125,087 (0.1594)
BGWA [13]	1,125,546 (0.2002)	1,126,126 (0.2518)	1,127,393 (0.3646)
HGICA [25]	1,124,565 (0.1129)	1,124,933 (0.1456)	1,125,147 (0.1647)
BWOA	**1,123,560 (0.0234)**	**1,123,978 (0.0600)**	**1,124,186 (0.0791)**

Bold values indicate the best results

mentioned techniques. The solutions (i.e., best, average, and worst cost) obtained from BFWA are comparable with HGICA. The best cost obtained from BFWA is $1,124,658, which is slightly higher than HGICA. The optimality gap is also reported in Table 14. The optimal solution value obtained from CPLEX is $1,123,297. The solution obtained from the proposed approach is within 0.07% of optimality. The results reveal that the BWOA provides better cost as compared with other reported techniques.

Case-II: 40-unit test system

The second test system consists of 40-unit system. For 40-unit test system, the characteristics of 10-unit test system were replicated twice and load demand was multiplied by 4. The optimal commitment and generation schedules over 24-hour time interval are mentioned in Tables 15 and 16 respectively. The operational and start-up costs obtained from proposed BWOA are $2,205,546 and $18380, respectively.

Table 17 shows the performance comparison of proposed BWOA with other well-established techniques reported in the literature. The “NA” mentioned in this table means that the values are not available in the literature. It is observed from table that the best cost obtained from BWOA is $2,223,926, which is far better than the other techniques. The average and worst cost obtained from BWOA are $2,228,753 and $2,231,547, respectively. These values are much better than the best value of other existing techniques. Due to better trade-off between exploration and exploitation, the BWOA provide better quality solution for unit commitment problem. The best cost obtained with HGICA algorithm for 40-unit system is $2,239,186, which is $15260 more than that of proposed approach. It can be seen from Table 17 that HGICA is the second-best algorithm among the above-mentioned techniques. The solutions obtained from BPSO are comparable with IBPSO. The best cost obtained from BPSO is $2,243,210, which is $4024 more than that of HGICA. The results reveal that the BWOA provides better cost as compared with other reported techniques.

5.3.3 Test system 3: large-scale power system

Large-scale power systems are tested for two different cases. The 80-unit and 100-unit test systems are taken into consideration for this power system. The scheduling periods are 24 h for these power systems. The ramp rate constraints are not considered for this test system. The proposed approach is compared with well-known techniques such as local convergence-averse binary particle swarm optimization (LCA-PSO) [27], evolutionary programming (EP) [22], Lagrangian relaxation (LR) [15], enhanced simulated annealing (ESA) [20], binary differential evolution (BDE) [40], binary PSO (BPSO) [21], improved binary PSO (IBPSO) [11], genetic algorithm (GA) [19], binary fireworks algorithm (BFWA) [42], binary grey wolf algorithm (BGWA) [13], and hybrid genetic imperialist competitive algorithm (HGICA) [25].

Case-I and II: 80 and 100-unit test system

For 80- and 100-unit test systems, the proposed BWOA is compared with other techniques which are reported in Table 18. The “NA” mentioned in this table indicates that the values are not available in the literature. For 80-unit system, the best cost obtained from BWOA is $4,478,412. The best cost obtained from HGICA is $4,485,936, which is $7524 more than that of proposed approach. HGICA is the second-best algorithm among the above-mentioned techniques. The solutions obtained from IBPSO are comparable with BDE. For 100-unit system, the best cost

Table 15 Commitment schedule of 40-unit power system using BWOA

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
PU1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
PU2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
PU3	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1
PU4	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1
PU5	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
PU6	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	1	1	0	0	0	0
PU7	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	0	0	0	0
PU8	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0
PU9	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0
PU10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PU11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
PU12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
PU13	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
PU14	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
PU15	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
PU16	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	0	0	0
PU17	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	1	1	1	0	0	0
PU18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PU19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
PU20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
PU21	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
PU22	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
PU23	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1
PU24	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
PU25	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
PU26	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	0	0	0
PU27	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0
PU28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
PU29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
PU30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PU31	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
PU32	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
PU33	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1
PU34	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	1	1	1	1	1	0
PU35	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
PU36	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1	1	1	0	0	0
PU37	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1	0	0	0	0
PU38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PU39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PU40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

obtained from BWOA is $5,599,281. The best cost obtained from HGICA is $5,604,022, which is $4741 more than that of proposed approach. The results reveal that the BWOA provides better cost as compared with other reported techniques for both 80- and 100-unit systems. It is also observed that HGICA is the second-best algorithm for these test systems.

