3-10-2023

Hybridizing gaining–sharing knowledge and differential evolution for large-scale power system economic dispatch problems

Qinghua Liu

Guizhou Key Laboratory of Intelligent Technology in Power System, College of Electrical Engineering, Guizhou University , Guiyang 550025 , China

Guojiang Xiong

Guizhou Key Laboratory of Intelligent Technology in Power System, College of Electrical Engineering, Guizhou University , Guiyang 550025 , China,Institute of Engineering Investigation & Design Co., Ltd., Guizhou University , Guiyang 550025 , China

Xiaofan Fu

Guizhou Key Laboratory of Intelligent Technology in Power System, College of Electrical Engineering, Guizhou University , Guiyang 550025 , China

Ali Wagdy Mohamed

Operations Research Department, Faculty of Graduate Studies for Statistical Research, Cairo University , Giza 12613 , Egypt,Department of Mathematics and Actuarial Science, School of Sciences & Engineering, The American University in Cairo , New Cairo 11835 , Egypt

Journal of Computational Design and Engineering, 2023, **10**, 615–631
DOI: 10.1093/jcde/qwad008
Advance access publication date: 20 March 2023
Research Article

Hybridizing gaining–sharing knowledge and differential evolution for large-scale power system economic dispatch problems

Qinghua Liu[1], Guojiang Xiong[1,2,*], Xiaofan Fu[1], Ali Wagdy Mohamed[3,4], Jing Zhang[1], Mohammed Azmi Al-Betar[5], Hao Chen[6,*], Jun Chen[7] and Sheng Xu[8]

[1]Guizhou Key Laboratory of Intelligent Technology in Power System, College of Electrical Engineering, Guizhou University, Guiyang 550025, China
[2]Institute of Engineering Investigation & Design Co., Ltd., Guizhou University, Guiyang 550025, China
[3]Operations Research Department, Faculty of Graduate Studies for Statistical Research, Cairo University, Giza 12613, Egypt
[4]Department of Mathematics and Actuarial Science, School of Sciences & Engineering, The American University in Cairo, New Cairo 11835, Egypt
[5]Artificial Intelligence Research Center (AIRC), College of Engineering and Information Technology, Ajman University, Ajman 346, UAE
[6]Fujian Provincial Key Laboratory of Intelligent Identification and Control of Complex Dynamic System, Quanzhou 362216, China
[7]Department of Electrical and Computer Engineering, Oakland University, Rochester, MI 48309, USA
[8]Guizhou Electric Power Grid Dispatching and Control Center, Guiyang 550002, China
*Corresponding authors. E-mail: gjxiongee@foxmail.com (GX); chenhao@fjirsm.ac.cn (HC)

Abstract

Economic dispatch (ED) of thermal power units is significant for optimal generation operation efficiency of power systems. It is a typical nonconvex and nonlinear optimization problem with many local extrema when considering the valve-point effects, especially for large-scale systems. Considering that differential evolution (DE) is efficient in locating global optimal region, while gain-sharing knowledge-based algorithm (GSK) is effective in refining local solutions, this study presents a new hybrid method, namely GSK-DE, to integrate the advantages of both algorithms for solving large-scale ED problems. We design a dual-population evolution framework in which the population is randomly divided into two equal subpopulations in each iteration. One subpopulation performs GSK, while the other executes DE. Then, the updated individuals of these two subpopulations are combined to generate a new population. In such a manner, the exploration and the exploitation are harmonized well to improve the searching efficiency. The proposed GSK-DE is applied to six ED cases, including 15, 38, 40, 110, 120, and 330 units. Simulation results demonstrate that GSK-DE gives full play to the superiorities of GSK and DE effectively. It possesses a quicker global convergence rate to obtain higher quality dispatch schemes with greater robustness. Moreover, the effect of population size is also examined.

Keywords: differential evolution, economic dispatch, gaining–sharing knowledge-based algorithm, power system, valve-point effect

List of symbols

C:	Total generation cost ($/h).
N:	Number of generators.
P_n:	Active power of the nth thermal power unit (MW).
$F_n(P_n)$:	Fuel cost function of the nth thermal power unit ($/h).
a_n, b_n, c_n:	Generation cost coefficients.
e_n, f_n:	Valve-point effect coefficients.
P_n^{min}:	Lower limit of active power for the nth unit (MW).
P_n^{max}:	Upper limit of active power for the nth unit (MW).
P_{loss}:	Total transmission network loss (MW).
P_D:	Total system load (MW).
B_{ij}, B_{0i}, B_{0j}:	Loss coefficients.
UR_n:	Increasing limit of the nth thermal power unit (MW).
DR_n:	Decreasing limit of the nth thermal power unit (MW).
P_n^{pr}:	Previous output of the nth thermal power unit (MW).
pz_n:	Number of prohibited operating zones of the nth generator.
$P_{n,k}^L, P_{n,k}^U$:	Lower and upper bounds of the kth prohibited operating zone of the nth generator (MW).
NP:	Number of individuals.
POZ:	Prohibited operating zone.
D:	Dimension.
x_i:	Individual in a D-dimensional space.
G:	Current generation.
GEN:	Maximum generation number.
K:	Knowledge rate.
k_r:	Knowledge ratio.
k_f:	Knowledge factor.
p:	Constant between 0 and 1.
x_{p-best}:	Random individual in the best group.
$x_{p-worst}$:	Random individual in the worst group.
v_i:	Mutant vector.
F:	Scaling factor.
r_1, r_2, r_3:	Mutually different individuals.
CR:	Crossover rate.
j_{rand}:	Random dimension of $\{1, 2, \ldots, D\}$.

1. Introduction

The purpose of economic dispatch (ED) of power systems is to reasonably allocate the active power of grid-connected thermal

Received: September 1, 2022. **Revised:** January 1, 2023. **Accepted:** January 18, 2023

power units in the system with known system load demand, so that the total generation cost of the system is minimized while satisfying the system power balance constraint and thermal power unit operation constraints. The generation fuel cost of the traditional ED problem is usually expressed using a linear function or quadratic polynomial, which is simple in form but often causes inaccuracy in the dispatch solution for the power system. The ED model is more accurate after involving the valve-point effects of units and network losses, but the solution space becomes highly nonlinear, and nonconvex, and has a large number of local extremes, especially for large-scale power systems, which makes the solution very complex. Because of this, scholars have presented many solutions to this issue. These solution approaches can be loosely classified into two categories based on their mathematical perspective, i.e., classical mathematical methods and intelligent optimization methods.

The classical mathematical methods include linear programming (Jabr *et al.*, 2000), nonlinear programming (Nanda *et al.*, 1994), dynamic programming (Muralidharan *et al.*, 2007), mixed integer programming (Azzam *et al.*, 2014), Lagrange relaxation (Hindi & Ghani, 1991), and so on. The main advantages of these methods are the maturity of the theory and the convergence speed of the solution. However, these methods face a lot of difficulties in solving the ED problem. For example, the linear programming requires a certain initial value and the objective function being convex, and its generality is not strong enough. The nonlinear programming requires a continuous and differentiable objective function, and results in large computational effort and poor stability for high-dimensional problems. The dynamic programming and the mixed integer programming are prone to "dimensional disaster" as the number of dimensions increases. The Lagrange relaxation method easily leads to oscillation.

To overcome the troubles associated with the classical mathematical methods, the intelligent optimization methods have been frequently applied as alternative solutions to the nonconvex ED problems. Some of these methods have been already implemented in their original form (Chiang, 2007). At the same time, in view of the shortcomings in the basic algorithms mentioned earlier, many scholars have made improvements to enhance their optimization performance in solving the ED problems. Among them, hybridization is a frequent and effective way for improvement because it can integrate the merits of different algorithms. For example, Victoire and Jeyakumar (2006) hybridized tabu search (TS), particle swarm optimization (PSO), and sequential quadratic programming (SQP) technique to deal with the fuzzy modeling unit input problem. Khamsawang and Jiriwibhakorn (2010) hybridized the distributed Sobol PSO and TS algorithm. The method starts with a Sobol sequence instead of the existing process to generate the inertia factors, and then, uses distributed programs and activates the TS strategy to quickly obtain the global optimum. Parouha and Das (2016) proposed a differential evolution (DE)-PSO-DE method. It divides the population into three groups, i.e., lower, intermediate, and upper groups. The lower and upper groups use DE, while the intermediate group uses PSO. The difference information from DE is combined with the memory information extracted by PSO in order to get rid of the local extreme points. Chansareewittaya (2017) proposed a hybrid bee algorithm (BA)/TS method. It uses the local search method in the BA to replace the TS in order to arrive at the global minimum solution. Chen and Marrani (2020) devised a hybrid solution approach that combines the PSO with imperialist competitive algorithm (ICA). In this method, the search strategy of PSO is applied to ICA to improve the exploitation capability. Zhang *et al.* (2013) put forward a hybrid algorithm based on PSO and DE. It integrates the variational operator and crossover operator of DE, and the chaotic sequence into PSO to improve the global search capability. Takeang and Aurasopon (2019) used the Lambda iteration to initialize a refined population instead of the traditional random values for simulated annealing method to prevent the rapid loss of population diversity.

Many scholars have accomplished a lot of works to achieve good results, but they were mainly applied to relatively small-scale systems. As the requirement for the power energy rises and the energy Internet progresses, the power system is growing in scale and sophistication significantly. In this context, the quantity of units in a power system grows quickly, making the ED problem solution space more complicated, especially when considering the valve-point effects. As a consequence, the difficulty for achieving an accurate solution for the ED problem increases considerably and it is urgent and essential to design an effective ED solution approach to adapt to large-scale power systems.

In this research, we attempt to design an effective approach to resolve the ED problem of large-scale power systems by hybridizing gain-sharing knowledge-based algorithm (GSK) and DE.

DE is one of the most extensively popular population-based algorithms and has the advantages of simple structure, easy to use, and robustness (Wang and Tan, 2020; Gao *et al.*, 2020). It has shown qualified performance in handling different optimization problems. Nevertheless, the fundamental DE algorithm has a high pressure on the selection of suitable control parameters, which often drags down the global convergence and results in poor local refinement search ability (Wang *et al.*, 2022). Therefore, it makes the performance of the algorithm deteriorate in the process of evolution.

GSK, a new metaheuristic algorithm proposed for combinatorial optimization, was inspired by people behavior of acquiring and sharing knowledge (Mohamed *et al.*, 2020). It has received a lot of attention for its ease of implementation and efficiency, and has worked well in several fields, including fault diagnosis of power systems (Li *et al.*, 2022; Xiong *et al.*, 2022b, c), knapsack problems (Agrawal *et al.*, 2022), travelling advisor problem (Hassan *et al.*, 2020), and parameter extraction of solar photovoltaic models (Xiong *et al.*, 2021). Nevertheless, like a coin has two sides, GSK also has some drawbacks. On the one hand, GSK is good at using local search capability to refine the solution thoroughly. On the other hand, when solving complex multimodal problems, it is easy to be bounded by the local extrema and converges prematurely because of insufficient global search capability.

