Mathematical Biosciences and Engineering

Mathematical Biosciences and Engineering, 16(4): 2775–2794.
DOI: 10.3934/mbe.2019138
Received : 27 January 2019
Accepted: 15 March 2019
Published: 29 March 2019

http://www.aimspress.com/journal/MBE

Research article

A hybrid invasive weed optimization algorithm for the economic load dispatch problem in power systems

Zhi-xin Zheng[1,]*, Jun-qing Li[1,2,3,]* and Hong-yan Sang[1]

[1] College of Computer Science, Liaocheng University, Liaocheng 252059, China
[2] School of information science and engineering, Shandong Normal University, 250014, China
[3] China Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, Nanjing 211189, China

* **Correspondence:** Email: zhengzhixin@lcu-cs.com, lijunqing@lcu-cs.com; Tel: +8613563500220, +8615063528919.

Abstract: In this study, a hybrid invasive weed optimization (HIWO) algorithm that hybridizes the invasive weed optimization (IWO) algorithm and genetic algorithm (GA) has been proposed to solve economic dispatch (ED) problems in power systems. In the proposed algorithm, the IWO algorithm is used as the main optimizer to explore the solution space, whereas the crossover and mutation operations of the GA are developed to significantly improve the optimization ability of IWO. In addition, an effective repair method is embedded in the proposed algorithm to repair infeasible solutions by handing various practical constraints of ED problems. To verify the optimization performance of the proposed algorithm and the effectiveness of the repair method, six ED problems in the different-scale power systems were tested and compared with other algorithms proposed in the literature. The experimental results indicated that the proposed HIWO algorithm can obtain the more economical dispatch solutions, and the proposed repair method can effectively repair each infeasible dispatch solution to a feasible solution. The convergence capability, applicability and effectiveness of HIWO were also demonstrated through the comprehensive comparison results.

Keywords: economic dispatch; hybrid invasive weed optimization; crossover operation; mutation operation; power system

1. Introduction

Economic dispatch (ED) [1] in power systems is an important issue for obtaining the steady-state and economic operations of systems that is a typical constrained optimization problem with multiple variables. The optimization goal of the ED problem is to determine the most economic power outputs of generators while satisfying multiple constraints, such as the generation capacity limits, power demand balance, network transmission losses, ramp rate limits and prohibited operating zones. Considering the valve-point effects (VPE) of multivalve steam turbines for the ED problem, the objective cost function is a nonlinear and nonconvex function, which is hard to solve [2]. Especially in large-scale power systems with multiple generators, the ED problem is a complex optimization problem with several local optimal solutions, and thus the global optimal solution is hard to find.

In recent years, several optimization algorithms, including conventional algorithms and meta-heuristic algorithms, have been proposed to solve the ED problems. Some conventional algorithms, such as linear programming (LP) [3], self-adaptive dynamic programming (SADP) [4], iterative dynamic programming (IDP) [1] and evolutionary programming (EP) [5], have been applied to solve the ED problems. These methods solve the ED problems using the simplified optimization model in which the valve-point effects, ramp rate limits, prohibited operating zones and transmission losses are not considered. Moreover, the optimal results obtained by these methods may be the local optima and have lower computational accuracy. The drawbacks of conventional algorithms prompt researchers to study meta-heuristic algorithms for solving ED problems.

Recently, many meta-heuristic algorithms have been proposed to solve the various optimization problems, such as flow shop scheduling [6–8], steelmaking scheduling [9], job shop scheduling [10–13], flexible task scheduling [14] and chiller loading optimization [15–17]. Due to the better optimization performance, many meta-heuristic algorithms have also been applied to solve the complex ED problems, and these algorithms include the genetic algorithm (GA) [18–21], particle swarm optimization (PSO) and its variants [22–26], firefly algorithm (FA) [27], oppositional real coded chemical reaction optimization (ORCCRO) [28], differential evolution (DE) [29,30], chaotic bat algorithm (CBA) [31], oppositional invasive weed optimization (OIWO) [32], teaching learning based optimization (TLBO) [33], tournament-based harmony search (THS) [34], grey wolf optimization (GWO) [35,36], hybrid artificial algae algorithm (HAAA) [37], orthogonal learning competitive swarm optimizer (OLCSO) [2], backtracking search algorithm (BSA) [38], social spider algorithm (SSA) [39], civilized swarm optimization (CSO) [40], kinetic gas molecule optimization (KGMO) [41] and hybrid methods [42–45]. Although the above meta-heuristic algorithms have been shown to be efficient in solving ED problems, the optimal results obtained by these algorithms are not the most economical.

By mimicking the colonization behavior of weeds in nature, the invasive weed optimization (IWO) algorithm was proposed by Mehrabian and Lucas [46] to optimize multidimensional functions. The experimental results demonstrated that IWO can obtain superior optimization results compared to other evolutionary-based algorithms. Due to its robustness, convergence, high accuracy and searching ability, the IWO algorithm has been applied to solve many engineering optimization problems. However, when IWO is used to solve the ED problem in large-scale power systems, the optimization power outputs of generators obtained by IWO consumes more generation costs compared to the reported methods in literature. To further improve the optimization performance of

IWO in solving ED problems, especially ED problems in the large-scale power systems, inspired by the effective application of hybrid methods in solving ED problems [37,42–45], a hybrid invasive weed optimization (HIWO) algorithm that hybridizes IWO with GA is developed in this study. The motivation behind choosing GA integrated with IWO is to get a better dispatch solution using the crossover operation between offspring weed and its parent weed to improve the local search ability of IWO, and executing the mutation operation on offspring weeds to increase the diversity of the population. The main contributions of this study are as follows: (1) the economic dispatch problem with various practical constraints is investigated by minimizing the total power generation cost; (2) the crossover and mutation operations of GA are proposed to improve the optimization performance of IWO; and (3) an effective repair method of handing constraints is investigated to repair the infeasible dispatch solutions.

The rest of this paper is organized as follows. Section 2 gives the mathematical formulation of the ED problem. Section 3 introduces a hybrid invasive weed optimization (HIWO) algorithm. Section 4 presents the application method of HIWO on ED problems. Section 5 shows the experimental results and analysis on six power systems with different scales. The conclusion is finally given in Section 6.

2. Problem formulation

The ED problem in power systems is to find the optimal dispatch solution of the power outputs of generators, while the total power generation cost of the system is minimized and all the constraints are satisfied.

2.1. Optimization objective

The optimization objective of the ED problem is to minimize the power generation cost (*SC*) consumed by *N* number of generators in the power system, as shown in Eq 1.

$$Min.SC = \sum_{i=1}^{N} C_i(P_i) \tag{1}$$

where P_i and C_i are the power output and generation cost of the *i*th generator, respectively.

For the ED problem neglecting valve-point effects, C_i is calculated by Eq 2. For the ED problem considering valve-point effects, Eq 3 is used to calculate C_i [2,32].

$$C_i(P_i) = a_i \cdot {P_i}^2 + b_i \cdot P_i + c_i \tag{2}$$

$$C_i(P_i) = a_i \cdot {P_i}^2 + b_i \cdot P_i + c_i + \left| e_i \cdot \sin\left(f_i \cdot \left(P_i^{min} - P_i\right)\right)\right| \tag{3}$$

where a_i, b_i and c_i are the cost coefficients of the *i*th generator; e_i and f_i are valve-point coefficients of the *i*th generator; P_i^{min} is the lower limit of P_i.