5.4 Effect of ramp rate constraints

To devise the power system more realistic, the ramp rate constraints are integrated in the formulation of UCP. By including ramp rate constraints, the minimum and maximum generating power of each unit can be computed thorough Eqs. (25) and (26), respectively. According to

Table 16 Unit output power of 40-unit power system using BWOA

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
PU1	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455
PU2	326	393	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	376	438
PU3	0	0	0	130	130	130	130	130	130	130	130	130	130	130	130	0	0	0	0	0	130	130	130	130
PU4	0	0	0	0	130	130	130	130	130	130	130	130	130	130	130	0	0	0	0	0	130	130	130	130
PU5	0	0	108	121	139	141	126	144	130	162	162	162	162	158	128	140	90	162	162	162	162	0	0	0
PU6	0	0	0	0	0	0	0	0	0	0	80	80	62	0	0	0	0	37	71	80	0	0	0	0
PU7	0	0	0	0	0	0	0	0	0	68	58	61	25	0	0	0	0	25	25	77	0	0	0	0
PU8	0	0	0	0	0	0	0	0	0	0	0	55	0	0	0	0	0	0	0	55	0	0	0	0
PU9	0	0	0	0	0	0	0	0	0	0	0	55	0	0	0	0	0	0	0	55	0	0	0	0
PU10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PU11	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455
PU12	326	393	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	376	438
PU13	0	0	0	0	0	130	130	130	130	130	130	130	130	130	130	0	0	0	0	0	0	0	0	0
PU14	0	0	0	0	0	130	130	130	130	130	130	130	130	130	130	0	0	0	0	0	0	0	0	0
PU15	0	0	108	121	139	141	126	144	130	162	162	162	162	158	128	140	90	162	162	162	162	0	0	0
PU16	0	0	0	0	0	0	0	0	0	0	80	80	62	0	0	0	0	0	71	80	79	0	0	0
PU17	0	0	0	0	0	0	0	0	0	68	58	61	0	0	0	0	0	0	25	77	25	0	0	0
PU18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PU19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	55	0	0	0	0
PU20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	21	0	0	0	0
PU21	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455
PU22	326	393	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	376	438
PU23	0	0	0	0	0	130	130	130	130	130	130	130	130	130	0	0	0	0	0	130	130	130	130	130
PU24	0	0	0	0	0	0	130	130	130	130	130	130	130	130	130	0	0	0	0	0	0	0	0	0
PU25	0	0	0	121	139	141	126	144	130	162	162	162	162	158	128	140	90	162	162	162	162	0	0	0
PU26	0	0	0	0	0	0	0	0	0	0	80	80	62	0	0	0	0	0	71	80	79	0	0	0
PU27	0	0	0	0	0	0	0	0	0	68	58	61	0	0	0	0	0	25	25	77	0	0	0	0
PU28, PU29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	55	0	0	0	0
PU30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PU31	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	455	0
PU32	0	0	0	0	0	0	0	0	455	455	455	455	455	455	455	455	455	455	455	455	455	455	0	0
PU33	0	0	0	0	0	0	130	130	130	130	130	130	130	130	0	0	0	0	0	130	130	130	130	130
PU34	0	0	0	0	0	0	0	130	130	130	130	130	130	0	0	0	0	0	130	130	130	130	130	0
PU35	0	0	0	121	139	141	126	144	130	162	162	162	162	158	128	140	90	162	162	162	162	110	0	0
PU36	0	0	0	0	0	0	0	0	0	0	0	80	62	20	0	0	0	0	71	80	79	0	0	0

Table 16 (continued)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
PU37	0	0	0	0	0	0	0	0	0	68	58	61	0	0	0	0	0	25	25	77	0	0	0	0
PU38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
PU39, PU40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Total operational cost =2,205,546 $										Total start-up cost = 18,380$										Total cost = 2,223,926$				

literature, the values of RD_i and RU_i are same and is to be taken as the 20% of maximum capacity of unit at each hour [20, 43]. Hence, the values of RD_i and RU_i for units in 10-unit system are set to 91, 91, 26, 26, 26, 33, 17, 16, 11, 11, and 11, respectively. The output power of generating unit is limited due to its ramp rate constraints. Therefore, the generation cost will be increases as compared to the cost without ramp rate constraints. The effectiveness of proposed BWOA is tested on 10, 20, 40, 80, and 100 units having ramp rate constraints. Table 19 shows the effect of ramp rate constraints on the performance of BWOA. The best cost obtained from BWOA is \$565,771, which is \$1835 more than the cost without ramp constraints. It can be inferred from table that the BWOA is able to handle the ramp rate constraints.

