energies

Article

A High-Performance Stochastic Fractal Search Algorithm for Optimal Generation Dispatch Problem

Ly Huu Pham [1], Minh Quan Duong [2], Van-Duc Phan [3,*], Thang Trung Nguyen [1,*] and Hoang-Nam Nguyen [4]

1 Power System Optimization Research Group, Faculty of Electrical and Electronics Engineering, Ton Duc Thang University, Ho Chi Minh City 700000, Vietnam; phamhuuly@tdtu.edu.vn

2 Department of Electrical Engineering, The University of Da Nang, University of Science and Technology, Da Nang city 550000, Vietnam; dmquan@dut.udn.vn

3 Center of Excellence for Automation and Precision Mechanical Engineering, Nguyen Tat Thanh University, Ho Chi Minh City 700000, Vietnam

4 Modeling Evolutionary Algorithms Simulation and Artificial Intelligence, Faculty of Electrical & Electronics Engineering, Ton Duc Thang University, Ho Chi Minh City 700000, Vietnam; nguyenhoangnam@tdtu.edu.vn

* Correspondence: pvduc@ntt.edu.vn (V.-D.P.); nguyentrungthang@tdtu.edu.vn (T.T.N.)

Received: 19 April 2019; Accepted: 7 May 2019; Published: 11 May 2019

Abstract: This paper proposes applications of a modified stochastic fractal search algorithm (MSFS) to solve the economic load dispatch problem (ELD) in which valve-point effects, prohibited operating zones, power losses in all conductors, multi-fuel sources and other constraints of power system are taken into consideration. The proposed method is first developed in the study by performing two modifications on two procedures of new solution generation from conventional stochastic fractal search (SFS). The first modification is used to change the strategy of producing new solutions of the first and the second update procedures while the second one is to newly update the worst solutions in the first update process and the best solutions in the second update process. These modifications have major influence on the solution search performance of the proposed method. All improvements of the proposed method can be illustrated by solving and analyzing results from various test systems with different system scales including 3-unit, 6-unit, 10-unit, and 20-unit systems. Comparison of results obtained by MSFS, SFS, and other existing methods indicates that the proposed MSFS method is more effective and robust than compared methods in terms of solution quality, high-quality solution search stability and convergence process. Consequently, the proposed method should be used as a very favorable optimization method for the ELD problem and it should be tried for other optimization problems in electrical engineering.

Keywords: modified stochastic fractal search algorithm; thermal generating units; total fuel cost; fitness function; optimal solution

1. Introduction

One of the most conspicuous trends in the 21st century is the economic growth, leading to ever-growing needs of energy consumption in all of the activities including production, services, and daily domestic households. They necessitate a power system to supply an adequate amount of energy in order to address rising power demands. However, one of the most complicated issues of power system supply is suitable electricity energy management that can use fuel cost-effectively and satisfy the system constraints exactly. The objective and constraints of such power system are taken into account in the economic load dispatch problem (ELD) [1].

Typically, the simplest cost-power feature of each thermal unit in the ELD problem is approximately represented as a quadratic function since valve-point effects and prohibited operation zones are not taken into consideration [2,3]. In practice, the thermal generating units can be supplied by one or more fuel sources, such as like coal, natural gas, and oil, and their cost function can be displayed as the piecewise quadratic function [4]. Moreover, the cost function of each unit still has complex characteristics when considering ramp rate limits, prohibited operating zones and non-convex. The above factors accompanied by power system constraints have turned the ELD problem into a complicated optimization problem.

Traditionally, the ELD problem used to be solved by using various classical optimization methods, such as the interior point method (IPM) [5], gradient method (GM) [6], lambda iteration method (LIM) [7], quadratic programming (QP) [8], Hopfield model (HM) [9], enhanced lagrangian ANN (ELANN) [10] and enhanced augmented Lagrange Hopfield network (EALHN) [11]. Among the classical approaches, ELANN and EALHN, which were based on neural network and a change in the state of neurons, were applied for solving ELD problem with the piecewise quadratic cost functions. However, these methods have coped with setbacks in handling nonlinear constraints, put up with too many numerical iterations and slow convergence. In general, the performance of conventional techniques is limited in dealing with the problems associated with multiple fuel sources, complex constraints and substantial system scale, especially real power systems. Although these shortcomings are challenges, they are also an inspiration as well as a promising research trend for many researchers to explore and exploit. To overcome these difficulties, researchers must make more efforts to find new algorithms better than previous algorithms. In recent years, a variety of techniques inspired by natural phenomena or behaviors of animals has been proposed, such as distributed auction-based algorithm (DAA) [12], artificial immune system (AISF) [13], biogeography-based optimization (BBO) [14,15], genetic algorithm (GA) [16], multi-objective evolutionary programming (MOEP) method [17], firefly algorithm (FA) [18], differential evolution (DE) [19], Krill herd algorithm (KHA) [20], Cuckoo search algorithm (CSA) [21,22]. Besides, to enhance the ability of finding global optimal solutions in large spaces, a high number of improved/modified versions of original methods have been proposed, including improved Tabu search (ITS) [23], one rank Cuckoo search algorithm (ORCSA) [24], improved Cuckoo search algorithm (ICSA) [25], anti-predatory particle swarm optimization (APSO) [26], modified PSO (MPSO) [27], improved quantum-behaved particle swarm optimization (IQPSO) [28], modified artificial bee colony algorithm (MABC) [29], iteration particle swarm optimization (IPSO) [30], modified symbiotic organism search algorithm (MSOS) [31], non-dominated sorting genetic algorithm-II (NSGA-II) [32], new adaptive particle swarm optimization (NAPSO) algorithm [33] and fuzzy logic controlled genetic algorithm (FCGA) [34].

Over a long period, many algorithms have successfully solved the ELD problem. However, they had different strengths and weaknesses. Therefore, many researchers have chosen outstanding algorithms and combined them to create new algorithms with more promising results than the original algorithms. For this purpose, many articles have productively introduced and launched highly effective algorithms for the ELD problem, such as hybrid PSO with real-valued mutation (RVM-PSO) [35], combination of the differential evolution and particle swarm optimization algorithms (DEPSO) [36], hybrid PSO with gravitational search algorithm (HPSO-GSA) [37], modified shuffled frog leaping algorithm (MSFLA) [38], global best harmony search algorithm (GHS) [38], hybrid SFLA-GHS algorithm [38], combination of hybrid SFLA-GH and shuffled differential evolution (MSFLA/GHS/SFLA-GHS/SDE) [38], hybrid fuzzy adaptive chaotic ant swarm optimization (FCASO) algorithm and sequential quadratic programming (SQP) technique (FCASO-SQP) [39], a new hybridization of particle swarm optimization with dynamic linkage discovery (PSO-RDL) [40], hybrid ant colony optimization and real-coded genetic algorithm (GAAIP) [41] and shuffled differential evolution (SED) [42]. All of the previous methods showed that they were powerful and effective techniques that could deal with most of such difficulties of classical methods.

In this paper, a novel method called modified stochastic fractal search algorithm (MSFS) for solving the ELD problem taking into consideration all constraints relating to the power system and generator features. MSFS is created by improving the performance of stochastic fractal search algorithm

(SFS), which is also a population-based meta-heuristic algorithm and was derived from the natural evolution process [43]. SFS has received much interest from researchers and it has been successfully utilized to resolve optimization problems in engineering fields so far. The main goal of this method was to create an efficient solution search strategy by using diffusion procedure and two update mechanisms. This algorithm can find near-optimal solutions and overtake the disadvantages from other meta-heuristic methods, such as premature convergence in a local optimum zone without finding the real optimal solution. The structure of SFS comprises three main stages for producing new solutions, such as the diffusion process, the first update process, and the second update process. In the first stage, each old solution is newly updated more than one time and there are at least two new solutions produced around each old solution. On the contrary, the two remaining stages produce fewer solutions dependent on solution quality in which the worst solutions have more chances to be newly updated while the best solutions have a lower possibility to be newly produced. In MSFS, we concentrate on the improvement of stage 2 and stage 3 by proposing two modifications. In the first modification, two step sizes are suggested to be used and a new technique is proposed for determining one more appropriate step size for updating new solutions. Furthermore, search spaces can be around each individual solution or around the best solution depending on the comparison between fitness function of the considered solution and the best solution. In the second modification, selected solutions to be newly updated are different from those in SFS. In SFS, the worst solutions have more chances to be newly updated rather than the best solutions in the two-update process. The strategy limits to exploit search spaces around solutions with high quality and miss good search spaces. In MSFS, the worst solutions are given priority to be updated in the first update process while the best solutions are selected to be updated in the second update process. In general, MSFS can overcome major shortcomings that SFS has coped with so far, but the implementation of the proposed MSFS method for obtaining promising results has also encountered several difficulties as follows:

(i) Set the most appropriate value to the probability of updating bad solutions. The probability is in the range from zero to one, but it should be selected by evaluating the results of the first some cases and then the best value is fixed for other cases. However, a randomly selected value in middle point, such as 0.6 or 0.5, can result in better solutions than SFS method.
(ii) Select the best values for population size and iterations. As setting the control parameter to high values, it is sure that the best performance is obtained but execution time may be long. Therefore, the best values must get two advantages, high-quality solutions and short enough simulation time.