From the above analysis, we found that the features of GSK and DE are highly complementary. Namely, GSK is good at exploiting the current region to refine the solution, while DE is adept in exploring the region where the global solution locates. These two search features are extremely indispensable for an algorithm. This motivates us that a logical hybridization of them may result in an effective algorithm with preeminent performance. Moreover, to our best knowledge, GSK and DE have not been hybridized before to solve the ED problems. In this work, a hybrid GSK-DE algorithm based on the idea of dual population is devised to solve large-scale ED problems. In view of increasing the efficiency of searching the solution space, a dual-population evolution framework that updates individuals making use of two different search strategies is presented. Specifically, after randomly dividing all individuals into two different subpopulations equally, one half executes GSK while the other half executes DE. In this way, the population uses a heterogeneous search strategy to increase the population diversity and speed up the convergence for the purpose of improving the quality of solutions found by the designed algo-

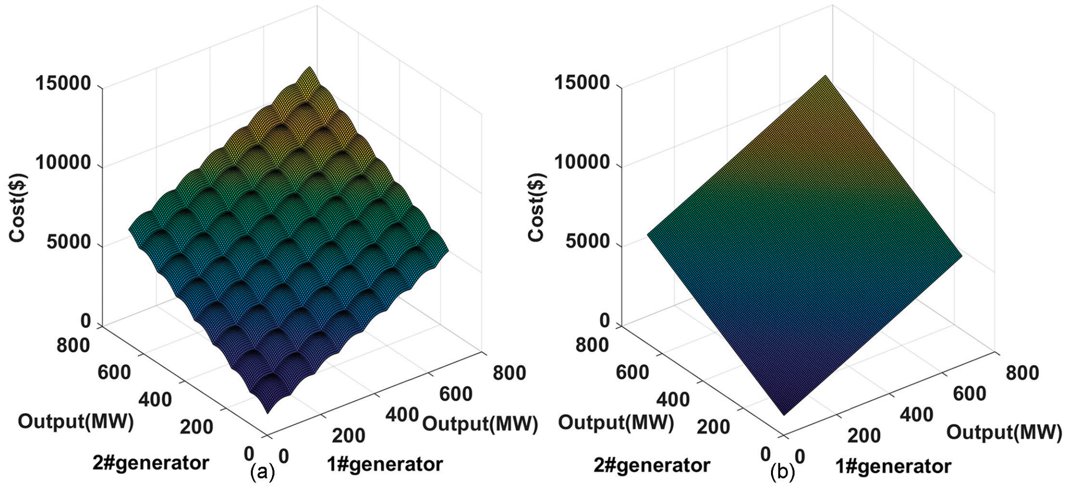

Figure 1: (a) Fuel cost function with the valve-point effect and (b) fuel cost function without the valve-point effect.

rithm. To evaluate the effectiveness of GSK-DE in solving large-scale ED problems, we focus on demonstrating and discussing its application in six test cases, including 15-unit, 38-unit, 40-unit, 110-unit, 120-unit, and 330-unit systems. A comparative analysis with other algorithms shows the superiority of the designed algorithm from convergence performance, solution quality, and robustness. The effect of population size is also discussed in depth.

The rest of this paper is laid out as follows: The mathematical model of the ED problem is described in Section 2. The proposed GSK-DE is presented in Section 3. The simulation results and discussions are presented in Section 4. The conclusions and future work are summarized in Section 5.

2. Mathematical Formulation of ED

2.1. Objective function

The objective of the ED problem is to reduce the total generation cost over the dispatch cycle by appropriately allocating the output of each thermal power unit, while meeting the requirements of load demand and operating constraints. The optimization objective of the total generation cost is expressed as (Sinha *et al.*, 2003)

$$\min C = \sum_{n=1}^{N} F_n(P_n) = \sum_{n=1}^{N} (a_n P_n^2 + b_n P_n + c_n), \tag{1}$$

where C represents the total fuel cost; P_n is the active power of the nth thermal power unit; N is the number of grid-connected units; a_n, b_n, and c_n denote the generation cost coefficients; and $F_n(P_n)$ is the consumption characteristic function, which represents the generation cost.

However, the steam inlet valve of the turbine during the actual operation of the thermal power unit can suddenly open, resulting in a pulling phenomenon, also known as the valve-point effect. A pulsation is superimposed on the unit consumption curve due to the valve-point effect. At this point, the optimization objective function of the generation cost is (Liu *et al.*, 2022; Xiong *et al.*, 2022a)

$$F_n(P_n) = a_n P_n^2 + b_n P_n + c_n + \left| e_n \times \sin(f_n \times (P_n^{\min} - P_n)) \right|, \tag{2}$$

where e_n and f_n are the valve-point effect coefficients, and $P_n^{\min}$ is the lower limit of active power for the nth unit. Figure 1a and b show the fuel cost function with and without valve-point effect, respectively.

2.2. Constraints

2.2.1. *Load balance constraint*

$$\sum_{n=1}^{N} P_n - P_D - P_{\text{loss}} = 0, \tag{3}$$

where P_{loss} and P_D represent the network loss and system load, respectively. P_{loss} is related to the active power of the unit set, transmission line parameters, and the system structure. The B-factor approach is used to calculate it:

$$P_{\text{loss}} = \sum_{i=1}^{N} \sum_{j=1}^{N} P_i B_{ij} P_j + \sum_{i=1}^{N} B_{0i} P_i + B_{00}, \tag{4}$$

where B_{ij}, B_{0i}, and B_{00} are coefficients.

2.2.2. *Unit capacity constraint*

$$P_n^{\min} \le P_n \le P_n^{\max},\ n = 1, 2, \cdots, N, \tag{5}$$

where $P_n^{\max}$ and $P_n^{\min}$ denote the upper and lower output limits, respectively.

2.2.3. *Ramp rate limits*

$$\begin{cases} P_n - P_n^{pr} \le UR_n \\ P_n^{pr} - P_n \le DR_n \end{cases},\ n = 1, 2, \cdots, N, \tag{6}$$

where UR_n and DR_n represent the increasing limit and decreasing limit, respectively, and P_n^{pr} denotes the output of the previous moment.

2.2.4. *Prohibited operating zones*

Prohibited operating zones (POZs) lead to a solution space with disjoint feasible regions for individual generators. The constraints can be described as follows (Xu *et al.*, 2022):

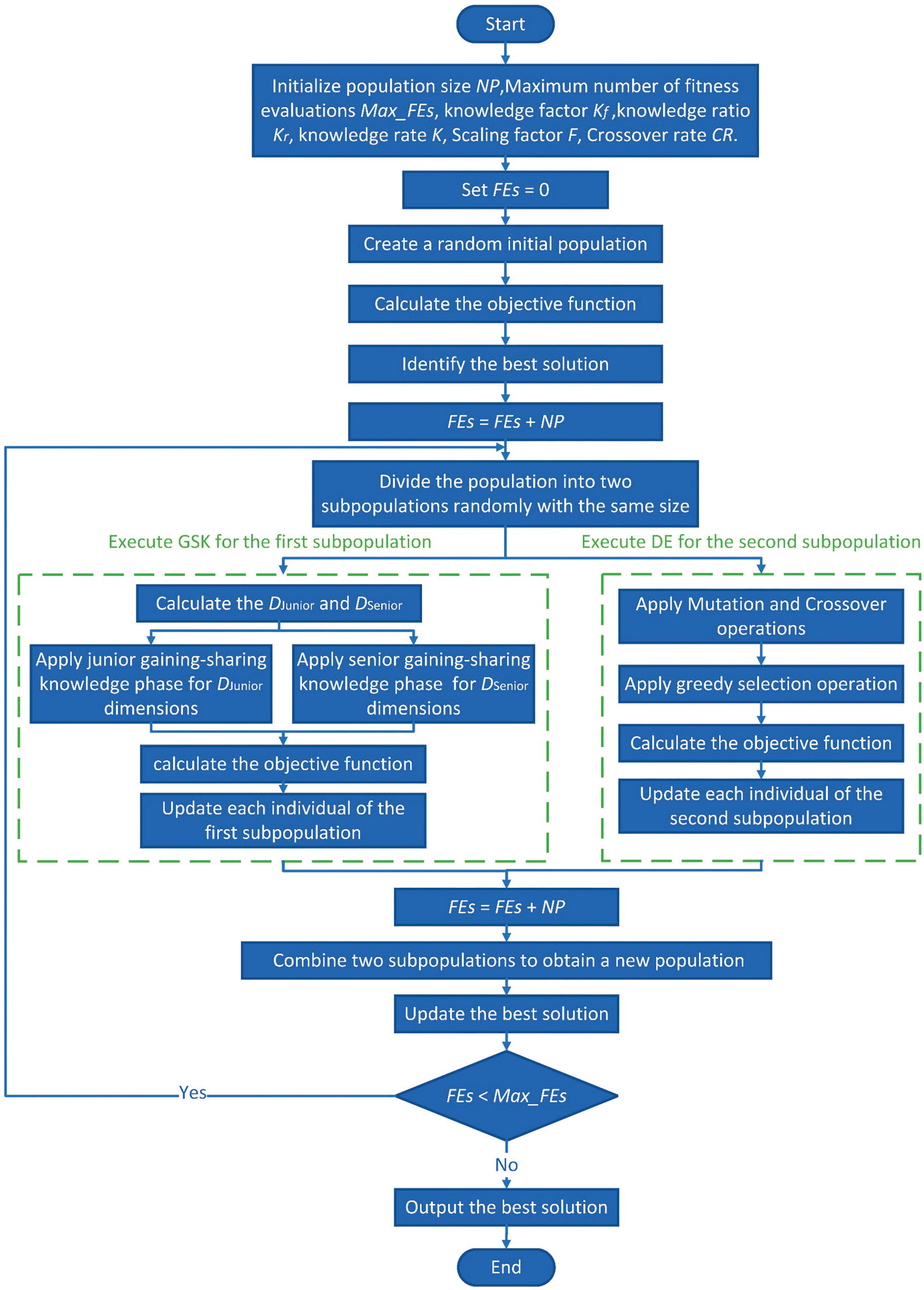

Figure 2: Flowchart of the proposed GSK-DE.

$$P_n \in \begin{cases} P_{n,\min} \le P_n \le P^L_{n,1} \\ P^U_{n,k-1} \le P_n \le P^L_{n,k} \\ P^U_{n,pz_n} \le P_n \le P_{n,\max} \end{cases}, \; k = 2, 3, \dots pz_n, \tag{7}$$

where pz_n is the number of POZs of the nth generator, and $P^L_{n,k}$ and $P^U_{n,k}$ are the lower and upper bounds of the kth POZ of the nth generator, respectively.

3. Proposed GSK-DE

3.1. Gaining–sharing knowledge-based algorithm

GSK was inspired by the process of acquiring and sharing knowledge throughout the human life cycle (Mohamed *et al.*, 2020). There are two important stages in this method including the junior acquiring–sharing and senior acquiring–sharing.

In GSK, a population includes NP individuals and an individual x_i in a D-dimensional space is expressed as $x_i = (x_{i1}, x_{i2}, \dots, x_{iD})$. D_{junior} and D_{senior} are to use the junior and senior schemes to update the dimension of individuals, respectively. They are given by

$$D_{\text{junior}} = D \times \left(1 - \frac{G}{GEN}\right)^K \tag{8}$$

$$D_{\text{senior}} = D - D_{\text{junior}}, \tag{9}$$

where G and GEN are the current generation and the maximum number of generations, respectively. K denotes the knowledge rate.

3.1.1. *Junior acquiring and sharing phase*

We first arrange all individuals in ascending order, i.e., $x_{\text{best}}, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_{\text{worst}}$. For an individual x_i, the nearest best neighbor x_{i-1} and the worst neighbor x_{i+1} are selected as the source of knowledge. If x_i is the best individual, the order would be $x_{\text{best}}, x_{\text{best}+1}, x_{\text{best}+2}$. If x_i is the worst one, the order would be $x_{\text{worst}-2}, x_{\text{worst}-1}, x_{\text{worst}}$. Finally, if a knowledge ratio k_r (> 0) is smaller than a random number rand(0, 1) generated within (0, 1), x_i retains its original value. Otherwise, the individual x_i is updated according to the following formula:

$$x_i^{\text{new}} = \begin{cases} x_i + k_f \cdot [(x_{i-1} - x_{i+1}) + (x_r - x_i)], & \text{if } f(x_i) > f(x_r) \\ x_i + k_f \cdot [(x_{i-1} - x_{i+1}) + (x_i - x_r)], & \text{otherwise} \end{cases}, \tag{10}$$

where $k_f > 0$ denotes the knowledge factor.

3.1.2. *Senior acquiring and sharing phase*

The individuals are divided into three groups after sorting in ascending order: the best group, the middle group, and the worst group. After that, two individuals of the best and worst $NP \times 100p\%$ are selected as the gaining part, while one of the middle $NP - (2 \times 100p\%)$ is randomly selected as the sharing part. Finally, if $\text{rand}(0, 1) > k_r$, x_i retains its original value. Otherwise, the individual x_i is updated according to the following formula:

$$x_i^{\text{new}} = \begin{cases} x_i + k_f \cdot \left[(x_{p-\text{best}} - x_{p-\text{worst}}) + (x_m - x_i)\right], & \text{if } f(x_i) > f(x_m) \\ x_i + k_f \cdot \left[(x_{p-\text{best}} - x_{p-\text{worst}}) + (x_i - x_m)\right], & \text{otherwise} \end{cases}, \tag{11}$$

where $p \in [0, 1]$ is a constant. $x_{p-\text{best}}$ denotes a random individual in the best group. $x_{p-\text{worst}}$ denotes a random individual in the worst group. x_m belongs to the middle group.