5.5 Complex power systems

In order to demonstrate the robustness of BWOA, it is tested on two complex power systems. These are IEEE 118-bus system and a real Taiwan Power 38-unit system.

5.5.1 IEEE 118-bus system

IEEE 118-bus system consists of 54 generating units. The generator data and load profile are taken from [46]. The comparison results of proposed technique with other existing techniques in terms of total operating cost are given in Table 20. It is inferred from table that the total cost obtained from BWOA is \$1,643,089, which is far better than the other techniques. Due to better trade-off between exploration and exploitation, the BWOA provide better quality solution for IEEE 118-bus system problem. The cost obtained with HGICA algorithm for this system is \$1,644,115, which is \$1026 more than that of proposed approach. It can be concluded that HGICA is the second-best algorithm among the other techniques. The cost obtained from BOWA is \$1,644,128 which is \$13 more than that of HGICA. The results reveal that the BWOA provides better cost as compared with other reported techniques.

5.5.2 Taiwan power system

The Taiwan Power 38-unit system is taken from [47]. The same conditions have been considered to execute the UCP, which are mentioned in [47]. Table 21 shows the comparison between proposed method and other existing techniques. It is observed from Table 21 that the proposed BWOA provides best solution as compared to existing techniques. The best cost obtained from BWOA is \$195,904,123. The best cost obtained from HGICA is \$195,912,018, which is \$7895 more than that of proposed approach. HGICA is the second-best algorithm among the above-mentioned techniques. It can be concluded that the BWOA is able to solve Taiwan power system more efficiently than the other existing techniques.

Table 17 Comparison for results obtained from 40-unit system without ramp rate constraints

Method	Best cost ($)	Average cost ($)	Worst cost ($)
LCA-PSO [27]	2,277,396	NA	NA
EP [22]	2,249,093	2,252,612	2,256,085
LR [15]	2,258,503	2,258,503	2,258,503
ESA [20]	2,250,012	2,252,125	2,254,539
BDE [40]	2,245,700	2,246,600	2,247,284
BPSO [21]	2,243,210	2,244,634	2,245,982
IBPSO [11]	2,243,728	NA	NA
GA [19]	2,252,909	2,262,585	2,269,282
BFWA [42]	2,248,228	2,248,572	2,248,645
BGWA [13]	2,252,475	2,257,866	2,263,333
HGICA [25]	2,239,186	2,242,612	2,246,085
BWOA	**2,223,926**	**2,228,753**	**2,231,547**

Bold values indicate the best results

6 Conclusions

In this paper, a novel binary version of the whale optimization algorithm namely BWOA has been proposed, which utilizes the concept of transfer function. The proposed approach has been tested on a set of various benchmark test functions. It has been compared with six well-known metaheuristic techniques. The statistical test has also been carried out to prove the statistical significance of proposed approach. The experimental results reveal that BWOA outperforms the existing metaheuristic algorithms. Furthermore, BWOA has been applied on unit commitment problem. The proposed approach has been tested on 4, 10, 20, 40, 80, and 100 power units. It has also been tested on IEEE 118-bus system and Taiwan 38 bus system. The simulation results show that the proposed method is effective for solving UCP. The total production costs over schedule time obtained from proposed BWOA are less expensive than other existing methods.

Table 18 Comparison of results obtained for 80- and 100-unit systems without ramp rate constraints

Method	80-unit system			100-unit system		
	Best cost ($)	Average cost ($)	Worst cost ($)	Best cost ($)	Average cost ($)	Worst cost ($)
LCA-PSO [27]	4,554,346	NA	NA	5,706,201	NA	NA
EP [22]	4,498,479	4,505,536	4,512,739	5,623,885	5,633,800	5,639,148
LR [15]	4,526,022	4,526,022	4,526,022	5,657,277	5,657,277	5,657,277
ESA [20]	4,498,076	4,501,156	4,503,987	5,617,876	5,624,301	5,628,506
BDE [40]	4,489,022	4,490,456	4,491,262	5,609,341	5,609,984	5,610,608
BPSO [21]	4,491,083	4,491,681	4,492,686	5,610,293	5,611,181	5,612,265
IBPSO [11]	4,488,351	NA	NA	5,608,792	NA	NA
GA [19]	4,507,692	4,525,204	4,552,982	5,626,361	5,669,362	5,690,086
BFWA [42]	4,491,284	4,492,550	4,493,036	5,610,954	5,612,422	5,612,790
BGWA [13]	4,495,173	4,506,362	4,513,026	5,628,975	5,637,659	5,643,899
HGICA [25]	4,485,936	4,487,958	4,489,283	5,604,022	5,608,561	5,613,260
BWOA	**4,478,412**	**4,479,584**	**4,480,924**	**5,599,281**	**5,601,975**	**5,603,122**