2.2. Constraints

The feasible dispatch solutions of the ED problem should satisfy the following constraints.

2.2.1. Generation capacity limits

The power output of each generator must be in the range specified by the minimum (P_i^{min}) and maximum (P_i^{max}) of the power output of the ith generator, as shown in Eq 4.

$$P_i^{min} \le P_i \le P_i^{max} \tag{4}$$

2.2.2. Power demand balance

The power outputs of generators should satisfy the system power demand (PD). For the ED problem neglecting network transmission losses (PL), the power demand balance is expressed as Eq 5 [30]. For the ED problem considering PL, the power demand balance is expressed as Eq 6.

$$\sum_{i=1}^{N} P_i = PD \tag{5}$$

$$\sum_{i=1}^{N} P_i = PD + PL \tag{6}$$

PL can be calculated using the power flow analysis method [47] or the B-coefficients method [48]. This study adopts the following B-coefficients method to calculate PL.

$$PL = \sum_{i=1}^{N}\sum_{j=1}^{N} P_i B_{ij} P_j + \sum_{i=1}^{N} B_{0i} P_i + B_{00} \tag{7}$$

where B_{ij}, B_{0i} and B_{00} represent the loss coefficients.

2.2.3. Ramp rate limits

In the actual operation of the power system, to avoid the excessive stress on the boiler and combustion equipment, the change rate of the power output of each generating unit should be within the ramp rate limit, as shown in Eq 8.

$$\begin{cases} P_i - P_i^0 \le UR_i \\ P_i^0 - P_i \le DR_i \end{cases} \tag{8}$$

where P_i^0 is the power output of the ith generator at the previous time interval. UR_i and DR_i represent the upper limits of ramp up and ramp down rate of the ith generator, respectively.

When taking into account both the generation capacity limits and ramp rate limits, the value range of P_i can be rewritten as Eq 9.

$$max\,\{P_i^{min}, P_i^0 - DR_i\} \le P_i \le min\{P_i^{max}, P_i^0 + UR_i\} \tag{9}$$

2.2.4. Prohibited operating zones

Considering the operation limitations of machine components, the power outputs of some generators cannot lie in the prohibited zones, as shown in Eq 10.

$$P_i \in \begin{cases} P_i^{min} \leq P_i \leq P_{i,1}^l \\ P_{i,k-1}^u \leq P_i \leq P_{i,k}^l \\ P_{i,np_i}^u \leq P_i \leq P_i^{max} \end{cases} \qquad k = 2, 3, \cdots, np_i \tag{10}$$

where $P_{i,k}^u$ and $P_{i,k}^l$ represent the upper and lower limits of the *k*th prohibited zone, respectively. np_i is the number of the prohibited zones of P_i.

3. Hybrid invasive weed optimization algorithm

3.1. IWO algorithm

IWO is a novel evolutionary computation algorithm based on weed swarm intelligence. By simulating the propagation and growth behaviors of weeds in nature, IWO searches for the optimal solution of the problem in the solution space. The calculation steps of IWO include initialization, reproduction, spatial dispersal and selection. The initial population with Nw_o weed individuals is randomly generated in the feasible solution space, in which each weed consisting of variables represents a feasible solution. Then, each weed W_j in the population reproduces seeds, and the seeds grow into offspring weeds through spatial dispersal. The amount (Ns_j) of seeds reproduced by W_j is calculated by using Eq 11.

$$Ns_j = \frac{Fit_j - Fit_{min}}{Fit_{max} - Fit_{min}} \cdot (Ns_{max} - Ns_{min}) + Ns_{min} \tag{11}$$

where Fit_j is the fitness value of W_j ; Fit_{min} and Fit_{max} are the minimum and maximum fitness values in the weed population, respectively; Ns_{min} and Ns_{max} are the minimum and maximum of the number of seeds, respectively.

The parent weeds with higher fitness values can reproduce more seeds, and they have more offspring weeds in the population. This reproduction strategy means that IWO can converge rapidly and reliably to the approximate optimal solution. Offspring weeds are randomly distributed around their parent weed according to a normal distribution with a standard deviation (σ_{it}). The calculation formula of σ_{it} is shown in Eq 12. Along with the increase of the iteration times, σ_{it} is gradually reduced from an initial value (σ_{iv}) to a final value (σ_{fv}), which makes the search range of *IWO* be gradually reduced. This strategy makes IWO have the whole space search capability in early iterations and high local convergence in later iterations. After all the seeds grow into weeds, the Nw_{max} weeds with higher fitness values are selected from all the weeds as the parent weeds of the next iteration. Through $Iter_{max}$ times iterations, the weed with the highest fitness value is the optimal solution of the problem.

$$\sigma_{it} = \frac{(Iter_{max} - Iter)^m}{{Iter_{max}}^m} \cdot \left(\sigma_{iv} - \sigma_{fv}\right) + \sigma_{fv} \tag{12}$$

where m is the nonlinear modulation index, and $Iter$ and $Iter_{max}$ are the current number and maximum of iterations, respectively.

3.2. Framework of HIWO

In the proposed HIWO algorithm, IWO is used to explore the solution space around parent weeds. After the seeds reproduced by parent weeds have grown into offspring weeds, the crossover and mutation operations of GA are performed on offspring weeds for improving the quality and diversity of solutions, which can improve the convergence speed and avoid the premature convergence of the algorithm.

```
Begin
      Step 1: Initialize the population with Nw_0 parent weeds
      Step 2: Repair the infeasible parent weeds
      Step 3: Initialize the iteration's counter, Iter=1
   While Iter<= Iter_max
      Step 4: Calculate the quantity (Ns_j) of seeds reproduced by each parent weed (W_j)
      Step 5: Seeds grow into offspring weeds through spatial dispersal
     Step 6: Perform the crossover and mutation operations of GA on each offspring weed
      Step 7: Repair the infeasible offspring weeds
      Step 8: Count the total number Nw_tn of parent weeds and offspring weeds
    If Nw_tn >Nw_max, then
      Step 9: Select Nw_max weeds with higher fitness values as the parent weeds of the next iteration
    Else
      Step 10: All weeds are used as the parent weeds of the next iteration
    End if
      Step 11: Update the iteration's counter, Iter= Iter+1
   End while
      Step 12: Select the weed with the highest fitness value as the optimal solution of the problem
End
```

Figure 1. Pseudo code of the hybrid HIWO algorithm.

3.2.1. Pseudo code of HIWO

The execution flow of HIWO is represented by the pseudo code shown in Figure 1.

3.2.2. Crossover operation

Each offspring weed ($OW_{(j,q)}$) (q=1, 2, …, Ns_j) crosses with its parent weed (W_j) to generate a new weed ($OW'_{(j,q)}$). For each variable P_i (i=1, 2, …, N), calculate the generation cost ($COW^i_{(j,q)}$) consumed by the P_i of $OW_{(j,q)}$ and the generation cost (CW_j^i) consumed by the P_i of W_j, and then determine the value of the P_i of $OW'_{(j,q)}$ according to the following two cases.