The application of these modifications is implemented and evaluated by testing the proposed MSFS method and SFS method on four systems with twelve cases with a different number of generators and different constraints. In addition, the proposed method is also compared to other existing methods. The main contributions of the study can be summarized as follows:

(i) Describe the SFS method in detail and point out its advantages
(ii) Propose highly effective modifications on SFS in aim to reduce values of population size and iterations, shortening simulation time and finding high performance.
(iii) Show the real performance of the proposed method when solving complicated systems. This can help readers to decide if the method should be used for their applied problems.

Other remaining parts of the paper are as follows: Section 2 describes the ELD problem in detail by explaining the objective function and considered constraints. Section 3 introduces the SFS method and then proposes MSFS to the overcome shortcomings of SFS. Section 4 instructs the application of MSFS for solving the ELD problem. Section 5 shows obtained results from MSFS and compares the results with those from other methods including SFS. Section 6 concluded the work in the paper and introduces future work. In addition, Appendix A is also added for showing optimal solutions obtained by MSFS.

2. Problem Formulation

2.1. Objective Function

The main duty of the ELD problem is to determine the total fuel cost of thermal units and satisfy any constraints of the power system. The total fuel cost can be calculated by using the fuel function of each thermal generating unit that can be expressed as follows:

2.1.1. Traditional Cost Function

When the thermal unit uses single fuel cost, the objective function of ELD can be expressed as a quadratic function and formulated as follows:

$$C_k(AP_k) = v_k + x_k AP_k + y_k AP_k^2 \quad (\$/\mathrm{h}) \tag{1}$$

So, the mathematical modeling for the objective function of the ELD problem is calculated by determining the total fuel cost of N units as follows:

$$\text{Min } C = \sum_{k=1}^{N} C_k(AP_k) \tag{2}$$

where C_k is the fuel cost of thermal generating unit k and its form is as shown in Equation (2) and in Figure 1.

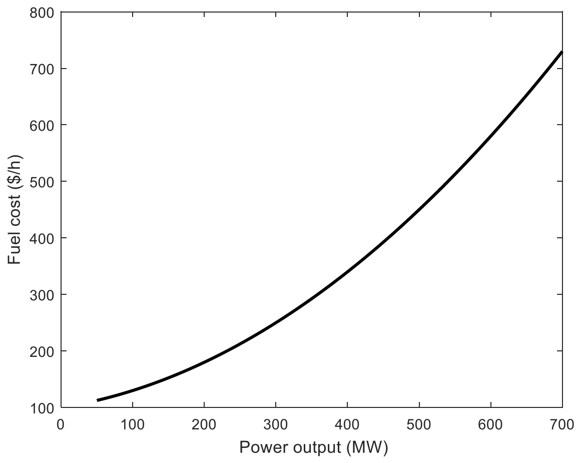

Figure 1. Fuel cost function for the case of single fuel option.

Traditionally, fuel cost function was represented as the second order form once the effects of thermal valves during the power change were not taken into account. However, it is more realistic since the valve effects are considered for the process of increasing or decreasing the power output of thermal generating units. The valve effects are represented as a sinusoidal term. Therefore, the fuel cost function is the sum of the second order Function (1) and the sinusoidal term as shown in Equation (3). The effects can be identified by observing the curve in red in Figure 2 [2].

$$C_k(AP_k) = v_k + x_k AP_k + y_k AP_k^2 + \left| s_k \times \sin\left[t_k \times \left(AP_k^{\min} - AP_k \right) \right] \right| (\$/\mathrm{h}) \tag{3}$$

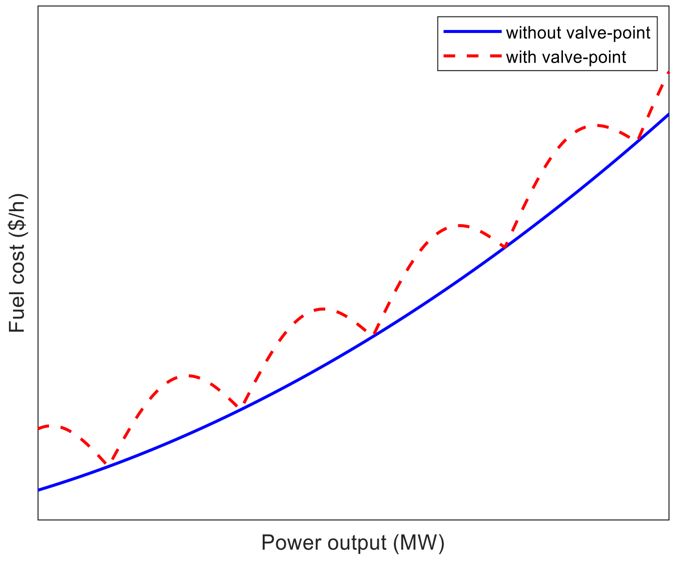

Figure 2. Fuel cost function with valve-point effects.

2.1.2. Cost Function with Multiple Fuels and Valve-Point Effects

Usually, thermal generating units can be provided with multiple fuels. So, the cost function can be piecewise curves consisting of two or three second order equations depending on the number of fuels. For this case, C_k can be described in Equation (4) and plotted in Figure 3.

$$C_k(AP_k) = \begin{cases} v_{k1} + x_{k1}AP_k + y_{k1}AP_k^2, \text{ fuel 1, } AP_k^{\min} \le AP_k \le AP_{k1}^{\max} \\ \ldots.. \\ v_{km} + x_{km}AP_k + y_{km}AP_k^2, \text{ fuel m, } AP_{km}^{\min} \le AP_k \le AP_{km}^{\max} \\ \ldots\ldots \\ v_{kMk} + x_{kMk}AP_k + y_{kMk}AP_k^2, \text{ fuel } Mk\text{, } AP_{kMk}^{\min} \le AP_k \le AP_{kMk}^{\max} \end{cases} \quad (4)$$

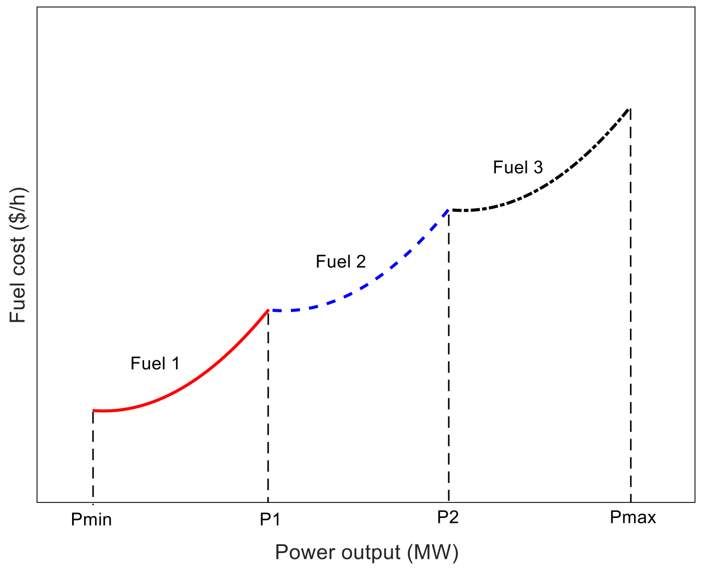

Figure 3. Fuel cost function for the case of multi-fuel options.