Bold values indicate the best results

Table 19 Performance evaluation of proposed approach with ramp rate constraints

Number of power units	Cost ($)		
	Best cost ($)	Average cost ($)	Worst cost ($)
10	565,771	565,873	565,978
20	1,126,625	1,127,031	1,127,640
40	2,235,113	2,236,342	2,236,937
80	4,490,172	4,491,236	4,491,882
100	5,614,038	5,616,185	5,619,514

Table 20 Comparison of results obtained from IEEE 118-bus system

Method	Operating cost ($)
Semi-definite programming (SDP) [46]	1,645,445.00
Artificial bee colony Lagrangian relaxation (ABC-LR) [48]	1,644,269.00
Binary real-coded firefly algorithm (BRCFF) [49]	1,644,141.00
Firefly algorithm with multiple workers (FAMW) [50]	1,644,134.00
Binary grey wolf algorithm (BGWA) [13]	1,644,128.00
Hybrid genetic imperialist competitive algorithm HGICA [25]	1,644,115.00
Binary whale optimization (BWOA)	**1,643,089.00**

Bold values indicate the best results

Table 21 Comparison of results obtained from Taiwan Power 38-unit system

Method	Minimum operating cost (million $)
Dynamic programming (DP) [47]	210.500000
Lagrangian relaxation (LR) [47]	209.000000
Simulated annealing (SA) [47]	207.800000
Real-coded genetic algorithm (RGA) [51]	204.600000
Fuzzy adaptive particle swarm optimization (FAPSO) [52]	196.730000
Binary real-coded firefly algorithm (BRCFF) [49]	195.916304
Firefly algorithm with multiple workers (FAMW) [50]	195.914900
Binary grey wolf algorithm (BGWA) [13]	195.914495
Hybrid genetic imperialist competitive algorithm HGICA [25]	195.912018
Binary whale optimization (BWOA)	195.904123

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Appendix 1

See Fig. 4 and Tables 22, 23, 24, 25, 26, 27, 28 and 29.

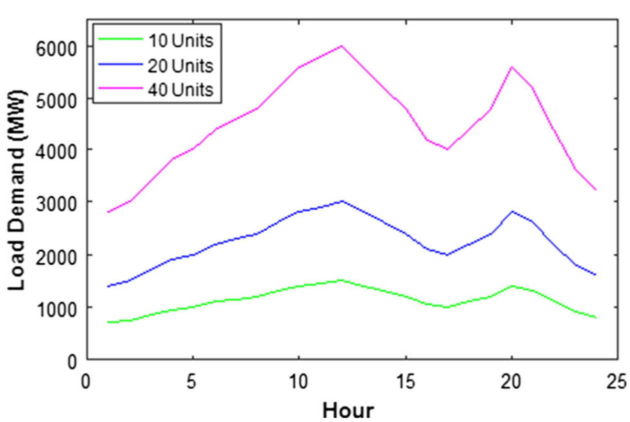

Fig. 4 Load demand pattern for power 10, 20, and 40-unit test systems

Table 22 Unimodal benchmark test functions

Benchmark test function	D	Range	$F_{\min}$
$F_1(Y) = \sum_{i=1}^{D} y_i^2$	5	[−100, 100]	0
$F_2(Y) = \sum_{i=1}^{D} \lvert y_i \rvert + \prod_{i=1}^{D} \lvert y_i \rvert$	5	[−10, 10]	0
$F_3(Y) = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} y_j \right)^2$	5	[−100, 100]	0
$F_4(Y) = \max_i \{ \lvert y_i \rvert,\ 1 \le i \le D \}$	5	[−100, 100]	0
$F_5(Y) = \sum_{i=1}^{D-1} \left[100 \left(y_{i+1} - y_i^2 \right)^2 + (y_i - 1)^2 \right]$	5	[−30, 30]	0
$F_6(Y) = \sum_{i=1}^{D} \left(\lfloor y_i + 0.5 \rfloor \right)^2$	5	[−100, 100]	0
$F_7(Y) = \sum_{i=1}^{D} i y_i^4 + random[0, 1)$	5	[−1.28, 1.28]	0