Algorithm 1 gives the pseudo-code of GSK.

Input: Control parameters: Population: NP; Dimension: D; Knowledge factor: K_f; Knowledge ratio: K_r; Knowledge rate: K; Content: p.
Output: The optimal solution
Create a random initial population x_i, $i = 1, 2, \cdots, NP$
Evaluate $f(x_i)$, $\forall i$, $i=1, 2, \cdots, NP$
Set $FEs = NP$
Determine the best solution
While $FEs <$ Max_ FEs **do**
  For i to NP **do**
    Calculate the D_{Junior} with Eq.(8)
    Calculate the D_{Senior} with Eq.(9)
    // Junior gaining-sharing knowledge phase
    For $j = 1$ to D_{Junior} **do**
      Calculate x_i^{new} with Eq.(10)
    End
    // Senior gaining-sharing knowledge phase
    For $j = 1$ to D_{Senior} **do**
      Calculate x_i^{new} with Eq.(11)
    End
    Evaluate the objection function value for x_i^{new}
    $FEs = FEs + 1$
    If $f(x_i^{new}) < f(x_i)$ **then**
      $x_i = x_i^{new}$
    End
  End
  Update the best solution
End while

3.2. Differential evolution

DE guides the direction of optimization search by mutation, crossover, and selection among individuals in a population (Storn & Price, 1997).

First, two different individuals are selected randomly and their vector differences are scaled on another different random individual to generate a mutant vector:

$$v_i = x_{r1} + F \cdot (x_{r2} - x_{r3}), \tag{12}$$

where v_i denotes the mutant vector. F denotes the scaling factor, and $r1$, $r2$, and $r3$ and i are mutually different individuals.

Then, the crossover operation between $x_{i,j}$ and the mutant vector $v_{i,j}$ is performed to generate the test individual $u_{i,j}$, as shown below:

$$u_{i,j} = \begin{cases} v_{i,j}, \text{ if } (\text{rand}(0,1) \le CR) & \text{or}(j = j_{\text{rand}}) \\ x_{i,j}, & \text{otherwise} \end{cases}, \tag{13}$$

where CR denotes the crossover rate. j_{rand} represents a random dimension of $\{1, 2, \dots, D\}$.

Subsequently, the better individual is selected for the new population by comparing the objective functions of individuals u_i and x_i, as shown below:

$$x_i^{\text{new}} = \begin{cases} u_i, \text{ if}(f(u_i) \le f(x_i)) \\ x_i, & \text{otherwise} \end{cases}. \tag{14}$$

Table 1: Defining parameters.

Algorithm	Parameters
ABC (Karaboga, 2010)	NP = 50, *FoodNumber* = NP/2, *limit* = 100
BBO (Simon, 2008)	NP = 50, *MutationProb* = 0.01
CLPSO (Liang *et al.*, 2006)	NP = 50, $c = 1.49445$
CPA (Tu *et al.*, 2021)	NP = 50
DE (Storn & Price, 1997)	NP = 50, CR = 0.9, F = 0.6
DE-WOA (Xiong et al., 2018)	NP = 50, CR = 0.9, F = 0.6, p = rand(0, 1)
EHO (Wang *et al.*, 2015)	NP = 50, *nClan* = 5, $\alpha = 0.5$, $\beta = 0.1$
EWA (Wang *et al.*, 2018)	NP = 50, *mutate* = 0.01, $\alpha = 0.98$, $\beta = 1$, $\gamma = 0.9$
GSK (Mohamed *et al.*, 2020)	NP = 50, $k_f = 0.5$, $k_r = 0.3$, K = 35, $p = 0.1$
HHO (Heidari *et al.*, 2019)	NP = 50, p = rand(0, 1)
HGS (Yang *et al.*, 2021)	NP = 50
IJAYA (Chen *et al.*, 2020)	NP = 50, rand1 = rand2 = rand(0, 1)
MBO (Wang *et al.*, 2019)	NP = 50, $S_{max} = 1$, BAR = p = 5/12, *peri* = 1.2
MS (Wang, 2018)	NP = 50, $\varphi = [(5)^{1/2} - 1)/2]$, $\beta = 1.5$, $S_{max} = 1$
PPSO (Ghasemi *et al.*, 2019)	NP = 50, $w = 0$
PSO (Kennedy & Eberhart, 1995)	NP = 50, $c = 1.49618$, $w = 0.7298$
QILDE (Xiong *et al.*, 2020)	NP = 50, CR = 0.9, F = 0.6
RUN (Ahmadianfar *et al.*, 2021)	NP = 50, $a = 20$, $b = 12$
SATLBO (Yu *et al.*, 2017)	NP = 50
SMA (Li *et al.*, 2020)	NP = 50, $z = 0.03$
WOA (Mirjalili and Lewis, 2016)	NP = 50, $r1 = r2 = p$ = rand(0, 1)
GSK-DE	NP = 50, $k_f = 0.5$, $k_r = 0.3$, K = 35, $p = 0.1$, CR = rand(0, 1), F = 0.1+0.9∗rand(0, 1)

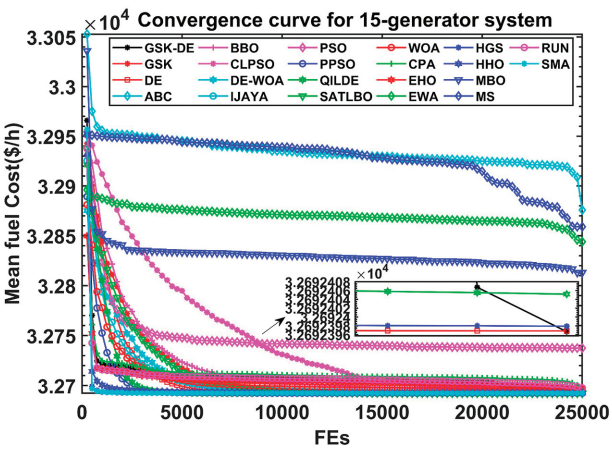

Figure 3: Convergence curves for case 1.

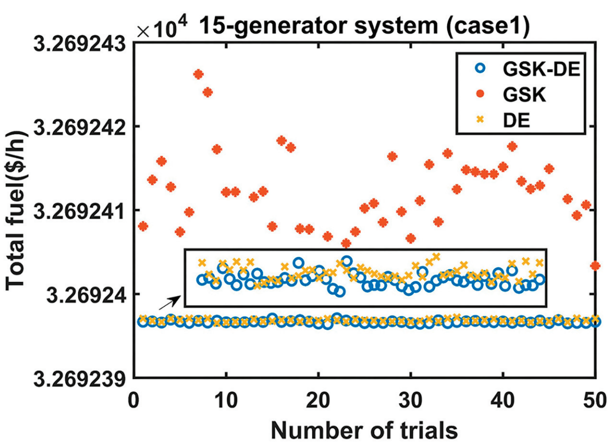

Figure 4: Fuel cost distribution for case 1.

Algorithm 2 reports the pseudo-code of DE.

```
Input: Control parameters: Population: NP; Dimension: D;
       Scaling factor: F; Crossover rate: CR.
Output: The optimal solution
Create a random initial population x_i, i = 1, 2, ⋯, NP
Evaluate f(x_i), ∀i, i=1, 2, ⋯, NP
Set FEs = NP
Determine the best solution
While FEs < Max_ FEs do
    For i to NP do
        // Mutation and crossover phases
        For j=1 to D do
            Calculate v_i,j with Eq.(12)
            Calculate u_i,j with Eq.(13)
        End
        // Greedy selection phase
        Evaluate the objection function value for u_i
        FEs = FEs + 1
        If f(u_i) < f(x_i) then
            x_i = u_i
        End
    End
    Update the best solution
End while
```

4. Hybrid GSK/DE

4.1. Motivations

(1) For metaheuristic algorithms, exploration and exploitation are key building elements. Although DE has an aptitude for exploring and discovering the global optimal area, it takes a long time to solution convergence since it uses three random individuals to update the target individual, as shown in equation (12). On the contrary, as shown in equations (10) and (11), GSK starts from the target individual, and combines the subtraction of

Table 2: Simulation results ($/h) for case 1.

Algorithm	Minimum	Maximum	Average	Standard deviation	Average CPU time (s)
ABC	32 758.0768	33 021.9735	32 875.8413	51.6310	0.5625
BBO	32 693.0586	32 698.6505	32 694.9159	1.1923	**0.3294**
CLPSO	32 692.5796	32 696.3495	32 693.9316	0.8474	0.7245
DE-WOA	32 692.4010	32 692.4423	32 692.4137	0.0082	0.4952
IJAYA	32 692.4035	32 692.4794	32 692.4284	0.0159	0.4035
PSO	32 704.8486	32 783.3540	32 737.4793	16.2014	0.6264
PPSO	32 692.3965	32 692.6409	32 692.4384	0.0783	0.7625
QILDE	32 692.3983	32 692.4779	32 692.4052	0.0114	0.8873
SATLBO	32 692.3970	32 692.5946	32 692.4280	0.0468	0.6987
WOA	32 692.4684	32 724.9243	32 696.8186	7.1013	0.6529
CPA	32 692.5358	32 721.3543	32 696.9899	8.2989	0.6133
EHO	32 693.3268	32 703.0514	32 696.1468	2.2703	0.6484
EWA	32 754.5062	33 024.0618	32 869.8206	57.7430	0.5568
HGS	32 692.3967	32 692.4125	32 692.3979	0.0034	0.4410
HHO	32 692.5315	32 692.5755	32 692.5438	0.0171	0.5837
MBO	32 701.8191	32 937.6803	32 835.1581	45.3703	0.5387
MS	32 696.3569	32 740.0716	32 715.3513	13.2282	0.6757
RUN	32 692.5414	32 713.4684	32 697.6633	4.8035	0.6459
SMA	32 692.4022	32 692.5522	32 692.4207	0.0203	0.5246
DE	32 692.3966	32 692.3972	32 692.3969	1.5630E-04	0.3679
GSK	32 692.4020	32 692.4276	32 692.4121	0.0053	0.3929
GSK-DE	**32 692.3964**	**32 692.3971**	**32 692.3967**	**1.4927E-04**	0.3969

Table 3: Simulation results ($/h) for case 2.

Algorithm	Minimum	Maximum	Average	Standard deviation	Average CPU time (s)
HHS (Fesanghary & Ardehali, 2009)	9417 325	9417 466	9417 336	NA	NA
HS (Fesanghary & Ardehali, 2009)	9419 960	9427 466	9421 056	NA	NA
PSO_TVAC (Chaturvedi *et al.*, 2009)	9500 448.307	NA	NA	NA	NA
New_PSO (Chaturvedi *et al.*, 2009)	9516 448.312	NA	NA	NA	NA
ABC	9417 245.7095	9417 319.2998	9417 270.8379	16.2102	0.4519
BBO	9418 421.4408	9423 359.2923	9420 377.7761	1423.1055	0.6225
CLPSO	9417 814.0080	9419 100.9668	9418 472.1254	265.7059	1.0356
DE-WOA	9417 261.5796	9417 341.5836	9417 297.1218	20.4463	0.6241
IJAYA	9417 395.9228	9417 592.2990	9417 461.7801	45.8085	0.4557
PSO	9420 941.5801	9443 360.6343	9428 233.1719	4471.0368	0.4870
PPSO	9417 236.6978	9417 253.4038	9417 240.9340	3.6754	1.0669
QILDE	9417 236.6386	9417 243.5056	9417 238.5651	1.4328	0.8036
SATLBO	9417 236.6672	9417 263.3901	9417 242.2887	6.2402	0.7057
WOA	9417 394.3570	9418 227.4177	9417 667.7679	178.1776	0.6075
CPA	9419 171.3897	9429 951.9933	9423 317.4902	2380.2966	0.7346
EHO	9509 316.1209	9588 229.6955	9555 390.0068	17 990.9888	0.6594
EWA	9441 166.2716	9502 976.1277	9465 842.0383	13 326.3170	0.8850
HGS	9417 235.7878	9417 374.2202	9417 239.5745	19.9346	0.6314
HHO	9476 382.6741	9550 358.2837	9507 898.0946	12 217.8650	0.8276
MBO	9423 178.6072	9454 216.1709	9436 372.6519	6460.7401	0.6164
MS	9431 431.0684	9485 715.3040	9453 822.8676	13 701.9560	0.6317
RUN	9418 955.5821	9475 216.8806	9426 220.5137	8450.3115	0.6813
SMA	9417 236.2740	9417 277.5731	9417 238.5444	6.3277	0.5815
DE	9417 237.7371	9417 242.4068	9417 239.1236	1.0005	0.4871
GSK	9417 237.0052	9417 238.4658	9417 237.5702	0.3627	**0.4404**
GSK-DE	**9417 235.7863**	**9417 236.2969**	**9417 236.0026**	**0.0903**	0.4405

neighboring individuals and the difference between the best individual and the worst individual to guide the search. This search strategy makes GSK apt at finding local solutions with higher searching capability and faster convergence. However, it will reduce the population diversity and thus easily lead to falling into local optimum. Therefore, our primary motivation is to integrate GSK and DE to avoid early convergence and speed up the global search.