(a) If $COW^i_{(j,q)} \leq CW_j^i$, the P_i of $OW'_{(j,q)}$ is equal to that of $OW_{(j,q)}$.

(b) If $COW_{(j,q)}^{i} > CW_{j}^{i}$, the P_i of $OW'_{(j,q)}$ is equal to that of W_j.

After the new offspring weed ($OW'_{(j,q)}$) is generated by the crossover operation, set $OW_{(j,q)}$ = $OW'_{(j,q)}$.

3.2.3. Mutation operation

For each offspring weed ($OW_{(j,q)}$) (q=1, 2, …, Ns_j), randomly select X mutation points from N variables $(P_1, P_2, \cdots, P_N)$, and then modify the P_i of each mutation point using a random number that is distributed around P_i according to a normal distribution with a standard deviation (σ_m). The calculation formula of σ_m is expressed as Eq 13.

$$\sigma_m = \left(P_i^{max} - P_i^{min}\right) \cdot rand(0,1) \tag{13}$$

where *rand* (0, 1) is a random number between 0 and 1.

4. Application of HIWO on ED problem

4.1. Implementation method

In the proposed HIWO algorithm, the first task is the encoding to represent each solution considering all of the constraints. Each weed (W_j) is represented as a row vector consisting of power outputs of generators, as shown in Eq 14. The weed population is initialized by randomly generating the power outputs of generators by using Eq 15. Then, infeasible weeds are repaired into feasible solutions by using the repair method in Section 4.2. Weeds in the initial population are used as the parent weeds to reproduce seeds, which grows into offspring weeds through spatial dispersal. The weeds with higher fitness value can reproduce more seeds. The fitness function used in this study is shown in Eq 16. Each offspring weed will perform the crossover and mutation procedures, like in the canonical GA, and thus can increase the diversity of the population. Then, the repair procedure is applied on the infeasible offspring weeds to make them satisfy with all of the constraints. If the total quantity of parent weeds and offspring weeds is larger than the specified population size, select the weeds with higher fitness values as the parent weeds of the next iteration. Otherwise, all the weeds are used as parent weeds. After multiple times iterations, the best weed with the highest fitness value is selected as the optimal dispatch solution of the ED problem.

$$W_j = (P_1, P_2, \cdots, P_N) \tag{14}$$

$$P_i = \left(P_i^{max} - P_i^{min}\right) \cdot rand(0,1) + P_i^{min} \qquad i = 1, 2, \cdots, N \tag{15}$$

$$Fit_j = \frac{1}{SC_j} \tag{16}$$

where SC_j and Fit_j represent the power generation cost and fitness value of the jth weed, respectively.

4.2. Repair method of infeasible solutions

An effective repair method of handing constraints is proposed in this study to repair infeasible weeds into feasible solutions. The detail repair steps are stated in the following.

Step 1: Modify the P_i $(i = 1, 2, \ldots, N)$ of W_j to satisfy the generation capacity limits and ramp rate limits by using Eq 17. If P_i violates the constraint of prohibited operating zone, update P_i using a upper or lower limit of the prohibited operating zones that is closest to P_i.

$$P_i = \begin{cases} max\,\{P_i^{min}, P_i^0 - DR_i\} & \text{if } P_i < max\,\{P_i^{min}, P_i^0 - DR_i\} \\ min\{P_i^{max}, P_i^0 + UR_i\} & \text{if } P_i > min\{P_i^{max}, P_i^0 + UR_i\} \end{cases} \quad (17)$$

Step 2: Calculate the constraint violation (V) of the power demand balance. For the ED problem considering transmission losses, V is calculated by using Eq 18. For the ED problem neglecting transmission losses, V is calculated by using Eq 19. If $V \neq 0$, go to Step 3. Otherwise, go to Step 5.

$$V = \left| \sum_{i=1}^{N} P_i - PD - \sum_{i=1}^{N}\sum_{j=1}^{N} P_i B_{ij} P_j - \sum_{i=1}^{N} B_{0i} P_i - B_{00} \right| \quad (18)$$

$$V = \left| \sum_{i=1}^{N} P_i - PD \right| \quad (19)$$

Step 3: Determine the modification sequence of N generators. For each generator i (i=1, 2, …, N), calculate the modification value P_i' of P_i using the power outputs of the other $N-1$ generators. Create the set R=$\{1, 2, \cdots, N\} - \{i\}$ to store the indexes of the other $N-1$ generators excluding the ith generator. For the *ED* problem neglecting transmission losses, Eq 5 is modified to Eq 20 to calculate P_i'. For the ED problem considering transmission losses, Eq 6 can be expressed as a second-order equation in P_i' as shown in Eq 21. One root of the second-order equation is chosen as P_i', as shown in Eq 22. Then, modify P_i' using Eq 17 to satisfy the generation capacity limits.

$$P_i' = PD - \sum_{r \in R} P_r \quad (20)$$

$$B_{ii}(P_i')^2 + \left(2\sum_{r\in R} P_r B_{ir} + B_{0i} - 1\right) P_i' + \left(PD + \sum_{r\in R}\sum_{t\in R} P_r B_{rt} P_t + \sum_{r\in R} B_{0r} P_r - \sum_{r\in R} P_r + B_{00}\right) = 0 \quad (21)$$

$$P_i' = \frac{-(2\sum_{r\in R} P_r B_{ir} + B_{0i} - 1) - \sqrt{(2\sum_{r\in R} P_r B_{ir} + B_{0i} - 1)^2 - 4B_{ii}(PD + \sum_{r\in R}\sum_{t\in R} P_r B_{rt} P_t + \sum_{r\in R} B_{0r} P_r - \sum_{r\in R} P_r + B_{00})}}{2B_{ii}} \quad (22)$$

For each generator i (i=1, 2, …, N), assume that the ith generator is selected as the revised generator, and P_i is replaced by P_i'. Calculate the cost change (CX_i) using Eq 23 and the constraint violation (PV_i) of the power demand balance using Eq 18 or 19. Then, calculate the percentage (PCV_i) according to CX_i and PV_i, as shown in Eq 24. Finally, create a set (S) to store the modification sequence of N variables, which is determined by the value of PCV_i sorted in ascending order.

$$CX_i = C_i(P_i') - C_i(P_i) \quad (23)$$

$$PCV_i = \frac{CX_i - min(CX)}{max(CX) - min(CX)} + \frac{PV_i - min(PV)}{max(PV) - min(PV)} \quad (24)$$

Step 4: Modify the power output of each generator in turn according to the modification sequence stored in S until the power demand balance constraint is satisfied. When the ith ($i \in S$) generator is selected as the modified generator, the modified value (P_i') of its power output is calculated by using Eq 20 or 22. If all the generators are modified and the power demand balance constraint is still not satisfied, go to Step 3. Otherwise, go to Step 5.

Step 5: Output the modified weed (W_j).