Table 23 Multimodal benchmark test functions

Objective function	D	Range	$F_{\min}$
$F_8(Y) = -\frac{1}{D}\sum_{i=1}^{D}\left(y_i \sin\left(\sqrt{\lvert y_i\rvert}\right)\right)$	5	[−500, 500]	− 2094.914
$F_9(Y) = \sum_{i=1}^{D}\left[y_i^2 - 10\cos(2\pi y_i) + 10\right]$	5	[−5.12, 5.12]	0
$F_{10}(Y) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} y_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi y_i)\right) + 20 + e$	5	[−32, 32]	0
$F_{11}(Y) = \frac{1}{4000}\sum_{i=1}^{D} y_i^2 - \prod_{i=1}^{D}\cos\left(\frac{y_i}{\sqrt{i}}\right) + 1$	5	[−600, 600]	0
$F_{12}(Y) = \frac{\pi}{D}\left\{10\sin(\pi z_1) + \sum_{i=1}^{D-1}(z_i - 1)^2\left[1 + 10\sin^2(\pi z_{i+1})\right] + (z_D - 1)^2\right\} + \sum_{i=1}^{D} u(y_i, 10, 100, 4)$ *where* $u(y_i, a, k, m) = \begin{cases} k(y_i - a)^m, & y_i > a, \\ 0, & -a \le y_i \le a, \\ k(-y_i - a)^m, & y_i < -a \end{cases}$ $z_i = 1 + \frac{1}{4}(y_i + 1)$	5	[−50, 50]	0
$F_{13}(Y) = 0.1\left\{\sin^2(3\pi y_1) + \sum_{i=1}^{D-1}(y_i - 1)^2\left[1 + \sin^2(3\pi y_{i+1})\right] + (y_D - 1)^2\left[1 + \sin^2(2\pi y_D)\right]\right\} + \sum_{i=1}^{D} u(y_i, 5, 100, 4)$	5	[−30, 30]	0

Table 24 Multimodal benchmark test functions with fixed dimension

Objective function	D	Range	F_{min}
$F_{14}(Y) = \left(\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j + \sum_{i=1}^{2}\left(x_i - a_{ij}\right)^6}\right)^{-1}$	2	[−65, 65]	1
$F_{15}(Y) = \sum_{i=1}^{11}\left[a_i - \frac{y_1\left(b_i^2 + b_i y_2\right)}{b_i^2 + b_i y_3 + y_4}\right]^2$	4	[−5, 5]	0.00030
$F_{16}(Y) = 4y_1^2 - 2.1y_1^4 + \frac{1}{3}y_1^6 + y_1 y_2 - 4y_2^2 + 4y_2^4$	2	[−5, 5]	− 1.0316
$F_{17}(Y) = \left(y_2 - \frac{5.1}{4\pi^2}y_1^2 + \frac{5}{\pi}y_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos y_1 + 10$	2	[−5, 5]	0.398

Table 24 (continued)

Objective function	D	Range	F_{min}
$F_{18}(Y)=\left[1+(y_1+y_2+1)^2\left(19-14y_1+6y_1y_2+3y_2^2\right)\right] \times\left[30+(2y_1-3y_2)^2\times\left(18-32y_1+12y_1^2+48y_2-36y_1y_2+27y_2^2\right)\right]$	2	[−2, 2]	3
$F_{19}(Y)=-\sum_{i=1}^{4}c_i\exp\left(-\sum_{j=1}^{3}a_{ij}\left(x_j-p_{ij}\right)^2\right)$	3	[1, 3]	− 3.86
$F_{20}(Y)=-\sum_{i=1}^{4}c_i\exp\left(-\sum_{j=1}^{6}a_{ij}\left(x_j-p_{ij}\right)^2\right)$	6	[0, 1]	− 3.32
$F_{21}(Y)=-\sum_{i=1}^{5}\left[(Y-a_i)(Y-a_i)^T+c_i\right]^{-1}$	4	[0, 10]	− 10.1532
$F_{22}(Y)=-\sum_{i=1}^{7}\left[(Y-a_i)(Y-a_i)^T+c_i\right]^{-1}$	4	[0, 10]	− 10.4028
$F_{23}(Y)=-\sum_{i=1}^{10}\left[(Y-a_i)(Y-a_i)^T+c_i\right]^{-1}$	4	[0, 10]	− 10.5363