However, in real operation conditions, the power output of the thermal generating units is controlled by their boilers that are influenced by valve-point effects. Hence, the fuel cost function associated with multiple fuels (MF) options and valve-point effects (VPF) can be mathematically formulated as follows [28]:

$$C_k(AP_k) = \begin{cases} v_{k1} + x_{k1}AP_k + y_{k1}AP_k^2 + \left|s_{k1} \times \sin(t_{k1} \times (AP_{k1}^{\min} - AP_k))\right| \\ \text{, for fuel 1, } AP_k^{\min} \leq AP_k \leq AP_{k1}^{max} \\ \ldots\ldots \\ v_{km} + x_{km}AP_k + y_{km}AP_k^2 + \left|s_{km} \times \sin(t_{km} \times (AP_{km}^{\min} - AP_k))\right| \\ \text{, for fuel m, } AP_{km}^{\min} \leq AP_k \leq AP_{km}^{\max} \\ \ldots\ldots. \\ v_{kMk} + x_{kMk}AP_k + y_{kMk}AP_k^2 + \left|s_{kMk} \times \sin(t_{kMk} \times (AP_{kMk}^{\min} - AP_k))\right| \\ \text{, for fuel M}_\text{k}\text{, } AP_{kMk}^{\min} \leq AP_k \leq AP_k^{\max} \end{cases} \tag{5}$$

2.2. Constraints

Real power balance: The total generated power must be equal to the sum of the total load demand and transmission line losses as the following rule:

$$\sum_{k=1}^{N} AP_k = P_{LD} + P_{TLL} \tag{6}$$

where P_{LD} is the total load demand (in MW). P_{TLL} is total transmission line losses (in MW) and can be calculated by using Kron's formula below:

$$P_{TLL} = \sum_{k=1}^{N}\sum_{h=1}^{N} AP_k B_{kh} AP_h + \sum_{k=1}^{N} B_{0k} AP_k + B_{00} \tag{7}$$

where B_{kh}, B_{0k}, B_{00} are B active power loss matrix coefficients and computed by the following way [44]:

(i) Run optimal power flow for a solution to record voltage magnitude and phase angle of all buses
(ii) Determine branch currents and impedance matrix
(iii) Form Hermitian matrix H
(iv) Calculate B coefficients

The value of coefficients has a great influence on the total transmission line losses and these coefficients should be updated per iteration due to the change of solutions from optimal power flow program. However, in this study, these coefficients remain unchanged with an assumption that new generation schedule and previous schedule are slightly different or approximately similar [44].

Active power output constraint: Active power output of each generator should be between its upper and lower power limits as shown in Equation (8) below:

$$AP_k^{\min} \leq AP_k \leq AP_k^{\max} \tag{8}$$

Prohibited operation zone constraint: In some situations, the thermal units have some prohibited operation zones (POZ) owing to the survival of some weakness in generators or in accessories. These prohibited operation zones create intermittent regions in the cost function. Therefore, when the units operate, it should avoid these regions as shown in the following model:

$$\begin{array}{l} {AP_k}^{\min} \leq AP_k \leq {AP_{k,1}}^{l} \\ AP_{k,j-1}^{u} \leq AP_k \leq AP_{k,j}^{l} \ldots, j = 2, \ldots, N_k \\ AP_{k,N_k}^{u} \leq AP_k \leq AP_k^{\max} \end{array} \tag{9}$$

3. The Proposed Modified Stochastic Fractal Search Algorithm

3.1. Stochastic Fractal Search Algorithm (SFS)

SFS was first proposed by Salimi in 2014 for stacking optimization problems [43]. SFS algorithm was modified based on basic fractal search (FS) to get rid of the disadvantages of FS, such as:

1. There are too many parameters which have to be carefully selected and the selection of these parameters must be suitable to avoid trapping in local solutions.
2. The speed of convergence of the algorithm is too slow because there is no exchange of information between individuals within the group.

SFS can deal with the above drawbacks by proposing two processes: the first update and the second update. So, the structure of SFS consists of the diffusion process, the first update process and the second update process corresponding to three times of newly generated solutions. Concrete information about such SFS is presented as follow:

3.1.1. Diffusion Process

The first process is considered as a narrow zone search technique based on the natural phenomenon of growth. In such process, each solution d from the population (N_{pop}) will diffuse into the number of diffusions ($N_{d,f}$) of new solutions by using Gaussian walk model. This process is carried out by the selection of Equation (10) or (11).

$$X_{d,f}^{new} = Gaussian(X_{best}, \Delta) + \lambda \times (X_{best} - X_d) \tag{10}$$

$$X_{d,f}^{new} = Gaussian(X_d, \Delta) \tag{11}$$

where $Gaussian(X_{best}, \Delta)$ and $Gaussian(X_d, \Delta)$ are two solutions, which are, respectively, found around X_{best} and X_d by using Gaussian distribution with the standard deviation of Δ [43] in which Δ is obtained by Equation (12) below.

$$\Delta = \left| \frac{log(I_{ter})}{I_{ter}} \times (X_d - X_{best}) \right| \tag{12}$$

where I_{ter} is the current iteration. X_{best} is represented as the position of the best solution and X_d is represented as the dth solution in the population.

After using both mentioned equations to update new solutions, the quality of all updated solutions $X_{d,f}^{new}$ is measured by determining their fitness function, which is represented as $FF_{d,f}$ (f = 1,..., $N_{d,f}$). Then, Fitness function values of new solutions ($FF_{d,f}$) are compared with each other to find the lowest fitness and the solution with the lowest fitness, called FF_d^{new} and X_d^{new}, respectively [45]. Finally, the comparison between the newly updated solution' fitness function (FF_d^{new}) and that of the old solution (FF_d) is executed to keep a better one.

3.1.2. The First and Second Update Processes

The first update process is considered a large zone search technique in charge of newly producing solutions around old solutions, which were retained after the diffusion process. First of all, all previous solutions are ranked and assigned to $rank_d$ based on their fitness function in which worse solutions are with smaller order and better solutions with higher order. Namely, the best solution with the lowest fitness function is ranked last corresponding to the N_{pop}-th order while the worst one is ranked first. Similarly, other solutions are assigned to their order. In the next step, a ratio (PG_d) is calculated using Equation (13).

$$PG_d = \frac{rank_d}{N_{pop}} \tag{13}$$

In the last step, old solutions receive the decision to update. If the solution owning PG_d with a smaller value than a random number θ_d, the solution will be newly updated based on Equation (14). We noted that θ_d is a random value within the range [0, 1]. It is different from a random confidence threshold B in [46]. In [46], authors have applied the random confidence threshold B based on different distribution functions, such as beta distribution function, truncated normal distribution function and two-point distribution function to determine the critical confidence threshold *E(B)* for a continuous opinion model. The value of θ_d is used for producing a possibility of updating new solutions rather than producing new solutions with high quality.

$$X_d^{new1} = X_{t1} - rand \times (X_{t2} - X_d) \tag{14}$$

In case that PG_d is not less than θ_d, there is no update action for the considered solution. The whole description can be summarized based on the following Algorithm 1.

Algorithm 1. The first update process of SFS method

```
  (i) Sort all solutions by based on the fitness function
  (ii) Determine $rank_d$ value for all sorted solutions
  (iii) Determine $PG_d$ by using Equation (13)
For $d = 1{:}N_{pop}$
    if $PG_d < \theta_d$
        Update solution $d$ by using Equation (14)
  else
        Solution $d$ does not change
  end
end
```

In the second update process, all steps are similar to those in Algorithm 1 but the Equation (14) for updating new solutions is replaced with Equations (15) and (16) below.

$$X_d^{new2} = X_d - \alpha \times (X_{t3} - X_{best}) for \, \beta \leq 0.5 \tag{15}$$

$$X_d^{new2} = X_d + \alpha \times (X_{t3} - X_{t4}) for \, \beta > 0.5 \tag{16}$$

3.2. The Proposed Modified Stochastic Fractal Search Algorithm

3.2.1. The Newly Proposed Technique for the Two Update Procedures

In the first update procedure of SFS, new solutions via Gaussian walk are updated by using a random solution with step size (ΔX) based on a random solution and solution d as Equation (14). Obviously, the search mechanism is a random search way without a good strategy because its generated solutions always search around the random solution with the small step size. So, the first modification of MSFS aims to manage the restrictions of SFS by using a novel search strategy consisting of Equations (17) and (18).

$$X_d^{new} = X_d + \gamma \times \Delta X \tag{17}$$

$$X_d^{new} = X_{Best} + \gamma \times \Delta X \tag{18}$$

Clearly, the search strategies applying two models in Equations (17) and (18) have a big difference from Equation (14) because the proposed model of Equation (17) is used to update new solutions around the old solutions whilst the purpose of applying the model in Equation (18) is to create new solutions around the so-far best solution. The choice of applying either the model of Equation (17) or (18) should be determined by a correct criterion. The criterion is the result of the comparison between EF_d and EF_{aver}. The two new definitions are shown in Equations (19) and (20) as below:

$$EF_d = \frac{FF_d - FF_{best}}{FF_{best}} \tag{19}$$

$$EF_{aver} = \frac{\left[\left(\sum_{d=1}^{Npop} FF_d \right) / N_{Pop} \right] - FF_{best}}{FF_{best}} \tag{20}$$

where FF_d is the fitness function value of each solution X_d, FF_{best} is the fitness function value of the current best solution X_{best}.