(2) The second motivation is that heterogeneity is a valid means to improve the optimization performance. The basic idea for hybridization is to maximize the complementary

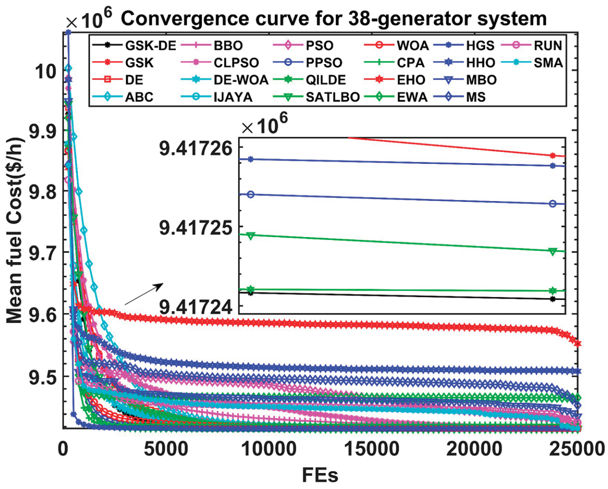

Figure 5: Convergence curves for case 2.

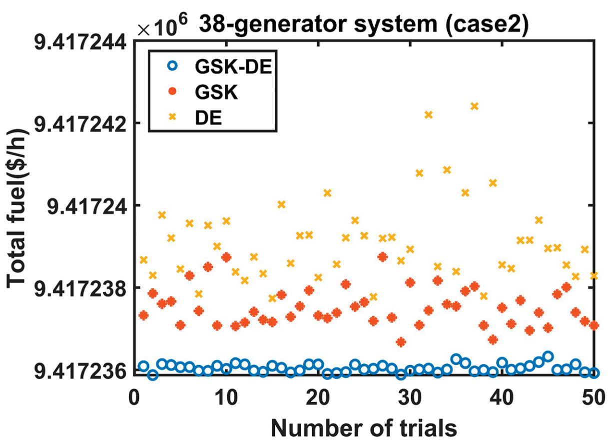

Figure 6: Fuel cost distribution for case 2.

strengths of various algorithms so that difficult optimization problems can be solved more efficiently.

(3) The third motivation is that this is the first time of the GSK and DE being hybridized as far as we know in solving the ED problems.

(4) The fourth motivation is that the dual-population evolutionary framework used in this work is beneficial for solving optimization problems. It can ensure both the population diversity and convergence speed, and thus to equilibrate the local search and global search abilities effectively. In addition, the optimization efficiency can be improved with two subpopulations based on parallel search.

4.2. Dual-population evolution framework

Step 1: In each iteration, the population is first divided into two subpopulations randomly with the same size.

Step 2: These two subpopulations are optimized using different optimization algorithms. DE is used for one group for maintaining the population diversity. The other group uses the GSK algorithm to boost the convergence.

Step 3: At the later stage, the two groups of generated offspring individuals are compared with their respective parent individuals to compete the opportunity into the next iteration.

Step 4: Finally, a new population is formed by combing the fittest members of these two subpopulations.

Compared with the basic DE and GSK, the proposed GSK-DE does not increase the overall computational complexity significantly. For the basic DE and GSK, the fitness of each individual is calculated once in each iteration and the corresponding complexity is $O(D)$. Therefore, the complexity of all individuals in each iteration is $O(NP \cdot D)$ and the total complexity is $O(GEN \cdot NP \cdot D)$. For the proposed GSK-DE, the additional complexity is coming from the population partition and its complexity is $O(NP)$. Hence, the total complexity of GSK-DE is $O(GEN \cdot NP \cdot (D+1))$, which is comparable to that of GSK and DE.

Algorithm 3 shows the pseudo-code of GSK-DE and Fig. 2 shows the flowchart of GSK-DE. It can be seen that GSK-DE employs a dual-population evolution framework to hybridize both GSK and DE harmoniously. In this way, the consistency of evolution is ensured. Moreover, it can improve the efficiency based on parallel search, thus balancing the exploitation and exploration effectively.

```
Input: Control parameters: Population: NP; Dimension:D; Knowledge factor: K_f ;
       Knowledge ratio: K_r ; Knowledge rate: K ; content: p. Scaling factor: F;
       Crossover rate: CR.
Output: The optimal solution
Create a random initial population x_i, i = 1, 2, ···, NP
Evaluate f(x_i), ∀i, i = 1, 2, ···, NP
Set FEs = NP
Determine the best solution
While FEs < Max_FEs do
    Divide the population into two equal subpopulations randomly
    // Execute GSK for the first subpopulation
    For i to NP/2 do
        Calculate the D_Junior with Eq.(8)
        Calculate the D_Senior with Eq.(9)
        // Junior gaining-sharing knowledge phase
        For j=1 to D_Junior do
            Calculate x_i^new with Eq.(10)
        End
        // Senior gaining-sharing knowledge phase
        For j=1 to D_Senior do
            Calculate x_i^new with Eq.(11)
        End
        Evaluate the objection function value for f(x_i^new)
        If f(x_i^new) < f(x_i) then
            x_i = x_i^new
        End
    End
    // Execute DE for the second subpopulation
    For i to NP/2 do
        // Mutation and crossover phases
        For j = 1 to D do
            Calculate v_i,j with Eq.(12)
            Calculate u_i,j with Eq.(13)
        End
        // Greedy selection phase
        Evaluate the objection function value for u_i
        FEs = FEs + 1
        If f(u_i) < f(x_i) then
            x_i = u_i
        End
    End
    Combine two subpopulations to obtain a new population
    Update the best solution
End while
```

5. Case Studies and Results

GSK-DE is applied to six ED problems with different characteristics. The first case (case 1) contains 15 units. The second case (case 2) contains 38 units. The third case (case 3) contains 40 units. The fourth case (case 4) triples case 3 to a system with 120 units. The fifth case (case 5) contains 110 units, which is a highly complex

Table 4: Simulation results ($/h) for case 3.

Algorithm	Minimum	Maximum	Average	Standard deviation	Average CPU time (s)
TSARGA (Subbaraj *et al.*, 2011)	121 463.07	124 296.54	122 928.31	NA	NA
NCS (Ciornei & Kyriakides, 2011)	121 430	121 645	121 525	35.758	NA
DEC-SQP (Coelho & Mariani, 2006)	121 741.98	NA	122 295.13	NA	NA
GA-PS-SQP (Alsumait *et al.*, 2010)	121 458.14	NA	122 039	NA	NA
QPSO (Meng *et al.*, 2009)	121 448.21	NA	122 225.07	NA	NA
FAPSO-NM (Niknam, 2010)	121 418.3	121 419.8	121 418.803	NA	NA
CCPSO (Park *et al.*, 2009)	121 412.5362	121 534.4934	121 454.3296	32.4898	NA
aBBOmDE (Lohokare *et al.*, 2012)	121 414.8734	121 568.3254	121 487.8532	NA	NA
IHSWM (Pandi *et al.*, 2011)	121 416.2652	121 855.5521	121 553.4208	90.1271	NA
HMAPSO (Kumar *et al.*, 2011)	121 412.57	**121 415.78**	**121 413.3879**	NA	NA
SQP-CLPSO (Wang *et al.*, 2010)	121 515.8	121 820.4	121 677.2	62.8009	NA
NPSO (Tsai & Yen, 2011)	121 855.114	NA	NA	NA	NA
IFEP (Sinha *et al.*, 2003)	122 624.35	125 740.63	123 382	NA	NA
EPUSPSO (Wu *et al.*, 2016)	122 897.69	123 121.78	NA	NA	NA
RN-MAPSO (Tang *et al.*, 2012)	122 402.28	123 256.18	122 654.53	NA	NA
THS (Al-Betar *et al.*, 2016)	121 425.15	NA	121 528.65	50.48	NA
CTLBO (He *et al.*, 2015)	121 553.83	122 116.18	121 790.23	150	NA
CSO (Guo & Xiong, 2017)	121 467.40	121 748.20	121 550.16	59.39	NA
ICSO (Guo & Xiong, 2017)	121 423.03	121 529.29	121 469.25	31.99	NA
ABC	121 418.1073	121 516.3408	121 457.1952	29.5663	4.0896
BBO	121 540.4482	122 207.8294	121 719.2843	116.1870	6.9617
CLPSO	121 462.9499	121 639.6798	121 537.1507	34.3218	12.3080
DE-WOA	121 421.2417	121 878.1390	121 530.7721	78.7937	6.0436
IJAYA	121 420.9110	121 756.4038	121 562.1977	94.8717	4.4808
PSO	121 603.9138	122 035.0364	121 784.9532	105.5785	4.0662
PPSO	121 536.7732	123 353.8999	122 148.4971	388.7109	12.3938
QILDE	121 415.5596	121 785.7928	121 552.1303	75.1653	13.6618
SATLBO	121 489.2483	122 546.4144	122 027.3334	223.0096	10.4820
WOA	121 570.9713	123 218.3472	121 954.2960	314.3370	9.9919
CPA	121 973.8518	123 835.7823	122 763.9552	420.3704	9.3274
EHO	122 759.1348	125 432.4822	124 214.1911	697.6182	6.8156
EWA	122 508.0536	125 610.3693	123 616.1339	704.8387	7.4515
HGS	121 509.5770	122 629.4682	121 940.3440	238.8059	7.6705
HHO	121 734.2013	121 814.2720	121 790.0531	33.4326	7.9784
MBO	122 917.7923	123 459.3824	123 194.1727	142.8960	6.7350
MS	121 951.2208	124 078.8569	122 838.7590	475.3829	8.4685
RUN	122 739.6918	125 664.0636	124 068.4510	712.7100	6.9118
SMA	121 443.1512	121 522.2280	121 448.9615	36.3049	7.5212
DE	121 420.8933	121 587.3117	121 479.7156	42.8857	**4.0309**
GSK	121 421.1734	121 605.7015	121 481.4039	53.8158	4.6404
GSK-DE	**121 412.5346**	121 506.6590	121 451.1886	**28.1149**	7.0656

system. The last case (case 6) expands case 5 three times to a system with 330 units. For these six cases, Table 1 shows the peer compared algorithms. Fifty separate experiments have been conducted for them. For the 40-unit system, the maximum number of function evaluations (Max_FEs) is set to $D*10\,000$, for the 120-unit system, the Max_FEs is set to $D*5000$, and for the other three systems, the Max_FEs is set to $D*1000$. All the numerical studies were conducted on a 2.3-GHz i7 PC with 8 GB of RAM in MATLAB R2018b.

5.1. Simulation results

5.1.1. *Case 1: 15-unit system*

This case consists of 15 generators considering ramp rate limits, POZ, and transmission network losses (Liu *et al.*, 2022). Figure 3 shows the convergence curves. Figure 4 depicts the detailed distribution of the total fuel costs of GSK, DE, and GSK-DE over 50 independent trials. Table S-1 in the supplemental file gives the best solution produced by GSK-DE. Table 2 shows the fuel cost information for involved algorithms.