Table 25 Composite benchmark test functions

Objective function	D	Range	$F_{\min}$
$F_{24}(CF1)$: $f_1, f_2, f_3, \ldots, f_{10}=$ Sphere Function $[\sigma_1,\ \sigma_2,\ \sigma_3, \ldots,\ \sigma_{10}]=[1,1,1,\ldots,1]$ $[\lambda_1,\ \lambda_2,\ \lambda_3,\ldots,\lambda_{10}]=[5/100,5/100,5/100,\ldots,5/100]$	10	[−5, 5]	0
$F_{25}(CF2)$: $f_1,f_2,f_3,\ldots,f_{10}=$ Griewank's function $[\sigma_1,\sigma_2,\sigma_3,\ldots,\sigma_{10}]=[1,1,1,\ldots,1]$ $[\lambda_1,\lambda_2,\lambda_3,\ldots,\lambda_{10}]=[5/100,5/100,\ 5/100,\ldots,5/100]$	10	[−5, 5]	0
$F_{26}(CF3)$: $f_1, f_2, f_3,\ldots,f_{10}=$ Griewank's Function $[\sigma_1,\sigma_2,\sigma_3,\ldots,\ \sigma_{10}]=[1,1,1,\ldots,1]$ $[\lambda_1,\ \lambda_2,\ \lambda_3,\ldots,\lambda_{10}]=[1,\ 1,\ 1,\ldots,1]$	10	[−5, 5]	0
	10	[−5, 5]	0

Table 25 (continued)

Objective function	D	Range	F_{min}
$F_{27}(CF4)$: $f_1, f_2 =$ Ackley's Function, $f_3, f_4 =$ Rastrigin's Function $f_5, f_6 =$ Weierstrass Function, $f_7, f_8 =$ Griewank's Function $f_9, f_{10} =$ Sphere Function $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] = [1, 1, 1, \ldots, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] = [5/32, 5/32, 1, 1, 5/0.5, 5/0.5, 5/100, 5/100, 5/100, 5/100]$	10	$[-5, 5]$	0
$F_{28}(CF5)$: $f_1, f_2 =$ Rastrigin's Function, $f_3, f_4 =$ Weierstrass Function $f_5, f_6 =$ Griewank's Function, $f_7, f_8 =$ Ackley's Function $f_9, f_{10} =$ Sphere Function $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] = [1, 1, 1, \ldots, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] = [1/5, 1/5, 5/0.5, 5/0.5, 5/100, 5/100, 5/32, 5/32, 5/100, 5/100]$	10	$[-5, 5]$	0
$F_{29}(CF6)$: $f_1, f_2 =$ Rastrigin's Function, $f_3, f_4 =$ Weierstrass Function $f_5, f_6 =$ Griewank's Function, $f_7, f_8 =$ Ackley's Function $f_9, f_{10} =$ Sphere Function $[\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{10}] = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]$ $[\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_{10}] = \begin{bmatrix} 0.1 \times 1/5, 0.2 \times 1/5, 0.3 \times 5/0.5, 0.4 \times 5/0.5, 0.5 \times 5/100, \\ 0.6 \times 5/100, 0.7 \times 5/32, 0.8 \times 5/32, 0.9 \times 5/100, 1 \times 5/100 \end{bmatrix}$			

Table 26 Load demand for 4-power unit system (MW)

Hour	1	2	3	4	5	6	7	8
Demand	450	530	600	540	400	280	290	500

Table 27 Test data for 4-power unit system

Unit	P_i^{Max} (MW)	P_i^{Min} (MW)	a (\$/MW^2)	b (\$/MW^2)	c (\$/MW^2)	MUT (h)	MDT (h)	HSC (\$)	CSC (\$)	CSH (h)	IS (h)
PU1	300	75	684.74	16.83	0.0021	5	4	500	1100	5	8
PU2	250	60	585.62	16.95	0.0042	5	3	170	400	5	8
PU3	80	25	213	20.74	0.0018	4	2	150	350	4	− 5
PU4	60	20	252	23.6	0.0034	1	1	0	0.02	0	− 6

Table 28 Load demand for 10-power unit system (MW)

Hour	1	2	3	4	5	6	7	8	9	10	11	12
Demand	700	750	850	950	1000	1100	1150	1200	1300	1400	1450	1500
Hour	13	14	15	16	17	18	19	20	21	22	23	24
Demand	1400	1300	1200	1050	1000	1100	1200	1400	1300	1100	900	800

Table 29 Test data for 10-power unit system

Unit	P_i^{Max} (MW)	P_i^{Min} (MW)	a (\$/MW^2)	b (\$/MW^2)	c (\$/MW^2)	MUT (h)	MDT (h)	HSC (\$)	CSC (\$)	CSH (h)	IS (h)
PU1	455	150	1000	16.19	0.00048	8	8	4500	9000	5	8
PU2	455	150	970	17.26	0.00031	8	8	5000	10000	5	8
PU3	130	20	700	16.60	0.0002	5	5	550	1100	4	− 5
PU4	130	20	680	16.50	0.00211	5	5	560	1120	4	− 5
PU5	162	25	450	19.70	0.00398	6	6	900	1800	4	− 6
PU6	80	20	370	22.26	0.00712	3	3	260	520	2	− 3
PU7	85	25	480	27.74	0.00079	3	3	260	520	2	− 3
PU8	55	10	660	25.92	0.00413	1	1	30	60	0	− 1
PU9	55	10	665	27.27	0.00222	1	1	30	60	0	− 1
PU10	55	10	670	27.29	0.00173	1	1	30	60	0	− 1