For the case that EF_d is bigger than EF_{aver}, it implies that the current solution is far away from the so-far best solution, new solutions are updated by utilizing Equation (18). On the contrary, it can be understood that the current solution is close to the so-far best solution. Thus, the newly generated solutions are produced by employing Equation (17).

Moreover, the step size has also played a major part in finding a good solution quality. As seen, the step size in Equation (14) is always small. Thus, it may be not useful for searching optimal solutions and time consuming for being convergent to the optimal global solutions. To escape the local search zone and handle the mentioned disadvantages, we propound two step size types in order to extend the practicable search zone and overcome the local minima. The details of selecting the step size are presented as follows:

$$\Delta X = \begin{cases} (X_{t1} - X_{t2}) & if\ \varepsilon_d \langle D \\ (X_{t1} - X_{t2} + X_{t3} - X_{t4}) else \end{cases} \tag{21}$$

where D is the probability of selecting step size types, which can be from 0.1 to 1. By the experience, D should be set by 0.1 to 0.3. This indicates that the step size based on two random points is only used up to thirty percent, while another step size called the large step size can be used up to seventy percent. Consequently, the contribution of the large step size is more significant than that of another.

The proposed update process can be easily understood as observing Figure 4. In the Figure, we suppose that there are five solutions in the population. Solution 1 and solution 2 in blue together with the best solution in red are in the group with EF_d not higher than EF_{aver} while solution 3 and solution 4 in black are in another group with EF_d higher than EF_{aver}. There are two main differences in the process of updating new solutions. Search space around good solution group is exploited while that around bad solution group is neglected. Instead, the best solution is a central solution that is used to find new solutions for bad solutions. In fact, two arrows pointing to the best solution from solutions 3 and 4 illustrates the statement. It is clearly seen that spaces around solution 1, solution 2 and the best solution are marked and sought many times while spaces around solution 3 and solution 4 are abandoned. That is the first difference between good solution group and bad solution group. In the second difference, it can be identified that search spaces around solution 1 are more narrow than those around solution 2 because ε_1 is smaller than random number D but ε_2 is equal to or higher than D. Similarly, space around the best solution is plotted by two circles with different radius, one circle with a higher radius and another with a smaller radius. The smaller circle is the search space for solution 3 because ε_3 is smaller than D whereas the large circle is the search space for solution 4 because D is not higher than ε_4. The implementation of the proposed update process can be accomplished by using Algorithm 2 below.

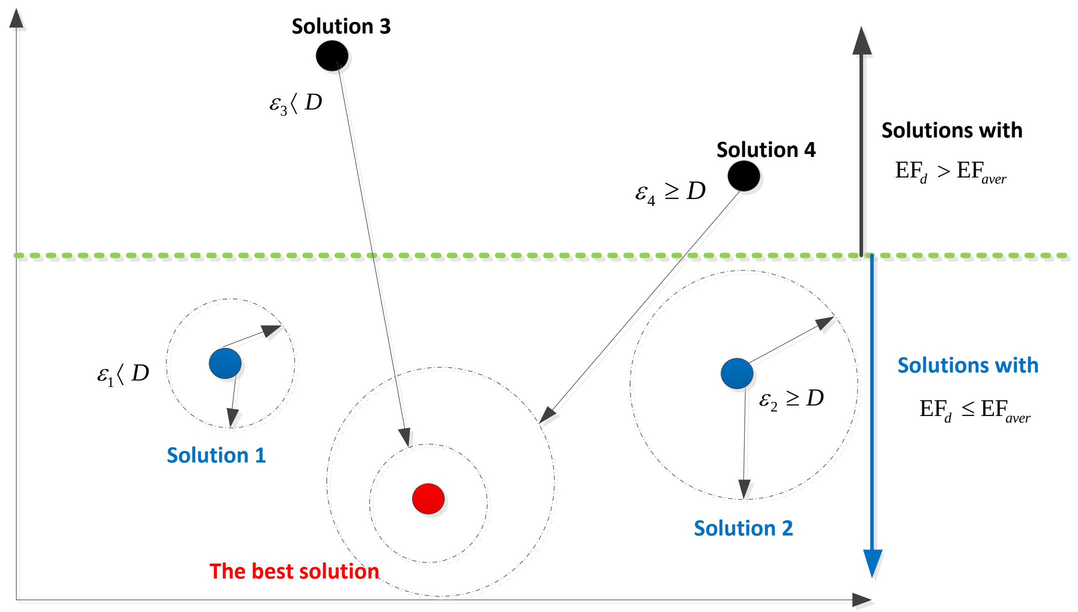

Figure 4. The description of the first update process of the proposed modified stochastic fractal search algorithm (MSFS) method.

Algorithm 2. The proposed search strategy for the first update process

```
Calculate given probability of solution $d$ by utilizing Equation (13)
    if $\varepsilon_d < D$
        $\Delta X = (X_{t1} - X_{t2})$
    else
        $\Delta X = (X_{t1} - X_{t2} + X_{t3} - X_{t4})$
    end
    if $EF_d > EF_{aver}$
        $X_d^{new} = X_d + rand \times \Delta X$
    else
        $X_d^{new} = X_{Best} + rand \times \Delta X$
    end
```

3.2.2. The Modifications on Selected Solutions for the Two Update Procedures

Equation (13) of SFS indicates that most of the solutions with bad fitness are always updated while solutions with better fitness have a lower possibility to be newly updated. Clearly, PG_d of the best solution is the largest value while that of the worst solution is the smallest value. Besides, the value of a random number of solution d, θ_d is always smaller than one and bigger than zero. Therefore, considering the comparison of PG_d and θ_d, only search spaces around the worst solutions are highly exploited while search spaces around the best solutions are disregarded. The manner has caused the main disadvantage for SFS in seeking solutions in local optimal zones or nearby global optimal solution. Therefore, in the modification, we suggest updating bad quality solutions in the first update procedure while good quality solutions are focused to be newly produced. For implementing the modification, the ratio of updating bad quality solutions R_{nbs} (where R_{nbs} is less than one and higher than zero) must be selected in advance and then the number of bad quality solutions is determined and updated in the first update process. Consequently, in the first update, $(R_{nbs} \times N_{pop})$ worst solutions are newly updated. After the first update, all solutions are ranked and then $[(1 - R_{nbs}) \times N_{pop}]$ best solutions are newly updated in the second update process.

The whole search process of the proposed MSFS method is described in detail in Figure 5 below.