5.1.2. *Case 2: 38-unit system*

This is a conventional ED (Coelho & Mariani, 2006). On the one hand, as indicated in Table 3, GSK-DE is compared to those achieved by the methods listed in Table 1. On the other hand, the reported results of some advanced algorithms are also used to compare with GSK-DE. Figure 5 shows the convergence curves. Figure 6 depicts the detailed distribution of the total generation costs of GSK, DE, and GSK-DE over 50 independent trials. Table S-2 in the supplemental file gives the best solution produced by GSK-DE.

5.1.3. *Case 3: 40-unit system*

The valve-point effects are crucial for this complex system containing a large number of local minima. The system's specifics can be found in Sinha *et al.* (2003). Table 4 shows the fuel cost information for involved algorithms. Figure 7 depicts the cost convergence curves. Figure 8 depicts the detailed distribution of the total generation costs of GSK, DE, and GSK-DE over 50 independent trials.

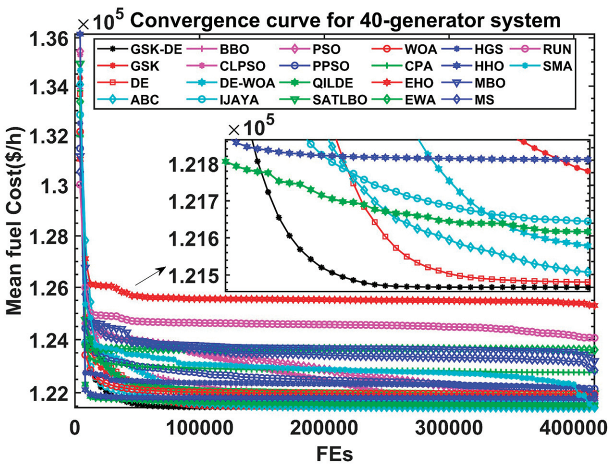

Figure 7: Convergence curves for case 3.

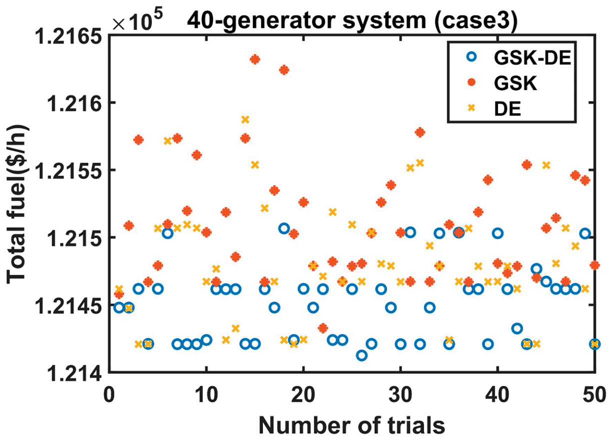

Figure 8: Fuel cost distribution for case 3.

Table S-3 in the supplemental file shows the best solution produced by GSK-DE.

5.1.4. *Case 4: 120-unit system*

This system is a threefold extension of case 3. The load requirement of this system is 31 500 MW. The objective function of case 4 considers the valve-point effects. This case's solution space is broader and more complicated, placing particularly high demands on the search capability of the solution method to get rid of the adsorption of local optimal points to locate the global optimal region. Table 5 shows the data of fuel cost. Figure 9 depicts the convergence curves, while Fig. 10 depicts the fuel cost distribution. Table S-4 in the supplemental file shows the best solution found by GSK-DE.

5.1.5. *Case 5: 110-unit system*

As a traditional ED, the specific details are available in Vishwakarma and Dubey (2012). Table 6 shows the fuel cost data. Figure 11 depicts the convergence curves, while Fig. 12 depicts the distribution of fuel costs. Table S-5 in the supplemental file shows the best solution found by GSK-DE.

5.1.6. *Case 6: 330-unit system*

By tripling the 110-unit system, the final case comprises 330 units. Table 7 shows the fuel cost data. Figures 13 and 14, respectively, depict the convergence curves and the distribution of fuel costs. Table S-6 in the supplemental file shows the best solution produced by GSK-DE.

5.2. Comparisons

5.2.1. *Solution quality*

Overall, the suggested GSK-DE method shows better results as shown in Tables 2–7.

For case 1, although its fuel cost characteristics are quadratic, its solution space is discontinuous due to the restrictions of the POZ. Even so, GSK-DE shows an encouraging performance, obtaining the best cost that is 32 692.3964 $/h. As can be seen from Table 2, all the minimum, maximum, and average costs obtained by GSK-DE are minimal compared to other methods.

For case 2, it is a typical single-peaked (i.e., unimodal) multiconstraint optimization problem. The solution method is required to have a good local search capability to improve the search accuracy. In Table 3, it is shown that GSK-DE demonstrates its superiority in the ability to search the solution space locally.

For cases 3 and 4, the valve-point effects make systems' local minima massive. To solve this difficult situation, the ED solution methods need to be highly explorative. Tables 4 and 5 show that GSK-DE achieves the minimum costs of 121 412.5346 and 364 277.7156 $/h, respectively. They are better than or highly competitive to the other peer algorithms. Compared with the original GSK and DE, the optimization performance of GSK-DE is superior, which indicates that the algorithm can effectively get rid of the adsorption of local minima and has stronger global search ability. This demonstrates that the designed enhancements successfully overcome the inadequacies of the GSK and DE algorithms. The comparison of the optimization results with some previous literature also further demonstrates the superiority of GSK-DE.

For cases 5 and 6, their mathematical models are larger scale single-peaked multiconstraint optimization problems. They are difficult for many solution methods because their solution spaces are broader and it is hard to achieve accurate enough solutions for them. Even so, as can be seen from Tables 6 and 7, GSK-DE shows an encouraging performance to obtain the optimal fuel costs of 197 988.1765 and 593 965.9437 $/h, respectively, superior to the other methods consistently. Additionally, other fuel cost indexes provided by GSK-DE are considerably highly competitive as well compared to other algorithms. Moreover, GSK-DE is much better than the original GSK and DE, reflecting that GSK-DE has better local search capability compared to the original GSK and DE. The comparison with some previous optimization results in the literature also further verifies the validity of GSK-DE.

From the above comparative analysis, obviously, GSK-DE successfully equilibrates local and global optimization capabilities and is highly competitive with other peer algorithms, stressing its ability in improving greater solution quality.

5.2.2. *Convergence*

The convergence speed and the accuracy of the convergence curve are two important indicators to examine the effectiveness of an algorithm. Figures 3, 5, 7, 9, 11, and 13 provide the convergence curves of GSK-DE and the other involved algorithms. As shown in Figs. 5, 11, and 13, it is more uniformly and comprehensively for the GSK-DE algorithm to exploit the solutions with an overall faster speed. As shown in Figs. 3, 7, and 9, the GSK-DE algorithm is more capable of skipping local extrema and can reach better points. Moreover, in the beginning, although some algorithms such as QILDE, PPSO, IJAYA, HHO, and WOA converge faster

Table 5: Simulation results ($/h) for case 4.

Algorithm	Minimum	Maximum	Average	Standard deviation	Average CPU time (s)
CSO (Guo & Xiong, 2017)	364 503.37	365 130.43	364 649.05	131.33	NA
ICSO (Guo & Xiong, 2017)	364 388.17	364 747.08	364 575.56	92.69	NA
ABC	364 408.7275	364 672.2249	364 540.8825	67.2764	**12.1653**
BBO	364 991.6232	366 564.9237	365 405.5925	298.4494	35.6322
CLPSO	365 336.9212	365 830.1879	365 600.6670	102.9413	39.7063
DE-WOA	364 606.1018	365 584.6349	364 956.5202	201.1553	31.5502
IJAYA	364 497.5480	366 446.2050	365 135.5332	422.8091	15.7579
PSO	370 850.1989	376 531.6036	372 560.5972	1029.7200	28.9983
PPSO	365 144.3749	369 353.2516	367 143.6853	780.6499	62.8449
QILDE	364 533.0035	365 464.7403	364 989.6659	256.4785	33.0904
SATLBO	366 272.4874	372 354.8907	368 222.7421	1273.4160	72.5774
WOA	364 690.2144	366 376.8266	365 363.3156	361.2936	67.1191
CPA	386 874.8069	396 255.9785	389 924.0067	1830.5308	38.4326
EHO	381 891.6190	383 948.7903	383 221.7287	428.9756	36.5271
EWA	376 804.4041	381 546.4833	379 300.2626	846.8604	43.0291
HGS	364 724.8655	365 962.5821	364 845.7630	208.5128	33.5762
HHO	373 567.3347	382 407.1716	374 095.9787	1208.1994	31.5833
MBO	376 804.4041	381 546.4833	379 300.2626	846.8604	41.2990
MS	383 632.6185	386 779.2156	385 676.2605	616.5349	38.3902
RUN	381 497.1444	387 000.0227	384 491.8464	1184.0562	37.8294
SMA	365 068.3306	365 208.3205	365 099.5326	**35.4442**	42.6249
DE	364 346.9152	**364 544.1285**	364 441.4859	44.8628	17.2175
GSK	364 414.9566	365 554.4175	364 882.4932	302.5617	18.0328
GSK-DE	**364 277.7156**	364 590.8126	**364 405.8217**	62.2550	23.7193

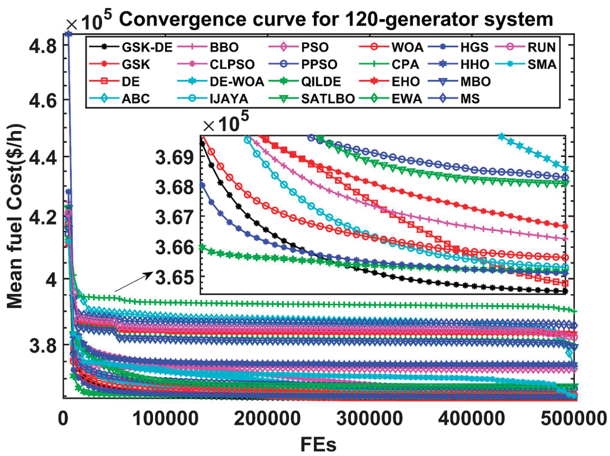

Figure 9: Convergence curves for case 4.

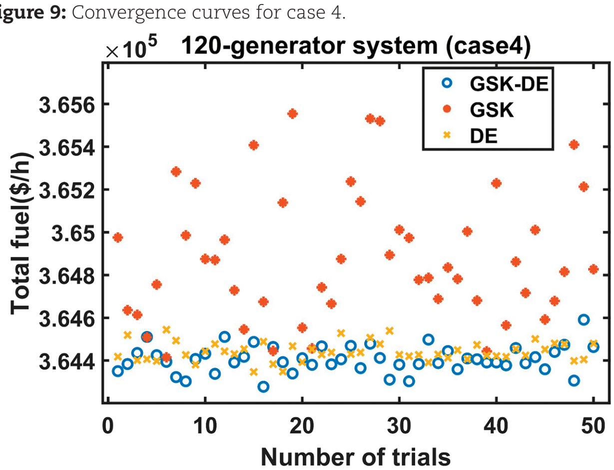

Figure 10: Fuel cost distribution for case 4.

than GSK-DE slightly, they quickly fall into local search and are surpassed by GSK-DE. As for DE and GSK, although their convergence trends are similar to that of GSK-DE, they converge more slowly. In a word, GSK-DE is able to obtain the optimal solution faster and has better convergence characteristics.

5.2.3. *Robustness*

This is a significant metric for evaluating consistency of meta-heuristic algorithms. The standard deviation values generated by GSK-DE are 1.4927E-04, 0.0903, 28.1149, 62.2550, 0.000117, and 0.3071, respectively. They are lower than other peer algorithms and most of the reported ED solutions, demonstrating that the GSK-DE algorithm has greater robustness. Moreover, Figs. 4, 6, 8, 10, 12, and 14 show that GSK-DE also has superior robustness.

In addition, we use an error percentage metric (Mokarram *et al.*, 2019) defined below to further illustrate the robustness of the proposed GSK-DE more closely.

$$\text{Error\%} = \frac{|\text{final value} - \text{best value}|}{\text{best value}} \times 100\% \tag{15}$$

The values of error% for 50 independent runs of case 1 are shown in Fig. S-1 in the supplementary file. It shows the great robustness of GSK-DE, as the error% is smaller than 0.000000025%. Similarly, its outstanding robustness in other cases is also shown in Figs. S-2 to S-6 in the supplementary file.