5. Experimental results and analysis

To validate the optimization ability of HIWO on ED problems with various practical constraints, six classical ED problems in the small, medium, large and very large-scale power systems were selected as the studied test cases. For each test case, the optimal dispatch results obtained by HIWO in 50 independent runs, including the minimum cost (SC_{min}), average cost (SC_{avg}), maximum cost (SC_{max}) and standard deviation of the costs (SC_{std}), are compared to those of algorithms reported in the literature. The best optimization performance among these algorithms is shown in boldface. The parameters of HIWO on six test systems are set as follows: the initial population size Nw_o = 30, maximum population size Nw_{max} = 50, minimum number of seeds Ns_{min} = 1, maximum number of seeds Ns_{max} = 5, nonlinear modulation index m = 5, initial standard deviation $\sigma_{iv} = 2$, final standard deviation $\sigma_{fv} = 0.0001$, maximum number of iterations $Iter_{max}$ = 2000, and number of mutation points X = 1 in the 160-generator system and X = 3 in the other systems. HIWO is implemented by using the *MATLAB* (R2016a) environment on an Intel core *i*7-4790 *CPU* with 8.00 GB *RAM* personal computer.

5.1. Small-scale test system

The 15-generator power system [2,24] considering transmission losses, ramp rate limits and prohibited operating zones is selected as the small-scale test system. The power load demand of the system is 2630 MW. In this test system study, the optimal power outputs of generators obtained by HIWO are shown in Table 1.The optimal dispatch results of HIWO are compared to those of OLCSO [2], WCA [49], ICS [50], FA [27], RTO [51], EMA [52] and IWO, as shown in Tables 2. Compared to other algorithms in terms of minimum, average, maximum and standard deviation of costs in 50 runs, the dispatch solution obtained by HIWO consumes the least cost.

Table 1. Optimal power output of HIWO for the 15-generator system.

Generators	P_i	Generators	P_i	Generators	P_i	Generators	P_i	Generators	P_i
1	455.0000	4	130.0000	7	430.0000	10	159.7871	13	25.0000
2	380.0000	5	170.0000	8	71.2594	11	80.0000	14	15.0000
3	130.0000	6	460.0000	9	58.4944	12	80.0000	15	15.0000

Table 2. Comparison of the optimal dispatch results for the 15-generator system.

Algorithms	SC_{min} ($)	SC_{avg} ($)	SC_{max} ($)	SC_{std}
EMA [52]	32704.4503	32704.4504	32704.4506	NA
FA [27]	32704.5000	32856.1000	33175.0000	147.17022
ICS [50]	32706.7358	32714.4669	32752.5183	NA
WCA [49]	32704.4492	32704.5096	32704.5196	4.513e-05
RTO [51]	32701.8145	32704.5300	32715.1800	5.07
OLCSO [2]	32692.3961	32692.3981	32692.4033	0.0022
IWO	32691.8615	32691.9392	32692.1421	0.0927
HIWO	**32691.5614**	**32691.8615**	**32691.8616**	**0.0001**

5.2. Medium-scale test system

The 40-generator power system [32] considering valve-point effects and transmission losses is selected as the medium-scale test system. The power load demand of the system is 10500 MW. The optimal power outputs obtained by HIWO are shown in Table 3. The optimal dispatch results of HIWO are compared to those of ORCCRO [28], BBO [28], DE/BBO [28], SDE [29], OIWO [32], HAAA [37] and IWO, as shown in Tables 4. Compared to other algorithms in the literature, the proposed HIWO algorithm can obtain the cheapest dispatch solution in terms of minimum, average and maximum of costs in 50 runs.

Table 3. Optimal power output of HIWO for the 40-generator system.

Generators	P_i	Generators	P_i	Generators	P_i	Generators	P_i	Generators	P_i
1	113.9993	9	289.4281	17	489.2798	25	523.2794	33	190.0000
2	113.9993	10	279.5996	18	489.2793	26	523.2794	34	200.0000
3	120.0000	11	243.5995	19	511.2795	27	10.0000	35	199.9999
4	179.7330	12	94.0000	20	511.2793	28	10.0000	36	164.7999
5	87.7999	13	484.0391	21	523.2794	29	10.0000	37	109.9998
6	139.9998	14	484.0390	22	523.2794	30	87.7999	38	110.0000
7	300.0000	15	484.0393	23	523.2794	31	190.0000	39	109.9999
8	299.9997	16	484.0391	24	523.2794	32	190.0000	40	549.9999

Table 4. Comparison of the optimal results for the 40-generator system.

Algorithms	SC_{min} ($)	SC_{avg} ($)	SC_{max} ($)	SC_{std}
ORCCRO [28]	136855.19	136855.19	136855.19	NA
BBO [28]	137026.82	137116.58	137587.82	NA
DE/BBO [28]	136950.77	136966.77	137150.77	NA
SDE [29]	138157.46	NA	NA	NA
OIWO [32]	136452.68	136452.68	136452.68	NA
HAAA [37]	136433.5	136436.6	NA	3.341896
IWO	136543.8580	137009.5641	137679.1073	292.9686
HIWO	**136430.9504**	**136435.2127**	**136441.1059**	4.3238

5.3. Large-scale test system

To verify the dispatch performance of HIWO on large-scale power systems with multiple local optimal solutions, two cases studies are performed to compare the optimization results of HIWO and other algorithms. The detail information of these two cases is shown as follows.

Case I: The 80-generator power system [37] considering valve-point effects. The power load demand is 21000 MW.

Case II: The 110-generator power system [20,32] neglecting valve-point effects and transmission losses. The power load demand is 15000 MW.

Table 5. Optimal power output of HIWO for the 80-generator system (Case I).

Generators	P_i	Generators	P_i	Generators	P_i	Generators	P_i	Generators	P_i
1	110.8335	17	489.3362	33	189.9994	49	284.6071	65	523.2794
2	111.5439	18	489.2794	34	165.1983	50	130.0000	66	523.2835
3	97.3834	19	511.2731	35	199.9997	51	94.0040	67	10.0000
4	179.7603	20	511.2666	36	199.9998	52	94.0000	68	10.0000
5	87.9806	21	523.2525	37	109.9999	53	214.7298	69	10.0000
6	139.9997	22	523.2805	38	110.0000	54	394.2675	70	87.8052
7	259.5584	23	523.2794	39	109.9987	55	394.2967	71	190.0000
8	284.7677	24	523.2794	40	511.2603	56	304.4839	72	189.9997
9	284.6331	25	523.2794	41	110.9296	57	489.3082	73	189.9991
10	130.0000	26	523.2958	42	110.8195	58	489.2773	74	164.7786
11	169.0220	27	10.0000	43	97.3706	59	511.2121	75	199.9994
12	94.0000	28	10.0000	44	179.7187	60	511.2992	76	200.0000
13	214.7422	29	10.0000	45	87.8560	61	523.2830	77	109.9990
14	394.1929	30	89.6856	46	139.9995	62	523.3201	78	110.0000
15	394.2794	31	189.9993	47	259.6320	63	523.2794	79	109.9996
16	394.3050	32	189.9992	48	284.6702	64	523.2794	80	511.2482

In the case I study, the optimal dispatch solution obtained by HIWO is shown in Table 5. The comparison results of generation costs generated by HIWO, THS [34], CSO [40], HAAA [37], GWO [35] and IWO are summarized in Table 6. It can be found from Table 6 that HIWO can obtain the cheapest dispatch solution compared to other algorithms.