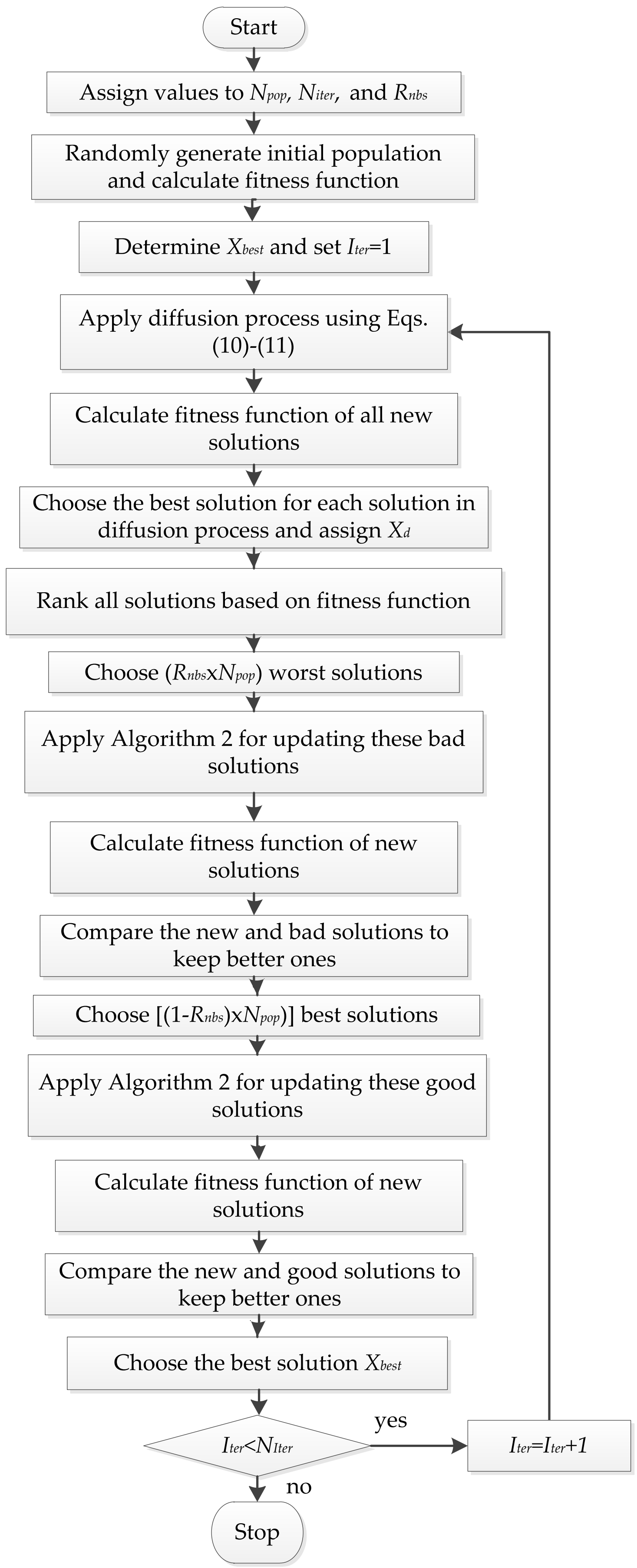
Start
Assign values to N_{pop}, N_{iter}, and R_{nbs}
Randomly generate initial population and calculate fitness function
Determine X_{best} and set I_{ter}=1
Apply diffusion process using Eqs. (10)-(11)
Calculate fitness function of all new solutions
Choose the best solution for each solution in diffusion process and assign X_d
Rank all solutions based on fitness function
Choose (R_{nbs}xN_{pop}) worst solutions
Apply Algorithm 2 for updating these bad solutions
Calculate fitness function of new solutions
Compare the new and bad solutions to keep better ones
Choose [(1-R_{nbs})xN_{pop})] best solutions
Apply Algorithm 2 for updating these good solutions
Calculate fitness function of new solutions
Compare the new and good solutions to keep better ones
Choose the best solution X_{best}
I_{ter}<N_{Iter}
yes
I_{ter}=I_{ter}+1
no
Stop

Figure 5. The flowchart of using the proposed method for solving a typical optimization problem.

4. The Implementation of MSFS for ELD Problem

4.1. Dealing with the Constraints

In ELD problem, solving load demand-supply balance constraint is one of the most important affairs because it affects the good solution quality and the speed of convergence. So, many methods have used two variable types to deal with the constraints. The first variable type is called a dependent variable and selected from one thermal generating unit. The second variable type is called a control variable and selected from other thermal generating units. In the search strategy of the proposed method, the variables are included in the position of each point in the initialization step and are updated in each iteration. Consequently, the position of point *d* will include thermal generating unit from unit 2 to unit *N* as the following formula.

$$X_d = [AP_{2,d}, AP_{3,d}, \ldots ., AP_{N,d}]; d = 1, \ldots, N_{pop} \tag{22}$$

where

$$X_{\min} \le X_d \le X_{\max} \tag{23}$$

$$X_{min} = [AP_{2,\min}, AP_{3,\min}, \ldots ., AP_{N,min}] \tag{24}$$

$$X_{max} = [AP_{2,\max}, AP_{3,\max}, \ldots ., AP_{N,max}] \tag{25}$$

As a result, the balance constraint can be solved by the expressions below.

$$AP_{1,d} = P_{LD} + P_{TLL} - \sum_{k=2}^{N} AP_{k,d} \tag{26}$$

4.2. Handling the Violation of the Dependent Variable

The value of $AP_{1,d}$ obtained from Equation (26) may violate constraint in Equation (8). Therefore, the violation must be evaluated in the quality of solutions. The task is performed by computing the following penalty term.

$$PEN_{AP_1} = \begin{cases} \left(AP_{1,d} - AP_1^{\max}\right) \ if \ \left(AP_{1,d} > AP_1^{\max}\right) \\ \left(AP_1^{\min} - AP_{1,d}\right) \ if \ \left(AP_{1,d} < AP_1^{\min}\right) \\ 0 \qquad else \end{cases} \tag{27}$$

As seen from Equation (27), the thermal generating unit 1 will be penalized if it is higher than upper bound or smaller than lower bound. On the contrary, it will not be penalized or the penalty term will be zero if it is within the upper bound and the lower bound.

4.3. Handling the Violation of Upper and Lower Boundaries

After newly updating the power output of from the second unit to the last unit, these units must be checked and corrected exactly by using the following equation:

$$AP_k = \begin{cases} AP_k^{\max} \ if \ AP_k > AP_k^{\max} \\ AP_k^{\min} \ if \ AP_k < AP_k^{\min} \\ 0 \qquad else \end{cases} ; k = 2, \ldots, N \tag{28}$$

4.4. Handling the Violation of Prohibited Operation Zones

If the active power output of the *k*th thermal generator falls into one out of prohibited operation zones, it should not be kept but correction and penalty need to be made depending on thermal generators. From the second generator to the last generator (AP_k with $k = 2, \ldots, N$) should be corrected by using Equation (29) while the first unit AP_1 should not be corrected but it needs to be penalized by using Equation (30). The two equations are as follows:

$$AP_k = \begin{cases} AP^l_{k,j} \ if \ AP_k \in \left[AP^l_{k,j}, AP^u_{k,j}\right] \& \left(AP_k < AP^{aver}_{k,j}\right) \\ AP^u_{k,j} \ if \ AP_k \in \left[AP^l_{k,j}, AP^u_{k,j}\right] \& \left(AP_k \geq AP^{aver}_{k,j}\right) \\ AP_k \ \ if \ AP_k \notin \left[AP^l_{k,j}, AP^u_{k,j}\right] \end{cases} ; k = 2, \ldots, N \,\&\, j = 1, \ldots, N_k \tag{29}$$

$$PEN_{POZ,AP_1} = \begin{cases} 0 \ \ if \ AP_1 \notin \left[AP^l_{1,j}, AP^u_{1,j}\right] \\ \left|AP_1 - AP^l_{1,j}\right| \ if \ AP_1 \notin \left[AP^l_{1,j}, AP^u_{1,j}\right] \& \left(AP_1 < AP^{aver}_{1,j}\right) \\ \left|AP_1 - AP^u_{1,j}\right| \ if \ AP_1 \notin \left[AP^l_{1,j}, AP^u_{1,j}\right] \& \left(AP_1 > AP^{aver}_{1,j}\right) \end{cases} ; j = 1, \ldots, N_1 \tag{30}$$

In Equations (29) and (30), average powers are defined as the following model:

$$AP^{aver}_{k,j} = \frac{AP^u_{k,j} + AP^l_{k,j}}{2}; \ j = 1, \ldots, N_k; \ k = 1, \ldots, N \tag{31}$$

4.5. Handling the Violation of the Spinning Reserve Power (SRP)

Under normal conditions, the operating reserve capacity of the generating thermal units has played an important role in meeting load demands within a short time in case a generator breaks or there is another disruption to the supply. If that is not satisfactory, the approach is penalized. This problem is presented as the following formula:

$$PEN_{SRP} = \begin{cases} 0 \qquad if \ \sum_{k=1}^{N} \left(AP^{\max}_k - AP_k\right) \ \geq \ AP_{SRP} \\ \left|\sum_{k=1}^{N} \left(AP^{\max}_k - AP_k\right) \ - \ AP_{SRP}\right| esle \end{cases} \tag{32}$$

4.6. Calculating Fitness Function

Most of the meta-heuristic methods need to calculate the fitness function of all the solutions in order to arrange the rank of all the solutions. Accordingly, the fitness function is determined as the following equation.