Furthermore, we also draw the box plots of solution distribution of GSK, DE, and GSK-DE in Figs. S-7 to S-12 in the supplementary file to compare their robustness more visually. Apparently, the average value of the results achieved by GSK-DE (see the red lines in Figs. S-7 to S-12 in the supplementary file) is smaller than those provided by GSK and DE. For the 38-unit system, the obtained minimum, average, maximum, and standard deviation of GSK-DE are better than the original GSK and DE algorithms. For the 15-unit, 40-unit, and 120-unit systems, GSK-DE is the best, DE

Table 6: Simulation results ($/h) for case 5.

Algorithm	Minimum	Maximum	Average	Standard deviation	Average CPU time (s)
SAB (Vishwakarma & Dubey, 2012)	206 912.91	NA	207 764.73	NA	NA
SAF (Vishwakarma & Dubey, 2012)	207 380.52	NA	207 813.37	NA	NA
SA (Vishwakarma & Dubey, 2012)	198 352.64	NA	201 595.19	NA	NA
ORCCRO (Bhattacharjee *et al.*, 2014)	198 016.29	198 016.89	198 016.32	NA	NA
CSO (Guo & Xiong, 2017)	198 023.98	198 053.82	198 036.74	7.04	NA
ICSO (Guo & Xiong, 2017)	197 995.21	198 000.95	197 995.21	2.01	NA
ABC	197 997.3271	198 026.6394	198 010.6699	6.4941	**1.9911**
BBO	198 340.4729	198 982.9466	198 564.0391	118.8951	5.5624
CLPSO	198 129.5661	198 236.9431	198 194.4675	20.8834	4.7928
DE-WOA	198 047.1180	198 139.0507	198 091.9985	23.4335	4.6651
IJAYA	198 054.2047	198 143.6200	198 093.4490	15.9289	2.9727
PSO	201 905.6200	209 791.5546	204 302.4878	1360.0922	3.0141
PPSO	197 989.8482	198 006.3411	197 996.5654	4.2178	5.2909
QILDE	197 992.0676	197 997.8835	197 994.1675	1.1097	5.0795
SATLBO	197 988.2243	198 033.1583	197 995.0874	8.9412	11.6996
WOA	198 076.7834	198 309.0437	198 177.7725	55.4238	10.3354
CPA	202 262.5564	206 581.3918	204 144.6012	1121.1992	5.1781
EHO	210 272.7513	214 303.8607	212 878.3043	742.8523	4.9144
EWA	199 061.7317	200 461.6279	199 607.6286	355.6936	3.4511
HGS	198 092.1899	199 442.8731	198 773.6190	336.2303	5.6740
HHO	198 067.1479	198 148.2223	198 072.2729	11.2656	4.9276
MBO	202 708.5163	204 339.7074	203 520.0706	375.4799	5.4713
MS	202 133.0647	207 127.0233	204 008.8140	977.0969	4.7808
RUN	198 890.3281	204 879.7653	200 464.9064	1142.2070	5.5744
SMA	197 992.9751	198 095.3547	198 007.1438	20.0246	3.8547
DE	198 056.9931	198 140.3893	198 103.9231	20.8581	3.1076
GSK	197 990.8224	197 992.0896	197 991.3532	0.3043	3.0012
GSK-DE	**197 988.1765**	**197 988.1769**	**197 988.1767**	**0.000117**	2.8828

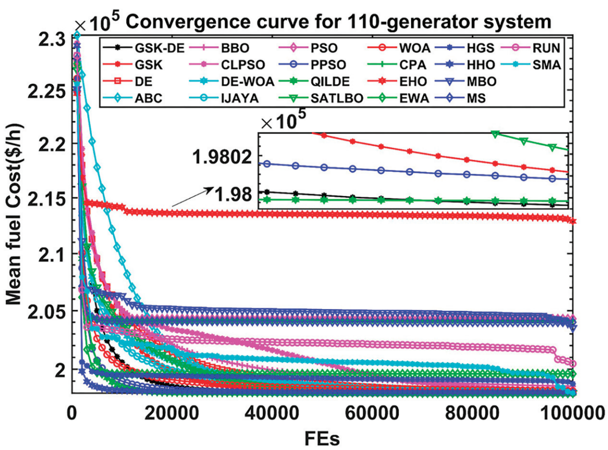

Figure 11: Convergence curves for case 5.

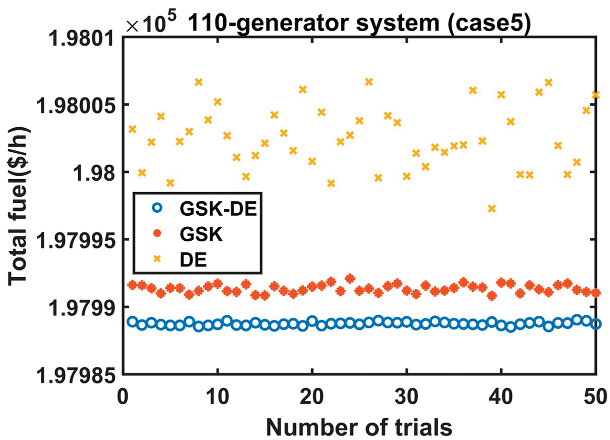

Figure 12: Fuel cost distribution for case 5.

is the second, and GSK is the worst. For the 110-unit and 330-unit systems, GSK-DE is the best, GSK is second, and DE is the worst. Besides, the results obtained by GSK-DE in 50 independent runs are more compact than those of GSK and DE (see the blue boxes in Figs. S-7 to S-12 in the supplementary file). This presents again that GSK-DE is considerably superior to GSK and DE in achieving higher quality solutions.

5.2.4. *Computation time*

Tables 2–7 present the average CPU time results for the proposed algorithm and other compared algorithms. The average CPU time values of GSK-DE are 0.3969, 0.4405, 7.0656, 23.7193, 2.8828, and 33.7255 s, respectively, for these six cases. On the whole, GSK-DE consumes slightly more time than GSK and DE, but the difference is not significant. This is mainly because the computational complexity of GSK-DE is comparable to that of GSK and DE, as shown in the analysis of Subsection 4.2. In addition, the computation speed of GSK-DE is highly competitive compared with other algorithms. In fact, the computation speed of GSK-DE is fully capable of meeting the practical needs of the power system, since the actual system generally requires only 15 min for a single computation (Xiong & Shi, 2018).

Table 7: Simulation results ($/h) for case 6.

Algorithm	Minimum	Maximum	Average	Standard deviation	Average CPU time (s)
CSO (Guo & Xiong, 2017)	594 212.72	594 302.51	594 258.67	27.82	NA
ICSO (Guo & Xiong, 2017)	593 992.51	594 022.00	594 007.06	7.14	NA
ABC	594 055.2351	594 140.5939	594 090.1884	18.9323	**14.0444**
BBO	597 338.7332	599 605.4973	598 391.3135	497.2178	54.4164
CLPSO	594 475.1565	594 662.3522	594 571.3372	38.2978	34.6367
DE-WOA	594 786.7548	595 464.1029	595 080.6109	158.0850	60.8164
IJAYA	594 296.9131	594 477.6573	594 368.8508	33.6956	26.0707
PSO	618 546.2544	644 331.2420	629 262.1670	5732.5116	34.3193
PPSO	593 983.1759	594 037.8721	593 999.3795	12.5244	49.2944
QILDE	594 003.6041	594 021.4598	594 010.7590	3.6309	56.8853
SATLBO	593 981.8074	594 451.1088	594 073.9685	95.0426	168.7618
WOA	594 375.3398	594 765.5284	594 594.2647	87.6611	136.1464
CPA	620 023.4754	649 682.0042	628 627.2683	6468.0730	51.9682
EHO	660 761.8463	670 895.6998	665 805.6153	2251.3015	44.9668
EWA	620 068.2582	639 510.1738	627 298.7521	4323.0321	45.2550
HGS	602 162.7923	605 797.2165	604 067.8779	844.1009	50.8761
HHO	594 156.7112	594 378.0305	594 171.9674	33.7799	36.8185
MBO	629 030.3297	633 192.7696	631 346.5197	936.5686	47.4090
MS	595 871.2843	601 496.1260	596 663.6509	1098.4535	53.7810
RUN	604 438.5492	648 148.0540	613 284.5219	5877.7215	42.4866
SMA	594 235.3988	595 442.7436	594 486.2273	257.3242	53.0160
DE	594 275.6698	594 634.4288	594 407.5995	61.5660	27.5203
GSK	593 968.2729	593 969.5760	593 968.9423	0.3325	24.7301
GSK-DE	**593 965.9437**	**593 967.0437**	**593 966.5379**	**0.3071**	33.7255

Table 8: *P* values and Wilcoxon's rank sum test result.

GSK-DE vs.	15-unit	38-unit	40-unit	120-unit	110-unit	330-unit
DE	2.0438E-07 +	7.9022E-08 +	4.7617E-05 +	1.1106E-05 +	3.1193E-15 +	3.4298E-17 +
GSK	2.9711E-03 +	2.1285E-02 +	7.4929E-13 +	3.2857E-33 +	7.5657E-04 +	1.6921E-02 +
ABC	3.8966E-18 +	1.4074E-07 +	4.8770E-01 +	9.1689E-08 +	1.6491E-09 +	3.5229E-12 +
BBO	2.1073E-03 +	1.5122E-17 +	2.3033E-19 +	9.2046E-20 +	3.2505E-20 +	1.1983E-24 +
CLPSO	7.1072E-06 +	2.1148E-18 +	1.2147E-20 +	1.4973E-26 +	4.2697E-20 +	1.0739E-21 +
DE-WOA	1.9512E-05 +	3.1743E-01 +	1.3627E-10 +	2.3381E-22 +	2.3127E-14 +	7.3102E-21 +
IJAYA	8.7968E-07 +	1.1672E-06 +	4.4879E-12 +	2.7828E-13 +	2.7893E-09 +	1.5003E-10 +
PSO	5.5962E-18 +	4.4248E-24 +	4.9846E-26 +	8.9772E-32 +	6.9694E-27 +	1.9481E-29 +
PPSO	1.8529E-12 +	6.1386E-01 ≈	5.2507E-26 +	1.4584E-26 +	4.9987E-02 +	2.4372E-03 +
QILDE	8.6915E-11 +	1.6586E-02 +	3.2145E-01 +	4.1167E-12 +	2.0870E-01 ≈	3.8639E-02 +
SATLBO	2.0752E-05 +	4.5394E-06 +	2.4468E-21 +	4.5498E-28 +	9.3930E-05 +	4.7418E-11 +
WOA	8.7378E-04 +	2.0988E-09 +	4.5704E-20 +	6.8101E-17 +	4.0704E-13 +	1.7296E-15 +
CPA	8.5761E-14 +	4.1394E-18 +	3.8966E-18 +	4.0162E-18 +	4.1394E-18 +	4.0162E-18 +
EHO	1.8631E-07 +	4.0152E-18 +	4.2663E-18 +	4.2663E-18 +	4.3971E-18 +	4.0162E-18 +
EWA	4.0162E-18 +	4.0159E-18 +	3.8966E-18 +	4.1394E-18 +	4.0162E-18 +	4.0162E-18 +
HGS	3.7589E-18 +	7.3431E-17 +	2.8380E-15 +	3.8966E-18 +	3.2876E-08 +	3.8966E-18 +
HHO	4.9610E-18 +	3.7589E-18 +	4.0751E-08 +	3.8966E-18 +	8.4462E-03 +	6.0843E-07 +
MBO	3.8966E-18 +	3.8966E-18 +	6.7024E-18 +	4.1394E-18 +	4.0162E-18 +	4.3971E-18 +
MS	4.0161E-18 +	5.9889E-17 +	6.5361E-17 +	3.8966E-18 +	3.8966E-18 +	4.0162E-18 +
RUN	1.8331E-07 +	7.3431E-17 +	7.3338E-18 +	4.0162E-18 +	7.6014E-15 +	8.5191E-18 +
SMA	4.2663E-18 +	7.3431E-18 +	6.5041E-18 +	7.3431E-17 +	4.5645E-13 +	4.6042E-17 +

5.2.5. *Statistical test*

We perform the Wilcoxon's rank sum test at a 5% confidence level to test the performance of GSK-DE. It is helpful to confirm the significance of the difference between the results of GSK-DE and other peer algorithms in Table 1.