References

1. Goldberg D (1989) Genetic algorithms in search, optimization, and machine. Addison-Wesley, Boston
2. Kennedy N, Eberhart RC (1995) Particle swarm optimization. In: Proceedings of IEEE international conference on neural networks, Perth, pp 1942–1948
3. Dorigo M, Birattari M, Stutzle T (2006) Ant colony optimization. IEEE Comput Intell Mag 1(4):28–35
4. Yang X-S, Gandomi AH (2012) Bat algorithm: a novel approach for global engineering optimization. Eng Comput 29(5):464–483
5. Rashedi E, Nezamabadi-pour H, Saryazdi S (2009) GSA: a gravitational search algorithm. Inf Sci 179:2232–2248
6. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evolut Comput 1:67–82
7. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
8. Wang X, Yang J, Teng X, Xia W, Jensen R (2007) Feature selection based on rough sets and particle swarm optimization. Pattern Recognit Lett 28:459–471
9. Srinivasa KG, Venugopal KR, Patnaik LM (2007) A self-adaptive migration model genetic algorithm for data mining applications. Inf Sci 177(20):4295–4313
10. Wu T-H, Chang C-C, Chung S-H (2008) A simulated annealing algorithm for manufacturing cell formation problems. Expert Syst Appl 34(3):1609–1617
11. Yuan X, Nie H, Su A, Wang L, Yuan Y (2009) An improved binary particle swarm optimization for unit commitment problem. Expert Syst Appl 36(4):8049–8055
12. Amiri M, Khanohamadi S (2013) A primary unit commitment approach with a modification process. Appl Soft Comput 13:1007–1015
13. Panwar LK, Reddy SK, Verma A, Panigrahi BK, Kumar R (2018) Binary grey wolf optimizer for large scale unit commitment problem. Swarm Evol Comput 38:251–266
14. Senjyu T, Shimabukuro K, Uezato K, Funabashi T (2003) A fast technique for unit commitment problem by extended priority list. IEEE Trans Power Syst 18(2):882–888
15. Ongsakul W, Petcharaks N (2004) Unit commitment by enhanced adaptive Lagrangian relaxation. IEEE Trans Power Syst 19:620–628
16. Venkatesh B, Jamtsho T, Gooi HB (2007) Unit commitment—a fuzzy mixed integer linear programming solution. IET Gener Transm Distrib 1(5):836–846
17. Cohen AI, Yoshimura M (1983) A Branch-and-Bound algorithm for unit commitment. IEEE Trans Power Appar Syst 102(2):444–451
18. Su C-C, Hsu Y-Y (1991) Fuzzy dynamic programming: an application to unit commitment. IEEE Trans Power Syst 6(3):1231–1237
19. Kazarlis SA, Bakirtzis AG, Petridis V (1996) A genetic algorithm solution to the unit commitment problem. IEEE Trans Power Syst 11(1):83–92
20. Simopoulos DN, Kavatza SD, Vournas CD (2006) Unit commitment by an enhanced simulated annealing algorithm. IEEE Trans Power Syst 21(1):68–76
21. Jeong YW, Park JB, Jang SH, Lee KY (2009) A new quantum inspired binary PSO for thermal unit commitment problems. In: Proceedings of IEEE international conference on intelligent system applications to power systems, Curitiba, Brazil, pp 1–6
22. Juste KA, Kita H, Tanaka E, Hasegawa J (1999) An evolutionary programming solution to the unit commitment problem. IEEE Trans Power Syst 14(4):1452–1459
23. Hadji MM, Vahidi B (2012) A solution to the unit commitment problem using imperialistic competition algorithm. IEEE Trans Power Syst 27(1):117–124
24. Balci H, Valenzuela J (2004) Scheduling electric power generations using particle swarm optimization combined with the Lagrangian relaxation method. Int J Appl Math Comput Sci 14(3):411–421