$$FF_d = \sum_{k=1}^{N} C_k(AP_k) + K \times (PEN_{AP_1})^2 + K \times (PEN_{POZ,AP_1})^2 + K \times (PEN_{SRP})^2 \tag{33}$$

4.7. The Detail of MSFS's Procedure for ELD Problem

Step 1. Select parameters for the proposed MSFS, such as the number of points N_{pop}, the maximum value of iterations N_{Iter}, and the ratio of updating bad quality solutions R_{nbs}

Step 2. Randomly initialize population based on the rule shown in Equation (23)

- Handling POZ constraint for unit 2 to unit N based on (29)
- Calculate dependent variables $AP_{1,d}$ by using (26)
- Penalize the violation of generation limits for $AP_{1,d}$ by using Equation (27)
- Penalize the violation of POZ constraint for $AP_{1,d}$ by using Equation (30).
- Calculate fitness function FF_d for the solution d by using (33)
- Find the best solution X_{Best} with the lowest fitness
- Start the first iteration (I_{ter} = 1)

Step 3. Produce new solutions of the diffusion process using Equations (10) and (11)
Step 4. Correct new solutions by using Equation (28)

- Handling POZ constraint for unit 2 to unit N based on (29)
- Calculate dependent variables $AP_{1,d}$ by using (26)
- Penalize the violation of generation limits for $AP_{1,d}$ by using Equation (27).
- Penalize the violation of POZ constraint for $AP_{1,d}$ by using Equation (30).
- Calculate fitness function FF_d for the solution d by using (33)

Step 5. Compare new solutions and old solutions to retain better ones.
Step 6. Update bad solutions by using the first update process in Algorithm 2

- Correct new solutions by using Equation (28)

Step 7. Ha$n$dling POZ constraint for unit 2 to unit N based on (29)

- Calculate dependent variables $AP_{1,d}$ by using (26)
- Penalize the violation of generation limits for $AP_{1,d}$ by using Equation (27).
- Penalize the violation of POZ constraint for $AP_{1,d}$ by using Equation (30).
- Calculate fitness function FF_d for the solution d by using (33)

Step 8. Compare new solutions and old solutions to retain better ones.
Step 9. Update good solutions by using the second update process in Algorithm 2

- Correct new solutions by using Equation (28)

Step 10. Handling POZ constraint for unit 2 to unit N based on (29)

- Calculate dependent variables $AP_{1,d}$ by using (26)
- Penalize the violation of generation limits for $AP_{1,d}$ by using Equation (27).
- Penalize the violation of POZ constraint for $AP_{1,d}$ by using Equation (30).
- Calculate fitness function FF_d for the solution d by using (33)

Step 11. Compare new solutions and old solutions to retain better ones.
Step 12. Determine the best solution with the smallest fitness function value.

5. Numerical Results

In this section, the proposed MSFS and SFS are applied to solve the ELD problem in four different test systems with 3, 6, 10 and 20 thermal units. The first test system considers two cases (cases 1–2), the second system evaluates four cases (cases 3–6), the third system analyzes five cases (cases 7–11), and the final system investigates only one case (case 12). The different cases are due to the different fuel cost function characteristics, different loads and different constraints in which fuel cost functions consider single fuel with and without valve-point effects (VPE) and multi-fuels (MF) with and without valve-point effects while different constraints are power losses (P_{TLL}), prohibited operating zone (POZ) and power balance. As a result, there are twelve study cases with a detailed description shown in Table 1.

Similar to the meta-heuristic algorithms, the proposed MSFS also needs to set some parameters, such as the population size, the maximum iteration number, the number of diffusions and the ratio of updating bad-quality solutions reported in Table 2. All study cases of the method have been performed in Matlab program language and a computer with 4 GB of Ram and 2.4 Ghz processor.

Table 1. Summary of twelve study cases.

Case	Unit	P_{LD} (MW)	POZ	P_{TLL} (MW)	VPE	MF
Case 1	3-unit	850	No	Yes	No	No
Case 2	3-unit	850	No	No	Yes	No
Case 3	6-unit	800	No	No	No	No
Case 4	6-unit	1200	No	No	No	No
Case 5	6-unit	1800	No	No	No	No
Case 6	6-unit	1263	Yes	Yes	No	No
Case 7	10-unit	2400	No	No	No	Yes
Case 8	10-unit	2500	No	No	No	Yes
Case 9	10-unit	2600	No	No	No	Yes
Case 10	10-unit	2700	No	No	No	Yes
Case 11	10-unit	2700	No	No	Yes	Yes
Case 12	20-unit	2500	No	Yes	No	No

Table 2. The selection of population size and the highest iteration number.

Case	N_{pop}	N_{Iter}	$N_{d,f}$	R_{nbs}
Case 1	3	15	2	0.6
Case 2	10	10	2	0.6
Cases 3–6	5	20	2	0.6
Cases 7–10	10	40	2	0.6
Case 11	30	100	2	0.6
Case 12	10	100	2	0.6

5.1. The Analysis of the Performance Improvement of MSFS

In order to test the level of improvement of MSFS compared to SFS, this section surveys the effect of population and the number of iterations on the results of the methods. The first survey is studied in a complex constraints case, specifically, case 6 with transmission power losses and prohibited operating zones as constraints. The results achieved by SFS and MSFS methods with respect to the best cost, mean cost, worst cost and standard deviation cost are shown in Table 3 and Figure 6. Among the four cost values, the best cost and standard deviation cost are the two most important values considered as major comparison criteria. It means that the comparison of the minimum cost is used to evaluate the best optimal solution and the comparison of the standard deviation cost is used to evaluate the stabilization of finding ability. From Table 3, the population of the two methods is set to five but the number of iterations is changed from 15 to 60 iterations. As observed from the minimum cost, it only needs 20 iterations for MSFS to get the best cost of 15,443.0752 ($/h). Meanwhile, the best cost of SFS gradually decreases from 15,443.1775 ($/h) to 15,443.0752 ($/h) when the number of iterations of SFS is changed from 15 to 60. Specifically, a cost of 15,443.1775 ($/h) is corresponding to the setting of 15 iterations, a cost of 15,443.1381 ($/h) is corresponding to the setting of 20 iterations, a cost of 15,443.1197 ($/h) is corresponding to the setting of 30 iterations, a cost of 15,443.1171 ($/h) is corresponding to the setting of 45 iterations, and, a cost of 15,443.0752 ($/h) is corresponding to the setting of 60 iterations. Clearly, to obtain the minimum cost of 15,443.0752 ($/h), SFS must use 60 iterations but MFSF only uses 20 iterations. As seen Figure 6, the minimum cost of MSFS is 15,443.0772 ($/h) at 15th iteration, but from the 20th iteration to the 60th iteration, cost of MSFS is also 15,443.0752 ($/h) and does not change. On the contrary, the best cost of SFS tends to be decreased from 15,443.1755 ($/h) to 15,443.0752 ($/h) corresponding to from the 15th iteration to the 60th iteration. This implies that MSF is more efficient than SFS. Besides, MSFS always has a lower standard deviation cost than SFS when the two methods use the same number of iterations. This points out that MSF is more robust than SFS.

Table 3. Results obtained by MSFS and stochastic fractal search (SFS) methods for case 6.

N_{Iter}	Method	Min. Cost ($/h)	Aver. Cost ($/h)	Max Cost ($/h)	Std dev. ($/h)
15	MSFS	15,443.0772	15,450.0960	15,484.6862	0.2508
	SFS	15,443.1755	15,470.3311	15,619.4444	3.9302
20	MSFS	15,443.0752	15,454.5582	15,600.7939	0.7644
	SFS	15,443.1381	15,457.2901	15,490.2090	1.2845
30	MSFS	15,443.0752	15,450.7556	15,483.2619	0.3138
	SFS	15,443.1197	15,459.8633	15,593.5810	2.5499
45	MSFS	15,443.0752	15,450.5985	15,476.4993	0.3194
	SFS	15,443.1171	15,462.6925	15,601.3773	2.6220
60	MSFS	15,443.0752	15,447.3049	15,471.5278	0.2170
	SFS	15,443.0752	15,460.7564	15,549.6672	2.1338