Table 8 gives the *P* values and the significant differences. The symbol "+" indicates that GSK-DE produces a remarkably superior result than the compared algorithms as the *P* value is smaller than 0.05. The symbol "≈" indicates no statistical difference between two algorithms. Clearly, GSK-DE has an observably statistical advantage over most algorithms in most cases. To be specific, in case 2 and case 5, PPSO and QILDE achieve statistically comparable results compared with GSK-DE, respectively. However, in the other three cases, GSK-DE has greater advantages. It beats all the compared algorithms consistently. These results clearly show again that the GSK-DE algorithm can obtain higher quality solutions and the improvement in its performance is statistically significant.

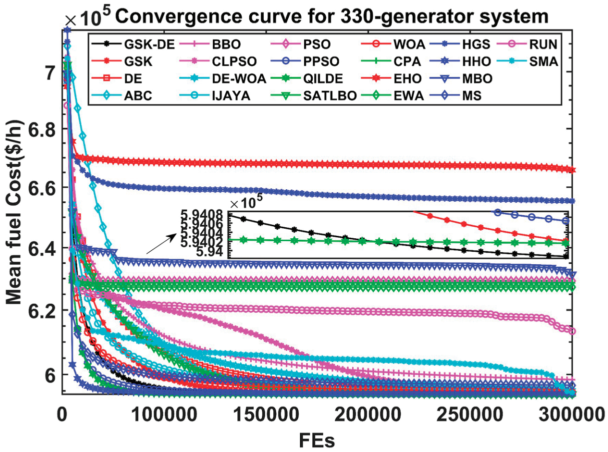

Figure 13: Convergence curves for case 6.

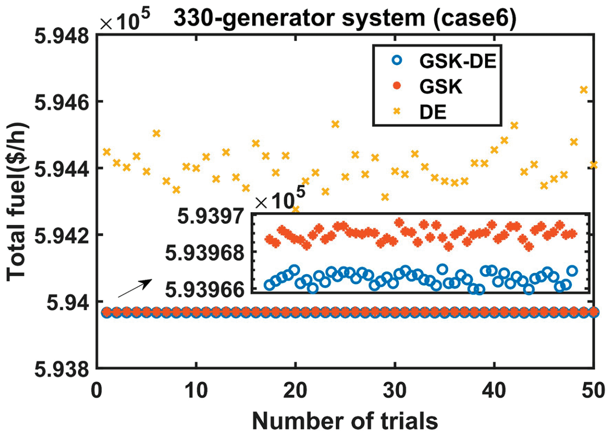

Figure 14: Fuel cost distribution for case 6.

5.3. Effect of population size on GSK-DE

Determining an appropriate population size *NP* is an important part of solving optimization problems using metaheuristics. An appropriate *NP* is helpful in improving the effectiveness of an algorithm. To assess the impact of *NP* on GSK-DE, a simulation is conducted on the 120-unit system (case 4). The value is set from 30 to 300, as presented in Fig. 15. Figure 15 shows the box plots of fuel cost distribution of 50 independent trials versus the population size. The best solutions related to each population size are presented in Table S-7 in the supplementary file.

Obviously, a small or a large *NP* is detrimental to the effectiveness of GSK-DE. The reason could be that although a large *NP* can enhance the population diversity, it decreases the likelihood of finding more promising solutions, which could lead to inefficiencies, especially for multipeaked problems. On the other hand, a smaller *NP* can result in premature solution and failure to find the optimal solution. Hence, the use of a medium size is highly suggested.

6. Conclusions and Future Works

As we know, no one method can achieve satisfactory solutions for all optimization problems. Although GSK and DE are not sufficient enough to obtain optimal solutions for complex ED problems, a reasonable combination of them can lead to powerful

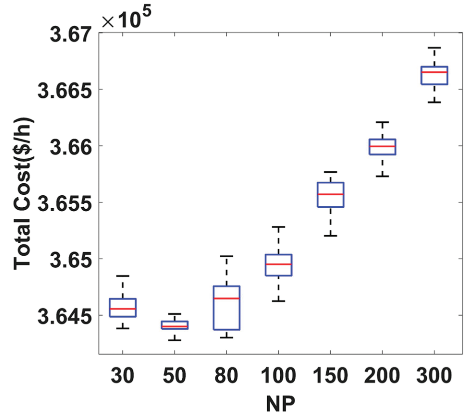

Figure 15: Influence of *NP* on GSK-DE for the 120-unit system.

performance. This work presents an enhanced algorithm namely GSK-DE by hybridizing DE and GSK based on a dual-population evolutionary framework for solving large-scale ED problems. The simulation results show that

(1) GSK-DE can quickly obtain better solutions for these cases, especially for more complex and higher dimensional problems. The results also show that GSK-DE is better than or comparable to many previously proposed solutions in both robustness and quality.
(2) Comparing the standard deviation values in these six cases, the standard deviation without the valve-point effects is much smaller than the standard deviation with the valve-point effects. This is because the valve-point effects make the model highly multipeaked, which causes the solution method easily falling into different local extremes in different trials.
(3) Moreover, a medium-sized population is strongly suggested for GSK-DE.

In summary, GSK-DE achieves more optimal solutions for different complex ED problems. In the near future, we will focus on solving various ED problems containing renewable energy sources and extending the proposed GSK-DE to solve them.

Supplementary data

Supplementary data is available at *JCDENG* Journal online.

Acknowledgments

The authors would like to thank the editor and the reviewers for their constructive comments. This research was funded by the Natural Science Foundation of Guizhou Province (QiankeheBasic-ZK[2022]General121), the National Natural Science Foundation of China (52167007), the Innovation Foundation of Guizhou University Institute of Engineering Investigation & Design Co., Ltd. (GuiDaKanCha[2022]03), the Modern Power System and Its Digital Technology Engineering Research Center (QianJiaoJi[2022]043), and the Open Project Program of Fujian Provincial Key Laboratory of Intelligent Identification and Control of Complex Dynamic System (2022A0008).

Conflict of interest statement

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Agrawal, P., Ganesh, T., & Mohamed, A. W. (2022). Solving knapsack problems using a binary gaining sharing knowledge-based optimization algorithm. *Complex and Intelligent Systems*, **8**, 43–63. https://doi.org/10.1007/s40747-021-00351-8.

Ahmadianfar, I., Heidari, A. A., Gandomi, A. H., Chu, X., & Chen, H. (2021). RUN beyond the metaphor: An efficient optimization algorithm based on Runge Kutta method. *Expert Systems with Applications*, **181**, 115079. https://doi.org/10.1016/j.eswa.2021.115079.

Al-Betar, M. A., Awadallah, M. A., Khader, A. T., & Bolaji, A. L. A. (2016). Tournament-based harmony search algorithm for non-convex economic load dispatch problem. *Applied Soft Computing*, **47**, 449–459. https://doi.org/10.1016/j.asoc.2016.05.034.

Alsumait, J. S., Sykulski, J. K., & Al-Othman, A. K. (2010). A hybrid GA–PS–SQP method to solve power system valve-point economic dispatch problems. *Applied Energy*, **87**, 1773–1781. https://doi.org/10.1016/j.apenergy.2009.10.007.

Azzam, M., Selvan, S., Lefèvre, A., & Absil, P. A. (2014). Mixed integer programming to globally minimize the economic load dispatch problem with valve-point effect. *preprint arXiv:1407.4261v1*. https://doi.org/10.48550/arXiv.1407.4261.

Bhattacharjee, K., Bhattacharya, A., & Dey, S. H. (2014). Oppositional real coded chemical reaction optimization for different economic dispatch problems. *International Journal of Electrical Power & Energy Systems*, **55**, 378–391. https://doi.org/10.1016/j.ijepes.2013.09.033.

Chansareewittaya, S. (2017). Hybrid BA/TS for economic dispatch considering the generator constraint. In *2017 International Conference on Digital Arts, Media and Technology (ICDAMT)* (pp. 115–119). IEEE. https://doi.org/10.1109/ICDAMT.2017.7904946.

Chaturvedi, K. T., Pandit, M., & Srivastava, L. (2009). Particle swarm optimization with time varying acceleration coefficients for non-convex economic power dispatch. *International Journal of Electrical Power & Energy Systems*, **31**, 249–257. https://doi.org/10.1016/j.ijepes.2009.01.010.

Chen, C., Zou, D., & Li, C. (2020). Improved Jaya algorithm for economic dispatch considering valve-point effect and multi-fuel options. *IEEE Access*, **8**, 84981–84995. https://doi.org/10.1109/ACCESS.2020.2992616.

Chen, J., & Marrani, I. H. (2020). An efficient new hybrid ICA-PSO approach for solving large scale non-convex multi area economic dispatch problems. *Journal of Electrical Engineering & Technology*, **15**, 1127–1145. https://doi.org/10.1007/s42835-020-00416-7.

Chiang, C. L. (2007). Genetic-based algorithm for power economic load dispatch. *IET Generation, Transmission & Distribution*, **1**, 261–269. https://doi.org/10.1049/iet-gtd:20060130.

Ciornei, I., & Kyriakides, E. (2011). A GA-API solution for the economic dispatch of generation in power system operation. *IEEE Transactions on Power Systems*, **27**, 233–242. https://doi.org/10.1109/TPWRS.2011.2168833.

Coelho, L. S., & Mariani, V. C. (2006). Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect. *IEEE Transactions on Power Systems*, **21**, 989–996. https://doi.org/10.1109/TPWRS.2006.873410.

Fesanghary, M., & Ardehali, M. M. (2009). A novel meta-heuristic optimization methodology for solving various types of economic dispatch problem. *Energy*, **34**, 757–766. https://doi.org/10.1016/j.energy.2009.02.007.

Gao, D., Wang, G. G., & Pedrycz, W. (2020). Solving fuzzy job-shop scheduling problem using DE algorithm improved by a selection mechanism. *IEEE Transactions on Fuzzy Systems*, **28**, 3265–3275. https://doi.org/10.1109/TFUZZ.2020.3003506.

Ghasemi, M., Akbari, E., Rahimnejad, A., Razavi, S. E., Ghavidel, S., & Li, L. (2019). Phasor particle swarm optimization: A simple and efficient variant of PSO. *Soft Computing*, **23**, 9701–9718. https://doi.org/10.1007/s00500-018-3536-8.

Guo, Y., & Xiong, G. (2017). Large scale power system economic dispatch based on an improved competitive swarm optimizer. *Power System Protection and Control*, **45**, 97–103. https://doi.org/10.7667/PSPC161194.

Hassan, S. A., Ayman, Y. M., Alnowibet, K., Agrawal, P., & Mohamed, A. W. (2020). Stochastic travelling advisor problem simulation with a case study: A novel binary gaining–sharing knowledge-based optimization algorithm. *Complexity*, **2020**, 6692978. https://doi.org/10.1155/2020/6692978.

He, X., Rao, Y., & Huang, J. (2015). A novel algorithm for economic load dispatch of power systems. *Neurocomputing*, **171**, 1454–1461. https://doi.org/10.1016/j.neucom.2015.07.107.

Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., & Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future Generation Computer Systems*, **97**, 849–872. https://doi.org/10.1016/j.future.2019.02.028.

Hindi, K. S., Ghani, Ab, & M., R. (1991). Dynamic economic dispatch for large scale power systems: A Lagrangian relaxation approach. *International Journal of Electrical Power & Energy Systems*, **13**, 51–56. https://doi.org/10.1016/0142-0615(91)90018-Q.

Jabr, R. A., Coonick, A. H., & Cory, B. J. (2000). A homogeneous linear programming algorithm for the security constrained economic dispatch problem. *IEEE Transactions on Power Systems*, **15**, 930–936. https://doi.org/10.1109/59.871715.

Karaboga, D. (2010). Artificial bee colony algorithm. *Scholarpedia*, **5**, 6915. https://doi.org/10.4249/scholarpedia.6915.

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of ICNN'95—International Conference on Neural Networks*(Vol. **4**, pp. 1942–1948). IEEE. https://doi.org/10.1109/ICNN.1995.488968.