Table 6. Comparison of the optimal results for the 80-generator system (Case I).

Algorithms	SC_{min} ($)	SC_{avg} ($)	SC_{max} ($)	SC_{std}
THS [34]	243192.6899	243457.36	NA	120.9889
CSO [40]	243195.3781	243546.6283	244038.7352	NA
HAAA [37]	242815.9	242883	242944.5	29.2849
GWO [35]	242825.4799	242829.8192	242837.1303	0.093
IWO	246386.4038	248088.2077	249888.0623	844.0919
HIWO	**242815.2096**	242836.1110	242872.4662	10.3458

In the case II study, the optimal dispatch solution obtained by HIWO is shown in Table 7. The generation cost generated by HIWO are compared to those of ORCCRO [28], BBO [28], DE/BBO [28], OIWO [32], OLCSO [2] and IWO, which are summarized in Table 8. Compared to other algorithms in terms of minimum, average, maximum and standard deviation of costs in 50 runs, the optimal dispatch solution obtained by HIWO generates the least generation cost.

Table 7. Optimal power output of HIWO for the 110-generator system (Case II).

Generators	P_i	Generators	P_i	Generators	P_i	Generators	P_i	Generators	P_i
1	2.4000	23	68.9000	45	659.9999	67	70.0000	89	82.4977
2	2.4000	24	350.0000	46	616.2499	68	70.0000	90	89.2333
3	2.4000	25	400.0000	47	5.4000	69	70.0000	91	57.5687
4	2.4000	26	400.0000	48	5.4000	70	359.9999	92	99.9986
5	2.4000	27	499.9992	49	8.4000	71	399.9999	93	439.9998
6	4.0000	28	500.0000	50	8.4000	72	399.9998	94	499.9999
7	4.0000	29	199.9997	51	8.4000	73	105.2864	95	600.0000
8	4.0000	30	99.9998	52	12.0000	74	191.4091	96	471.5717
9	4.0000	31	10.0000	53	12.0000	75	89.9996	97	3.6000
10	64.5432	32	19.9993	54	12.0000	76	49.9999	98	3.6000
11	62.2465	33	79.9950	55	12.0000	77	160.0000	99	4.4000
12	36.2739	34	249.9998	56	25.2000	78	295.4962	100	4.4000
13	56.6406	35	359.9999	57	25.2000	79	175.0102	101	10.0000
14	25.0000	36	399.9997	58	35.0000	80	98.2829	102	10.0000
15	25.0000	37	39.9998	59	35.0000	81	10.0000	103	20.0000
16	25.0000	38	69.9996	60	45.0000	82	12.0000	104	20.0000
17	154.9999	39	99.9998	61	45.0000	83	20.0000	105	40.0000
18	154.9993	40	119.9984	62	45.0000	84	199.9999	106	40.0000
19	155.0000	41	157.4299	63	184.9996	85	324.9972	107	50.0000
20	155.0000	42	219.9999	64	184.9996	86	440.0000	108	30.0000
21	68.9000	43	439.9999	65	184.9984	87	14.0886	109	40.0000
22	68.9000	44	559.9998	66	184.9997	88	24.0910	110	20.0000

Table 8. Comparison of the optimal results for the 110-generator system (Case II).

Algorithms	SC_{min} ($)	SC_{avg} ($)	SC_{max} ($)	SC_{std}
ORCCRO [28]	198016.29	198016.32	198016.89	NA
BBO [28]	198241.166	198413.45	199102.59	NA
DE/BBO [28]	198231.06	198326.66	198828.57	NA
OIWO [32]	197989.14	197989.41	197989.93	NA
OLCSO [2]	197988.8576	197989.5832	197990.4551	0.3699
IWO	198252.3594	198621.3233	198902.7697	138.4714
HIWO	**197988.1927**	**197988.1969**	**197988.2045**	**0.0025**

5.4. Very large-scale test system

To investigate the dispatch performance of HIWO on very large-scale power systems, the following two cases studies are performed for comparing the optimization results of HIWO and other algorithms.

Case I: The 140-generator Korea power system [23,32] neglecting transmission losses. The 12 generators consider the valve point effects. The power load demand is 49342 MW.

Case II: The 160-generator power system [32] considering valve-point effects. The power load demand is 43200 MW.

Table 9. Optimal power output of HIWO for the 140-generator system (Case I).

Generators	P_i	Generators	P_i	Generators	P_i	Generators	P_i	Generators	P_i
1	115.2442	29	500.9998	57	103.0000	85	115.0000	113	94.0000
2	189.0000	30	500.9994	58	198.0000	86	207.0000	114	94.0000
3	190.0000	31	505.9993	59	311.9941	87	207.0000	115	244.0000
4	190.0000	32	505.9997	60	281.1604	88	175.0000	116	244.0000
5	168.5393	33	506.0000	61	163.0000	89	175.0000	117	244.0000
6	189.9932	34	505.9998	62	95.0000	90	175.0000	118	95.0000
7	489.9992	35	499.9996	63	160.0000	91	175.0000	119	95.0000
8	489.9996	36	500.0000	64	160.0000	92	579.9998	120	116.0000
9	495.9997	37	240.9993	65	489.9465	93	645.0000	121	175.0000
10	495.9994	38	240.9999	66	196.0000	94	983.9998	122	2.0000
11	495.9997	39	773.9996	67	489.9717	95	977.9993	123	4.0000
12	496.0000	40	769.0000	68	489.9908	96	681.9997	124	15.0000
13	506.0000	41	3.0000	69	130.0000	97	719.9998	125	9.0000
14	509.0000	42	3.0000	70	234.7202	98	717.9993	126	12.0000
15	506.0000	43	249.2474	71	137.0000	99	719.9997	127	10.0000
16	504.9997	44	246.0287	72	325.4950	100	963.9998	128	112.0000
17	505.9997	45	249.9973	73	195.0000	101	958.0000	129	4.0000
18	505.9997	46	249.9863	74	175.0000	102	1006.9992	130	5.0000
19	504.9994	47	241.0622	75	175.0000	103	1006.0000	131	5.0000
20	505.0000	48	249.9950	76	175.0000	104	1012.9999	132	50.0000
21	504.9998	49	249.9916	77	175.0000	105	1019.9996	133	5.0000
22	505.0000	50	249.9995	78	330.0000	106	953.9999	134	42.0000
23	504.9998	51	165.0000	79	531.0000	107	951.9998	135	42.0000
24	504.9996	52	165.0000	80	530.9995	108	1005.9996	136	41.0000
25	536.9997	53	165.0000	81	398.6524	109	1013.0000	137	17.0000
26	536.9995	54	165.0000	82	56.0000	110	1020.9998	138	7.0000
27	548.9998	55	180.0000	83	115.0000	111	1014.9996	139	7.0000
28	548.9993	56	180.0000	84	115.0000	112	94.0000	140	26.0000

In the case I study, the optimal dispatch solution obtained by HIWO is shown in Table 9. The optimal results of HIWO are compared to those of SDE [29], OIWO [32], HAAA [37], GWO [35], KGMO [41] and IWO, as shown in Table 10. The corrected optimal result of OIWO is shown in italics. Compared to other algorithms in terms of minimum, average, maximum and standard deviation of costs in 50 runs, HIWO can obtain the cheapest dispatch solution.