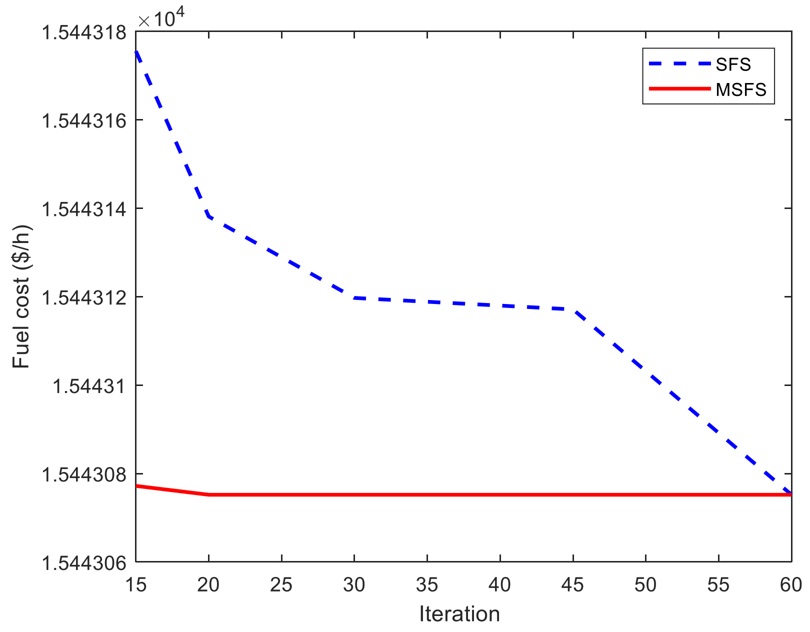

Figure 6. Fuel cost obtained by SFS and MSFS with different values of N_{Iter} for case 6.

For other remaining cases, the strong point of the proposed method over SFS is the same manner. The obtained results of MSFS and SFS in regard to the best cost, the average cost, the worst cost and the standard deviation are seen in Table 4 for the rest of the cases. With the same population and iterations, the best cost of MSFS is always smaller than those of SFS. In addition, to see the improvement of MSFS over SFS, we have calculated the reduction cost and added it as the last column in Table 4. For different cases or different test systems, the value of reduction is also different. For example, that of the first test system is from 0.02 ($/h) to 0.04 ($/h), for the second test system is from 0.019 ($/h) to 0.409 ($/h), for the third test system is from 0.0001 ($/h) to 0.0429 ($/h), and for the last test system is 0.006 ($/h). In general, the reduction cost is not high for each hour but the reduction for one day with 24 h or one year with 8760 h is highly significant. Furthermore, the improvement of the proposed method over SFS can be identified more clearly as focusing on average cost, maximum cost as well as standard deviation cost.

To give further evidence of the stability of the proposed algorithm, we also test the distribution of the minimum costs over fifty independent trial runs for case 6 and case 11. Case 6 takes transmission power losses and prohibited operating zones into account and case 11 considers multi-fuels options and valve-point effects as constraints. The best costs of 50 runs obtained by MSFS and SFS are plotted in Figures 7 and 8 for case 6 and case 11, respectively. These figures show a random distribution of

50 values of fitness function values obtained by MSFS and SFS over 50 fruitfully independent runs. It means that after each trial run, there is a red point of MSFS and a blue point of SFS allocated on such curves. These points can be increased or decreased without any rules. The deviation between the peak of SFS and MSFS is very high. The fluctuations of SFS are high since all points of MSFS are approximately lied on a line. Furthermore, it can point out that nearly all points of MSFS have the same cost as the best point but only a few points of SFS are around the best point of MSFS. The manner indicates that SFS hardly ever finds an optimal solution.

Table 4. Results obtained by MSFS and SFS methods for the rest of the cases.

Case	Method	Min. Cost ($/h)	Aver. Cost ($/h)	Max. Cost ($/h)	Std. Dev ($/h)	Reduction Cost ($/h)
Case 1	SFS	8344.613	8354.989	8484.935	10.4561	0.02
	MSFS	8344.593	8347.9508	8364.731	5.2664	
Case 2	SFS	8234.112	8240.3524	8252.073	6.0824	0.04
	MSFS	8234.072	8240.7777	8251.061	4.0656	
Case 3	SFS	8227.506	8278.6951	8448.024	5.1367	0.409
	MSFS	8227.097	8244.5885	8439.616	1.486	
Case 4	SFS	11,477.109	11,515.151	11,656.08	4.5367	0.019
	MSFS	11,477.09	11,495.095	11,560.31	0.7406	
Case 5	SFS	16,579.35	16,588.184	16,644.100	1.1320	0.02
	MSFS	16,579.33	16,591.024	16,716.37	0.6577	
Case 7	SFS	481.7240	484.2762	537.0084	6.5127	0.0014
	MSFS	481.7226	483.196	526.4292	3.7283	
Case 8	SFS	526.2389	530.1006	565.0843	7.1695	0.0001
	MSFS	526.2388	527.6925	556.1475	3.3526	
Case 9	SFS	574.3817	578.1708	608.8124	6.9787	0.0009
	MSFS	574.3808	575.2905	589.5618	2.1429	
Case 10	SFS	623.8126	627.9128	662.3985	6.0800	0.0034
	MSFS	623.8092	624.8894	645.5859	2.0321	
Case 11	SFS	623.8764	627.4806	655.5396	6.1576	0.0429
	MSFS	623.8335	623.9309	626.3913	0.0415	
Case 12	SFS	62,456.639	62,460.654	62,482.88	6.3851	0.006
	MSFS	62,456.633	62,456.676	62,457.94	0.19766	

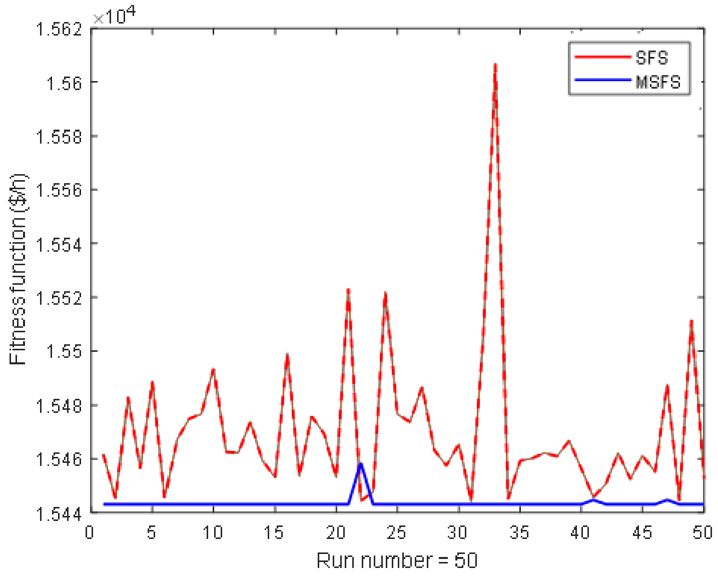

Figure 7. Distribution of 50 runs obtained by SFS and MSFS for case 6.

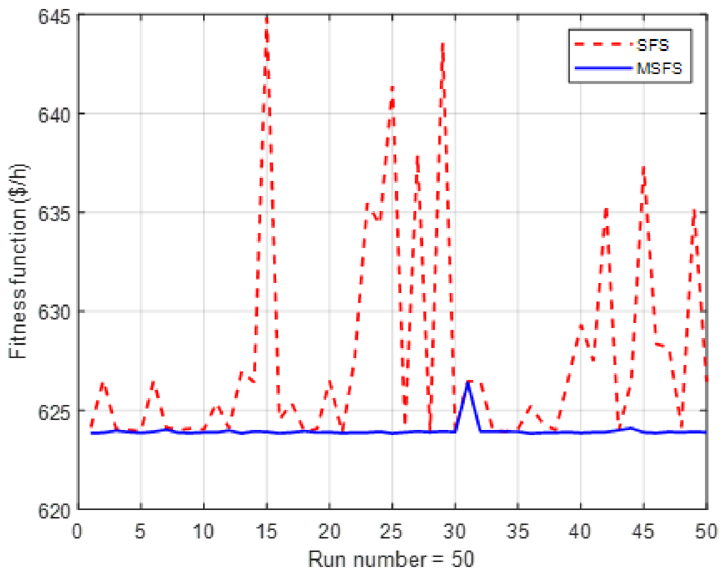

Figure 8. Distribution of 50 runs obtained by SFS and MSFS for case 11.