25. Saber NA, Salimi M, Mirabbbasi D (2016) A priority list based approach for solving thermal unit commitment problem with novel hybrid genetic-imperialist competitive algorithm. Energy 117(1):272–280
26. Kamboj VK, Bath SK, Dhillon JS (2016) Implementation of hybrid harmony search/random search algorithm for single area unit commitment problem. Int J Electr Power Energy Syst 77:228–249
27. Wang B, Li Y, Watada J (2011) Re-scheduling the unit commitment problem in fuzzy environment. In: Proceedings of IEEE international conference on fuzzy systems, Taipei, Taiwan, pp 1090–1095
28. Kamboj VK (2016) A novel hybrid PSO–GWO approach for unit commitment problem. Neural Comput Appl 27(6):1643–1655
29. Mallipeddi R, Suganthan PN (2014) Unit commitment—a survey and comparison of conventional and nature inspired algorithms. Int J BioInspir Comput 6(2):71–90
30. Rashedi E, Nezamabadi-pour H, Saryazdi S (2010) BSGA: binary gravitational search algorithm. Nat Comput 9(3):727–745
31. Hussien AG, Houssein EH, Hassanien AE (2017) A binary whale optimization algorithm with hyperbolic tangent fitness function for feature selection. In: Proceedings of IEEE international conference on intelligent computing and information systems, Egypt, pp 166–172
32. Eid HF (2017) Binary whale optimization: an effective swarm algorithm for feature selection. Int J Metaheuristics 7(1):67–79
33. Digalakis JG, Margaritis KG (2002) An experimental study of benchmarking functions for genetic algorithms. Int J Comput Math 79(4):403–416
34. Liang JJ, Suganthan PN, Deb K (2005) Novel composition test functions for numerical global optimization. In: Proceedings of IEEE swarm intelligence symposium, Pasadena, CA, pp 68–75
35. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–681
36. Kennedy J, Eberhart RC (1997) A discrete binary version of the particle swarm algorithm. In: Proceedings of IEEE international conference on computational cybernetics and simulation, Orlando, pp 4104–4108
37. Mirjalili S, Mirjalili SM, Yang X-S (2014) Binary bat algorithm. Neural Comput Appl 25(3–4):663–681
38. Mirjalili S (2016) Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural Comput Appl 27(4):1053–1073
39. Emary E, Zawbaa HM, Hassanien AE (2016) Binary grey wolf optimization approaches for feature selection. Neurocomputing 172:371–381
40. Jeong Y-W, Lee W-N, Kim H-H, Park J-B, Shin J-R (2009) Thermal unit commitment using binary differential evolution. J Electr Eng Technol 4(3):323–329
41. Sriyanyong P, Song YH (2005) Unit commitment using particle swarm optimization combined with Lagrange relaxation. In: IEEE power engineering society general meeting, San Francisco, USA, pp 1–8
42. Panwar LK, Reddy SK, Kumar R (2015) Binary fireworks algorithm based thermal unit commitment. Int J Swarm Intell Res 6(2):87–101
43. Singhal PK, Naresh R, Sharma V (2015) A novel strategy-based hybrid binary artificial bee colony algorithm for unit commitment problem. Arab J Sci Eng 40(5):1455–1469
44. Khanmohammadi S, Amiri M, TarafdarHaque M (2010) A new three-stage method for solving unit commitment problem. Energy 35(7):3072–3080
45. Roque LAC (2016) Optimization methods for the unit commitment problem in electric power systems. Dissertation, University of Porto
46. Bai X, Wei H (2009) Semi-definite programming-based method for security-constrained unit commitment with operational and optimal power flow constraints. IET Gener Transm Distrib 3(2):182–197
47. Huang KY, Yang HT, Huang CL (1998) A new thermal unit commitment approach using constraint logic programming. IEEE Trans Power Syst 13(3):936–945
48. Chandrasekaran K, Hemamalini S, Simon SP, Padhy NP (2009) Thermal unit commitment using binary/real coded artificial bee colony algorithm. Electr Power Syst Res 84:109–119
49. Chandrasekaran K, Simon SP (2013) Optimal deviation based firefly algorithm tuned fuzzy design for multi-objective UCP. IEEE Trans Power Syst 28:460–471
50. Koodalsamy B, Veerayan MB, Koodalsamy C, Simon SP (2016) Firefly algorithm with multiple workers for the power system unit commitment. Turk J Electr Eng Comput Sci 24:4773–4789
51. Sun L, Zhang Y, Jiang C (2006) A matrix real-coded genetic algorithm to the unit commitment problem. Elect Power Syst Res 76(9–10):716–728
52. Saber AY, Senjyu T, Yona A, Funabashi T (2007) Unit commitment computation by fuzzy adaptive particle swarm optimization. IET Gener Transm Distrib 1:456–465