Table 10. Comparison of the optimal results of HIWO for the 140-generator system (Case I).

Algorithms	SC_{min} (\$)	SC_{avg} (\$)	SC_{max} (\$)	SC_{std}
SDE [29]	1560236.85	NA	NA	NA
OIWO [32]	*1559712.2604*	NA	NA	NA
HAAA [37]	1559710.00	1559712.87	1559731.00	4.1371
GWO [35]	1559953.18	1560132.93	1560228.40	1.024
KGMO [41]	1583944.60	1583952.14	1583963.52	NA
IWO	1564050.0027	1567185.2227	1571056.6280	1678.8488
HIWO	**1559709.5266**	**1559709.6956**	**1559709.8959**	**0.0856**

In the case II study, the optimal dispatch solution obtained by HIWO is shown in Table 11. The optimal results of HIWO are compared to those of ORCCRO [28], BBO [28], DE/BBO [28], CBA [31], OIWO [32] and IWO, as shown in Table 12. Compared to other algorithms, HIWO can also obtain the cheapest dispatch solution in terms of minimum, average, maximum and standard deviation of costs.

Table 11. Optimal power output of HIWO for the 160-generator system (Case II).

Generators	P_i	Generators	P_i	Generators	P_i	Generators	P_i	Generators	P_i
1	218.6095	33	280.6560	65	279.6118	97	287.7203	129	431.0758
2	209.2361	34	238.9676	66	238.5645	98	238.6988	130	275.8790
3	279.6486	35	279.9554	67	287.7296	99	426.2750	131	219.6189
4	240.3113	36	240.9831	68	241.2519	100	272.6741	132	210.4739
5	280.0206	37	290.1069	69	427.7708	101	217.5647	133	281.6640
6	238.4301	38	240.0425	70	272.9907	102	211.9593	134	238.9676
7	288.2326	39	426.3102	71	218.5918	103	280.6578	135	276.5752
8	239.5051	40	275.6392	72	212.7020	104	239.2363	136	239.3707
9	425.6549	41	219.6195	73	281.6629	105	276.3263	137	287.7806
10	275.6903	42	210.9690	74	238.9676	106	240.7144	138	238.5645
11	217.5646	43	282.6711	75	279.3688	107	290.0715	139	430.7874
12	212.4544	44	240.3113	76	237.6239	108	238.8332	140	275.8606
13	280.6558	45	279.7868	77	289.9995	109	425.7918	141	218.6539
14	238.6988	46	237.4895	78	239.9082	110	275.2705	142	210.7215
15	279.9370	47	287.7274	79	425.2406	111	217.5671	143	281.6640
16	240.7144	48	240.0425	80	276.0112	112	212.2069	144	239.3707
17	287.6968	49	427.4497	81	218.5923	113	281.6664	145	276.3578
18	239.7738	50	275.6817	82	212.2069	114	239.6394	146	239.6394
19	427.4049	51	219.6197	83	282.7049	115	276.0940	147	287.7565
20	275.6990	52	213.4447	84	237.7582	116	240.3113	148	239.3707
21	217.5665	53	282.6717	85	279.7940	117	290.0972	149	426.3023

Continued on next page

Generators	P_i	Generators	P_i	Generators	P_i	Generators	P_i	Generators	P_i
22	212.2069	54	237.8926	86	239.3707	118	239.5051	150	275.6371
23	283.6805	55	276.2856	87	290.0916	119	429.4367	151	217.5647
24	239.7738	56	239.5051	88	239.2363	120	275.6690	152	212.2069
25	279.9011	57	287.6883	89	427.0504	121	217.5656	153	279.6493
26	240.9831	58	238.5645	90	275.7937	122	210.2264	154	238.4301
27	290.0737	59	429.9489	91	217.5643	123	280.6617	155	279.9078
28	240.8488	60	275.5096	92	212.9496	124	239.7738	156	240.4457
29	427.1007	61	218.5915	93	282.6732	125	275.9409	157	287.7385
30	276.2995	62	212.9496	94	240.4457	126	240.1769	158	238.5645
31	219.6189	63	282.6705	95	279.4854	127	287.6965	159	426.9110
32	211.7117	64	239.9082	96	240.1769	128	238.4301	160	272.7775

Table 12. Comparison of the optimal results for the 160-generator system (Case II).

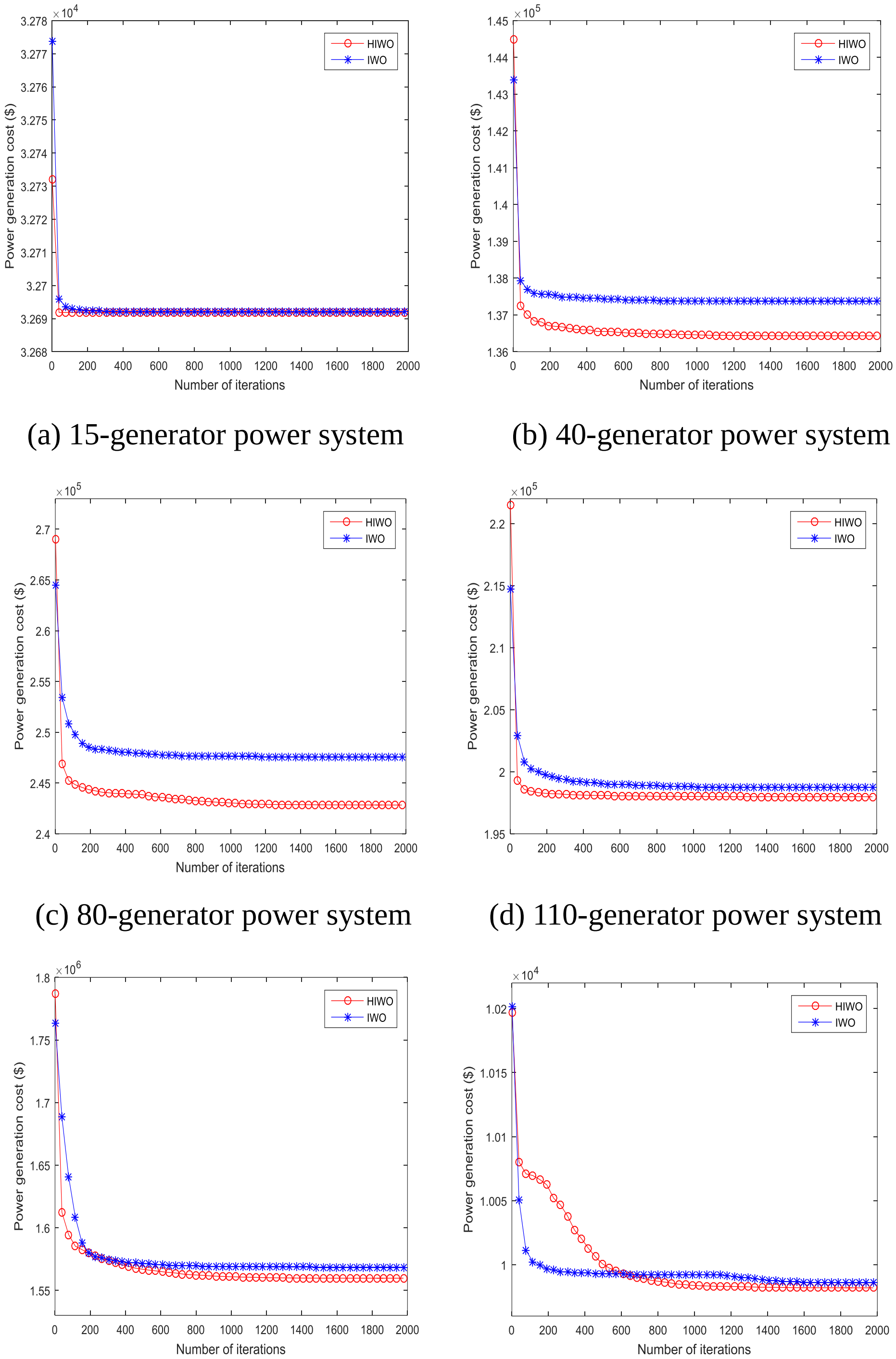
Algorithms	SC_{min} ($)	SC_{avg} ($)	SC_{max} ($)	SC_{std}
ORCCRO [28]	10004.20	10004.21	10004.45	NA
OIWO [32]	9981.9834	9982.991	9983.998	NA
BBO [28]	10008.71	10009.16	10010.59	NA
DE/BBO [28]	10007.05	10007.56	10010.26	NA
CBA [31]	10002.8596	10006.3251	10045.2265	9.5811
IWO	9984.8409	9985.5127	9986.1947	0.3252
HIWO	**9981.7867**	**9982.0010**	**9982.1922**	**0.0934**