Khamsawang, S., & Jiriwibhakorn, S. (2010). DSPSO–TSA for economic dispatch problem with nonsmooth and noncontinuous cost functions. *Energy Conversion and Management*, **51**, 365–375. https://doi.org/10.1016/j.enconman.2009.09.034.

Kumar, R., Sharma, D., & Sadu, A. (2011). A hybrid multi-agent based particle swarm optimization algorithm for economic power dispatch. *International Journal of Electrical Power & Energy Systems*, **33**, 115–123. https://doi.org/10.1016/j.ijepes.2010.06.021.

Li, C., Xiong, G., Fu, X., Mohamed, A. W., Yuan, X., Al-Betar, M. A., & Suganthan, P. N. (2022). Takagi–Sugeno fuzzy based power system fault section diagnosis models via genetic learning adaptive GSK algorithm. *Knowledge-Based Systems*, **255**, 109773. https://doi.org/10.1016/j.knosys.2022.109773.

Li, S., Chen, H., Wang, M., Heidari, A. A., & Mirjalili, S. (2020). Slime mould algorithm: A new method for stochastic optimization. *Future Generation Computer Systems*, **111**, 300–323. https://doi.org/10.1016/j.future.2020.03.055.

Liang, J. J., Qin, A. K., Suganthan, P. N., & Baskar, S. (2006). Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Transactions on Evolutionary Computation*, **10**, 281–295. https://doi.org/10.1109/TEVC.2005.857610.

Liu, T., Xiong, G., Mohamed, A. W., & Suganthan, P. N. (2022). Opposition-mutual learning differential evolution with hybrid mutation strategy for large-scale economic load dispatc h problems with valve-point effects and multi-fuel options. *Information Sciences*, **609**, 1721–1745. https://doi.org/10.1016/j.ins.2022.07.148.

Lohokare, M. R., Panigrahi, B. K., Pattnaik, S. S., Devi, S., & Mohapatra, A. (2012). Neighborhood search-driven accelerated biogeography-based optimization for optimal load dispatch. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, **42**, 641–652. https://doi.org/10.1109/TSMCC.2012.2190401.

Meng, K., Wang, H. G., Dong, Z., & Wong, K. P. (2009). Quantum-inspired particle swarm optimization for valve-point economic load dispatch. *IEEE Transactions on Power Systems*, **25**, 215–222. https://doi.org/10.1109/TPWRS.2009.2030359.

Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. *Advances in Engineering Software*, **95**, 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008.

Mohamed, A. W., Hadi, A. A., & Mohamed, A. K. (2020). Gaining–sharing knowledge based algorithm for solving optimization problems: A novel nature-inspired algorithm. *International Journal of Machine Learning and Cybernetics,* **11**, 1501–1529. https://doi.org/10.1007/s13042-019-01053-x.

Mokarram, M. J., Niknam, T., Aghaei, J., Shafie-khah, M., & Catalao, J. P. (2019). Hybrid optimization algorithm to solve the nonconvex multiarea economic dispatch problem. *IEEE Systems Journal*, **13**, 3400–3409. https://doi.org/10.1109/JSYST.2018.2889988.

Muralidharan, S., Srikrishna, K., & Subramanian, S. (2007). Self-adaptive dynamic programming technique for economic power dispatch. *International Journal of Power and Energy Systems*, **27**, 340. https://doi.org/10.2316/Journal.203.2007.4.203-3673.

Nanda, J., Hari, L., & Kothari, M. L. (1994). Economic emission load dispatch with line flow constraints using a classical technique. *IEE Proceedings – Generation, Transmission and Distribution*, **141**, 1–10. https://doi.org/10.1049/ip-gtd:19949770.

Niknam, T. (2010). A new fuzzy adaptive hybrid particle swarm optimization algorithm for non-linear, non-smooth and non-convex economic dispatch problem. *Applied Energy*, **87**, 327–339. https://doi.org/10.1016/j.apenergy.2009.05.016.

Pandi, V. R., Panigrahi, B. K., Mohapatra, A., & Mallick, M. K. (2011). Economic load dispatch solution by improved harmony search with wavelet mutation. *International Journal of Computational Science and Engineering*, **2**, 122–131. https://doi.org/10.1504/IJCSE.2011.041220.

Park, J. B., Jeong, Y. W., Shin, J. R., & Lee, K. Y. (2009). An improved particle swarm optimization for nonconvex economic dispatch problems. *IEEE Transactions on Power Systems*, **25**, 156–166. https://doi.org/10.1109/TPWRS.2009.2030293.

Parouha, R. P., & Das, K. N. (2016). DPD: An intelligent parallel hybrid algorithm for economic load dispatch problems with various practical constraints. *Expert Systems with Applications*, **63**, 295–309. https://doi.org/10.1016/j.eswa.2016.07.012.

Simon, D. (2008). Biogeography-based optimization. *IEEE Transactions on Evolutionary Computation*, **12**, 702–713. https://doi.org/10.1109/TEVC.2008.919004.

Sinha, N., Chakrabarti, R., & Chattopadhyay, P. K. (2003). Evolutionary programming techniques for economic load dispatch. *IEEE Transactions on Evolutionary Computation*, **7**, 83–94. https://doi.org/10.1109/TEVC.2002.806788.

Storn, R., & Price, K. (1997). Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization, Berkeley*, **11**, 341–359. https://doi.org/10.1023/A:1008202821328.

Subbaraj, P., Rengaraj, R., & Salivahanan, S. (2011). Enhancement of self-adaptive real-coded genetic algorithm using Taguchi method for economic dispatch problem. *Applied Soft Computing*, **11**, 83–92. https://doi.org/10.1016/j.asoc.2009.10.019.

Takeang, C., & Aurasopon, A. (2019). Multiple of hybrid lambda iteration and simulated annealing algorithm to solve economic dispatch problem with ramp rate limit and prohibited operating zones. *Journal of Electrical Engineering & Technology*, **14**, 111–120. https://doi.org/10.1007/s42835-018-00001-z.

Tang, X., Zhou, H., Li, J., & Zhou, W. (2012). An economic load dispatch method of power system based on multi-agent particle swarm optimization algorithm. *Power System Protection and Control*, **10**, 42–47. https://doi.org/10.3969/j.issn.1674-3415.2012.10.008.

Tsai, M. T., & Yen, C. W. (2011). The influence of carbon dioxide trading scheme on economic dispatch of generators. *Applied Energy*, **88**, 4811–4816. https://doi.org/10.1016/j.apenergy.2011.06.025.

Tu, J., Chen, H., Wang, M., & Gandomi, A. H. (2021). The colony predation algorithm. *Journal of Bionic Engineering*, **18**, 674–710. https://doi.org/10.1007/s42235-021-0050-y.

Victoire, T. A. A., & Jeyakumar, A. E. (2006). A tabu search based hybrid optimization approach for a fuzzy modelled unit commitment problem. *Electric Power Systems Research*, **76**, 413–425. https://doi.org/10.1016/j.epsr.2005.08.004.

Vishwakarma, K. K., & Dubey, H. M. (2012). Simulated annealing based optimization for solving large scale economic load dispatch problems. *International Journal of Engineering Research and Technology*, **1**, 1–8. https://doi.org/10.48175/ijarsct-832.

Wang, G. G. (2018). Moth search algorithm: A bio-inspired metaheuristic algorithm for global optimization problems. *Memetic Computing*, **10**, 151–164. https://doi.org/10.1007/s12293-016-0212-3.

Wang, G. G., Deb, S., & Coelho, L. D. S. (2015). Elephant herding optimization. In *2015 3rd International Symposium on Computational and Business Intelligence (ISCBI)* (pp. 1–5). IEEE. https://doi.org/10.1109/ISCBI.2015.8.

Wang, G. G., Deb, S., & Coelho, L. D. S. (2018) Earthworm optimization algorithm: A bio-inspired metaheuristic algorithm for global optimization problems. *International Journal of Bio-Inspired Computation*, **12**, 1–22. https://doi.org/10.1504/IJBIC.2018.093328.

Wang, G. G., Deb, S., & Cui, Z. (2019). Monarch butterfly optimization. *Neural Computing and Applications*, **31**, 1995–2014. https://doi.org/10.1007/s00521-015-1923-y.

Wang, G. G., Gao, D., & Pedrycz, W. (2022). Solving multi-objective fuzzy job-shop scheduling problem by a hybrid adaptive differential evolution algorithm. *IEEE Transactions on Industrial Informatics*, **18**, 8519–8528. https://doi.org/10.1109/TII.2022.3165636.

Wang, G. G., & Tan, Y. (2020). Improving metaheuristic algorithms with information feedback models. *IEEE Transactions on Cybernetics*, **49**, 542–555. https://doi.org/10.1109/TCYB.2017.2780274.

Wang, Y., Li, B., & Yuan, B. (2010). Hybrid of comprehensive learning particle swarm optimization and SQP algorithm for large scale economic load dispatch optimization of power system. *Science China Information Sciences*, **53**, 1566–1573. https://doi.org/10.1007/s11432-010-4034-5.

Wu, C. B., Li, H. M., Liu, D., Wu, Z. Y., & Wu, L. (2016). Application of improved particle swarm optimization algorithm to power system economic load dispatch. *Power System Protection and Control*, **44**, 44–48. https://doi.org/10.7667/PSPC151119.

Xiong, G., Li, L., Mohamed, A. W., Yuan, X., & Zhang, J. (2021). A new method for parameter extraction of solar photovoltaic models using gaining–sharing knowledge based algorithm. *Energy Reports*, **7**, 3286–3301. https://doi.org/10.1016/j.egyr.2021.05.030.

Xiong, G., & Shi, D. (2018). Orthogonal learning competitive swarm optimizer for economic dispatch problems. *Applied Soft Computing*, **66**, 134–148. https://doi.org/10.1016/j.asoc.2018.02.019.

Xiong, G., Shuai, M., & Hu, X. (2022a). Combined heat and power economic emission dispatch using improved bare-bone multi-objective particle swarm optimization. *Energy*, **244**, 123108. https://doi.org/10.1016/j.energy.2022.123108.

Xiong, G., Yuan, X., Mohamed, A. W., Chen, J., & Zhang, J. (2022c). Improved binary gaining–sharing knowledge-based algorithm with mutation for fault section location in distribution networks. *Journal of Computational Design and Engineering*, **9**, 393–405. https://doi.org/10.1093/jcde/qwac007.

Xiong, G., Yuan, X., Mohamed, A. W., & Zhang, J. (2022b). Fault section diagnosis of power systems with logical operation binary gaining–sharing knowledge-based algorithm. *International Journal of Intelligent Systems*, **37**, 1057–1080. https://doi.org/10.1002/int.22659.

Xiong, G., Zhang, J., Shi, D., Zhu, L., & Yuan, X. (2020). Parameter extraction of solar photovoltaic models via quadratic interpolation learning differential evolution. *Sustainable Energy & Fuels*, **4**, 5595–5608. https://doi.org/10.1039/D0SE01000F.

Xiong, G., Zhang, J., Yuan, X., Shi, D., He, Y., & Yao, G. (2018). Parameter extraction of solar photovoltaic models by means of a hybrid differential evolution with whale optimization algorithm. *Solar Energy*, **176**, 742–761. https://doi.org/10.1016/j.solener.2018.10.050.

Xu, S., Xiong, G., Mohamed, A. W., & Bouchekara, H. R. (2022). Forgetting velocity based improved comprehensive learning particle swarm optimization for non-convex economic dispatch problems with valve-point effects and multi-fuel options. *Energy*, **256**, 124511. https://doi.org/10.1016/j.energy.2022.124511.

Yang, Y., Chen, H., Heidari, A. A., & Gandomi, A. H. (2021). Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts. *Expert Systems with Applications*, **177**, 114864. https://doi.org/10.1016/j.eswa.2021.114864.

Yu, K., Chen, X., Wang, X., & Wang, Z. (2017). Parameters identification of photovoltaic models using self-adaptive teaching-learning-based optimization. *Energy Conversion and Management*, **145**, 233–246. https://doi.org/10.1016/j.enconman.2017.04.054.

Zhang, M., Hu, Z., Suo, J., & Zhang, Z. (2013). A new hybrid algorithm for economic dispatch considering the generator constraints. In *2013 IEEE International Conference of IEEE Region 10 (TENCON 2013)* (pp. 1–4). IEEE. https://doi.org/10.1109/TENCON.2013.6718481.