5.5. Convergence tests

To illustrate the convergence ability of HIWO for solving different-scale ED problems with various constraints, the convergence curves of HIWO and IWO on six test systems are drawn, as shown in Figure 2. It can be found from Figure 2 that HIWO can converge to the optimal areas in the six test systems, and the convergence speed of HIWO on the 15, 40, 80,110 and 140-generator power systems, is faster than that of IWO. Although the convergence speed of HIWO on the 160-generator power system is slower than that of IWO in the early evolutionary stage, it is faster than that of IWO in the later evolutionary stage. The reason is that the crossover and mutation decrease the fitness value of offspring weeds in 160-generator power system having lots of constraints, and then reduce the convergence speed in the early evolutionary stage, but increase the diversity of the population to jump out local optimization in the later stage.

6. Conclusion

In this paper, a hybrid HIWO algorithm combining IWO with GA is proposed to solve ED problems in power systems. The HIWO adopts IWO to explore the various regions in the solution space, while the crossover and mutation operations of GA are applied to improve the quality and diversity of solutions, thereby preventing the optimization from prematurity and enhancing the search capability. Moreover, an effective repair method is proposed to repair infeasible solutions to feasible solutions. The experimental results of the six test systems studies show that HIWO can obtain the cheapest dispatch solutions compared to other algorithms in the literature, and have a better optimization ability and faster convergence speed compared to the classical IWO. In summary, the proposed HIWO algorithm is an effective and promising approach for solving ED problems in different-scale power systems.



(a) 15-generator power system (b) 40-generator power system

(c) 80-generator power system (d) 110-generator power system

(e) 140-generator power system (f) 160-generator power system

Figure 2. Convergence curves of HIWO and IWO on the six test systems.

Acknowledgments

This research is partially supported by the National Science Foundation of China under grant numbers 61773192 and 61773246, the Key Laboratory of Computer Network and Information Integration (Southeast University), the Ministry of Education (K93-9-2017-02), and the State Key Laboratory of Synthetical Automation for Process Industries (PAL-N201602).

Conflicts of interest

The authors declare no conflict of interest.

References

1. Z. X. Liang and J. D. Glover, A zoom feature for a dynamic programming solution to economic dispatch including transmission losses, *IEEE T. Power Syst.*, **7** (1992), 544–550.
2. G. Xiong and D. Shi, Orthogonal learning competitive swarm optimizer for economic dispatch problems, *Appl. Soft. Comput. J.*, **66** (2018), 134–148.
3. R. A. Jabr, A. H. Coonick and B. J. Cory, A homogeneous linear programming algorithm for the security constrained economic dispatch problem, *IEEE T. Power Syst.*, **15** (2000), 930–936.
4. S. Muralidharan, K. Srikrishna and S. Subramanian, Self-adaptive dynamic programming technique for economic power dispatch, *Int. J. Power Energy Syst.*, **27** (2007), 340–345.
5. N. Sinha, R. Chakrabarti and P. K. Chattopadhyay, evolutionary programming techniques for economic load dispatch, *IEEE T. Evol. Comput.*, **7** (2003), 83–94.
6. J. Q. Li, H. Y. Sang, Y. Y. Han, et al., Efficient multi-objective optimization algorithm for hybrid flow shop scheduling problems with setup energy consumptions, *J. Clean. Prod.*, **181** (2018), 584–598.
7. J. Q. Li, S. C. Bai, P. Y. Duan, et al., An improved artificial bee colony algorithm for addressing distributed flow shop with distance coefficient in a prefabricated system, *Int. J. Prod. Res.*, (2019).
8. J. Q. Li, Q. K. Pan and K. Mao, A discrete teaching-learning-based optimisation algorithm for realistic flowshop rescheduling problems, *Eng. Appl. Artif. Intell.*, **37** (2015), 279–292.
9. J. Q. Li, P. Y. Duan, H. Y. Sang, et al., An efficient optimization algorithm for resource-constrained steelmaking scheduling problems, *IEEE Access*, **6** (2018), 33883–33894.
10. J. Q. Li, P. Y. Duan, J. D. Cao, et al., A hybrid Pareto-based tabu search for the distributed flexible job shop scheduling problem with E/T criteria, *IEEE Access*, **6** (2018), 58883–58897.
11. J. Q. Li, Q. K. Pan and M. F. Tasgetiren, A discrete artificial bee colony algorithm for the multi-objective flexible job-shop scheduling problem with maintenance activities, *Appl. Math. Model.*, **38** (2014), 1111–1132.
12. J. Q. Li, Q. K. Pan and S. X. Xie, An effective shuffled frog-leaping algorithm for multi-objective flexible job shop scheduling problems, *Appl. Math. Comput.*, **218** (2012), 9353–9371.
13. J. Q. Li, Q. K. Pan and K. Z. Gao, Pareto-based discrete artificial bee colony algorithm for multi-objective flexible job shop scheduling problems, *Int. J. Adv. Manuf. Technol.*, **55** (2011), 1159–1169.

14. J. Q. Li, A hybrid multi-objective artificial bee colony algorithm for flexible task scheduling problems in Cloud computing system, *Cluster Comput.*, (2019), 1–24.
15. P. Y. Duan, J. Q. Li, Y. Wang, et al., Solving chiller loading optimization problems using an improved teaching-learning-based optimization algorithm, Optim, *Control Appl. Met.*, **39** (2018), 65–77.
16. Z. X. Zheng, J. Q. Li and P.Y. Duan, Optimal chiller loading by improved artificial fish swarm algorithm for energy saving, *Math. Comput. Simul.*, **155** (2019), 227–243.
17. Z. X. Zheng and J. Q. Li, Optimal chiller loading by improved invasive weed optimization algorithm for reducing energy consumption, *Energy Build*, **161**(2018), 80–88.
18. S. Kumar and R. Naresh, Nonconvex economic load dispatch using an efficient real-coded genetic algorithm, *Appl. Soft Comput. J.*, **9** (2009), 321–329.
19. P. Subbaraj, R. Rengaraj and S. Salivahanan, Enhancement of Self-adaptive real-coded genetic algorithm using Taguchi method for Economic dispatch problem, *Appl. Soft Comput. J.*, **11** (2011), 83–92.
20. S. O. Orero and M. R. Irving, Large scale unit commitment using a hybrid genetic algorithm, *Int. J. Electr. Power Energy Syst.*, **19** (1997), 45–55.
21. P. Chen and H. Chang, Large-scale economic dispatch by genetic algorithm, *IEEE T. Power Syst.*, **10** (1995), 1919–1926.
22. A. I. Selvakumar, A new particle swarm optimization solution to nonconvex economic dispatch problems, *IEEE T. Power Syst.*, **22** (2007), 42–51.
23. J. B. Park, Y. W. Jeong, J. R. Shin, et al., An improved particle swarm optimization for nonconvex economic dispatch problems, *IEEE T. Power Syst.*, **25** (2010), 156–166.
24. Z. Gaing, Particle swarm optimization to solving the economic dispatch considering the generator constraints, *IEEE T. Power Syst.*, **18** (2003), 1187–1195.
25. K. T. Chaturvedi, M. Pandit and L. Srivastava, Self-organizing hierarchical particle swarm optimization for nonconvex economic dispatch, *IEEE T. Power Syst.*, **23** (2008), 1079–1087.
26. Q. Qin, S. Cheng, X. Chu, et al., Solving non-convex/non-smooth economic load dispatch problems via an enhanced particle swarm optimization, *Appl. Soft Comput. J.*, **59** (2017), 229–242.
27. X. S. Yang, S. S. S. Hosseini and A. H. Gandomi, Firefly Algorithm for solving non-convex economic dispatch problems with valve loading effect, *Appl. Soft Comput. J.*, **12** (2012), 1180–1186.
28. K. Bhattacharjee, A. Bhattacharya and S. H. N. Dey, Oppositional real coded chemical reaction optimization for different economic dispatch problems, *Int. J. Electr. Power Energy Syst.*, **55** (2014), 378–391.
29. A. S. Reddy and K. Vaisakh, Shuffled differential evolution for large scale economic dispatch, *Electr. Power Syst. Res.*, **96** (2013), 237–245.
30. D. Zou, S. Li, G. G. Wang, et al., An improved differential evolution algorithm for the economic load dispatch problems with or without valve-point effects, *Appl. Energy*, **181** (2016), 375–390.
31. B. R. Adarsh, T. Raghunathan, T. Jayabarathi, et al., Economic dispatch using chaotic bat algorithm, *Energy*, **96** (2016), 666–675.
32. A. K. Barisal and R. C. Prusty, Large scale economic dispatch of power systems using oppositional invasive weed optimization, *Appl. Soft Comput. J.*, **29** (2015), 122–137.

33. S. Banerjee, D. Maity and C. K. Chanda, Teaching learning based optimization for economic load dispatch problem considering valve point loading effect, *Int. J. Electr. Power Energy Syst.*, **73** (2015), 456–464.
34. M. A. Al-Betar, M. A. Awadallah, A. T. Khader, et al., Tournament-based harmony search algorithm for non-convex economic load dispatch problem, *Appl. Soft Comput. J.*, **47** (2016), 449–459.
35. M. Pradhan, P. K. Roy and T. Pal, Grey wolf optimization applied to economic load dispatch problems, *Int. J. Electr. Power Energy Syst.*, **83** (2016), 325–334.
36. T. Jayabarathi, T. Raghunathan, B. R. Adarsh, et al., Economic dispatch using hybrid grey wolf optimizer, *Energy*, **111** (2016), 630–641.
37. M. Kumar and J. S. Dhillon, Hybrid artificial algae algorithm for economic load dispatch, *Appl. Soft Comput.*, **71** (2018), 89–109.
38. M. Modiri-Delshad and N. A. Rahim, Solving non-convex economic dispatch problem via backtracking search algorithm, *Energy*, **77** (2014), 372–381.
39. J. J. Q. Yu and V. O. K. Li, A social spider algorithm for solving the non-convex economic load dispatch problem, *Neurocomputing*, **171** (2016), 955–965.
40. A. I. Selvakumar and K. Thanushkodi, Optimization using civilized swarm: Solution to economic dispatch with multiple minima, *Electr. Power Syst. Res.*, **79** (2009), 8–16.
41. M. Basu, Kinetic gas molecule optimization for nonconvex economic dispatch problem, *Int. J. Electr. Power Energy Syst.*, **80** (2016), 325–332.
42. J. Cai, Q. Li, L. Li, et al., A hybrid FCASO-SQP method for solving the economic dispatch problems with valve-point effects, *Energy*, **38** (2012), 346–353.
43. J. S. Alsumait, J. K. Sykulski and A. K. Al-Othman, A hybrid GA-PS-SQP method to solve power system valve-point economic dispatch problems, *Appl. Energy*, **87** (2010), 1773–1781.
44. J. Cai, Q. Li, L. Li, et al., A hybrid CPSO-SQP method for economic dispatch considering the valve-point effects, *Energy Convers. Manag.*, **53** (2012), 175–181.
45. S. Sayah and A. Hamouda, A hybrid differential evolution algorithm based on particle swarm optimization for nonconvex economic dispatch problems, *Appl. Soft Comput. J.*, **13** (2013), 1608–1619.
46. A. R. Mehrabian and C. Lucas, A novel numerical optimization algorithm inspired from weed colonization, *Ecol. Inform.*, **1** (2006), 355–366.
47. T. A. A. Victoire and A. E. Jeyakumar, Reserve constrained dynamic dispatch of units with valve-point effects, *IEEE T. Power Syst.*, **20** (2005), 1273–1282.
48. T. Niknam and F. Golestaneh, Enhanced adaptive particle swarm optimisation algorithm for dynamic economic dispatch of units considering valve-point effects and ramp rates, *IET Gener. Transm. Distrib.*, **6** (2012), 424–435.
49. M. A. Elhameed and A. A. El-Fergany, Water cycle algorithm-based economic dispatcher for sequential and simultaneous objectives including practical constraints, *Appl. Soft Comput. J.*, **58** (2017), 145–154.
50. E. Afzalan and M. Joorabian, An improved cuckoo search algorithm for power economic load dispatch, *Int. Trans. Electr. Energy Syst.*, **25** (2015), 958–975.
51. Y. Labbi, D. B. Attous, H. A. Gabbar, et al., A new rooted tree optimization algorithm for economic dispatch with valve-point effect, *Int. J. Electr. Power Energy Syst.*, **79** (2016), 298–311.

52. N. Ghorbani and E. Babaei, Exchange market algorithm for economic load dispatch, *Int. J. Electr. Power Energy Syst.*, **75** (2016), 19–27.