5.2. Comparison and Discussion

In this part, we will conduct an investigation of the best simulation results of the proposed algorithm and those from other approaches available in the paper. Besides, another comparison formula is concerned to be the number of fitness assessments (N_{Ger}), which is obtained by Equation (34) below:

$$N_{Ger} = t \times N_{pop} \times N_{Iter} \tag{34}$$

where t stands for the number of generations in every iteration. Depending on the structure of optimization algorithms, they have a difference of the number of generations. For example, CSA and ORCSA technique with two solution generations, t is two while t is one for other technique with one solution generation, such as DE, GA, and PSO methods. For the proposed MSFS, the number of generations in each iteration equals the sum of the number of diffusions and number one ($N_{d,f} + 1$). This formula indicates that an algorithm with lower N_{Ger} is assessed to be a more effective method if it has equal or lower best cost.

5.2.1. Test System 1 with 3 Units

In the first test system, a small size system with three generators is dissected with two different study cases in which transmission power losses are investigated in case 1 and valve-point effects are investigated in case 2. The load demand in test system 1 is set to 850 MW. The detailed data input of the first case related to the fuel cost function coefficient and transmission power losses matrix are shown in [17] and [32]. In this case, we discuss the results of the proposed method and others such as MOEP [17], NSGA-II [32], ICSA [25], CSA [22], and BBO [15] regarding the best cost, the population (N_{pop}), the number of iterations (N_{Iter}), and the number of fitness evaluations (N_{Ger}) in Table 5. As seen in Table 5, the solution quality of the MSFS method is the same as that of ICSA [25], CSA [22], and BBO [15] but better than that of MOEP [17] and NSGA-II [32]. Besides, N_{Ger} of MOEP, NSGA-II and BBO is very high, 150,000 for MOEP, 10,000,000 for NSGA-II and 6000 for BBO while N_{Ger} of MSFS is only 135. For that reason, MSFS is capable of searching very good quality solutions for case 1.

Case 2 considers valve-point effects and its input data is taken from Reference [39]. Table 6 shows a comparison between MSFS and other ones. As shown in Table 6, all of the methods can solve the case with the same best cost. However, MSFS has only used 300 evaluations while most of the other methods have used many times of fitness evaluation. N_{Ger} of SDE [42] cannot be calculated because it has not reported population and iterations.

Table 5. Result comparisons for case 1.

Method	Min. Cost ($/h)	N_{pop}	N_{Iter}	N_{Ger}
BBO [15]	8344.59	30	200	6000
MOEP [17]	8344.60	500	300	150,000
CSA [22]	8344.59	-	-	-
ICSA [25]	8344.59	-	-	-
NSGA-II [32]	8344.60	500	20,000	10,000,000
MSFS	8344.59	3	15	135

Table 6. Comparison of the implemented results for case 2.

Method	Min. Cost ($/h)	N_{pop}	N_{Iter}	N_{Ger}
FA [18]	8234.07	25	100	5000
ITS [23]	8234.07	20	50	1000
MPSO [27]	8234.07	20	100	2000
FCASO-SQP [39]	8234.07	20	200	12,000
PSO-RDL [40]	8234.07	20	50	1000
SDE [42]	8234.07	-	-	-
MSFS	8234.07	10	10	300

The optimal solutions of the two cases are shown in Table A1 in Appendix A.

5.2.2. Test System 2 with 6 Units

The second test system is established by six thermal units and is divided into four cases from case 3 to case 6. In case 3, case 4, and case 5, the fuel cost function of thermal units is a quadratic function, the data implemented in this situation are obtained from [34] and the required load demand is set to be 800, 1200, and 1800 MW without transmission losses. For case 6, both the prohibited operating zones and transmission losses are considered while the active load demand is set to 1263 MW [33].

Table 7 provides the results yielded by MSFS and other approaches. With regard to the best cost, MSFS gets a cost of 8227.09 ($/h) for case 3, 11,477.09 ($/h) for case 4 and 16,579.33 ($/h) for case 5. These results absolutely suppress FCGA [34] and CGA [34] but share the same position with GA [16], CSA [22], and ICSA [25] for cases 3, 4 and 5.

Table 7. Comparison of the implemented results for case 3, case 4, and case 5.

Method	Case 3	Case 4	Case 5	N_{Ger}
	Min. Cost ($/h)			
GA [16]	8227.09	11,477.09	16,579.33	-
CSA [22]	8227.10	11,477.09	16,579.33	-
ICSA [25]	8227.10	11,477.09	16,579.33	-
FCGA [34]	8231.03	11,480.03	16,585.85	10,000
CGA [34]	8232.89	11,493.74	16,589.05	10,000
MSFS	8227.09	11,477.09	16,579.33	300

Table 8 layouts the minimum cost obtained by seven approaches. From Table 8, three out of seven methods obtained the same best cost value equal to 15,443.08$. They are KHA [20], CSA [21], and MSFS, respectively. This result is considerably lower than those from BBO [14], NAPSO [33], IPSO [30], and GAAPI [41] by 0.02 ($/h), 0.69 ($/h), 0.92 ($/h), and 6.62 ($/h), respectively. Furthermore, the number of fitness evaluations of MSFS is only 300 for cases 3, 4, 5, and 6 and is smaller than that of the others. Clearly, MSFS is superior to these compared methods.

Table 8. Comparison of the implemented results for case 6.

Method	Min. Cost ($/h)	N_{pop}	N_{Iter}	N_{Ger}
BBO [14]	15,443.1	50	100	5000
KHA [20]	15,443.08	50	100	5000
CSA [21]	15,443.08	50	300	30,000
IPSO [30]	15,444	20	200	4000
NAPSO [33]	15,443.77	5	100	3000
GAAPI [41]	15,449.7	-	-	1000
MSFS	15,443.08	5	20	300

Optimal solutions of the systems are shown in Table A2 in Appendix A.

5.2.3. Test System 3 with 10 Units

In this part, a medium scale power system with 10 generators is employed for five cases from case 7 to case 11. Among these five cases, only case 11 takes valve-point loading effects into account but all the cases use the piecewise quadratic function [11]. Four levels of the load demand are examined, which are found in the literature: P_{DL} = 2400 MW (in case 7), P_{DL} = 2500 MW (in case 8), P_{DL} = 2600 MW (in case 9), and P_{DL} = 2700 MW (in case 10). The value of the best cost and the fitness evaluations obtained by the proposed algorithm and ten other methods are listed in Table 9. This table reveals that the best cost of MSFS has equally optimal solution quality with eight compared methods and lower cost than two other ones, such as ELANN [10] and IQPSO [28]. However, MSFS is ranked at the first position amongst these methods because of having the lowest value of N_{Ger} with 1.200 while AIS [12] and MPSO [27] have used 3000, DE [5] has used 12.000, IQPSO [28] has used 40.000, and MSFLA, GHS, SFLA-GHS, and SDE [38] have used 9.000. Clearly, MSFS is the standout method with the lowest evaluation for these four cases.

Table 9. Comparison of minimum cost ($/h) for cases 7, 8, 9 and 10.

Method	Case 7	Case 8	Case 9	Case 10	N_{Ger}
ELAHN [10]	481.740	526.270	574.410	623.880	-
EALHN [11]	481.723	526.239	574.381	623.809	-
AIS [13]	481.723	526.240	574.381	623.809	3000
DE [19]	481.723	526.239	574.381	623.809	12,000
MPSO [27]	481.723	526.239	574.381	623.809	3000
IQPSO [28]	481.732	526.245	574.387	623.832	40,000
SDE [38]	481.723	526.239	574.381	623.809	9000
MSFLA [38]	481.723	526.239	574.381	626.254	9000
GHS [38]	481.723	526.239	574.381	623.809	9000
SFLA-GHS [38]	481.723	526.239	574.381	623.809	9000
MSFS	481.723	526.239	574.381	623.809	1200

Another case (case 11) in such test system is also investigated in order to verify the efficiency of the proposed method where both valve-point loading effects and multi-fuels options are assessed. The complete data are given in [35]. The best cost, the population, the number of iterations, and the number of fitness evaluations achieved by eleven methods are arranged in Table 10. As Table 10 shows, the best cost of BBO is only equal to 605.639 ($/h) and is lower than those of other methods. Nevertheless, it is difficult to determine whether the result is accurate. Referring to BBO [14], we see that the best solution of generator 3 is 332.02 MW. It is clear that this value is corresponding to the second fuel type but the authors [14] have used fuel cost function coefficients of the third fuel type for calculating the fuel cost of generator 3. This mistake has resulted in a smaller fuel cost than the accurate value. Considering the solution quality aspect, the minimum cost of MSFS obtains 623.834 ($/h), is lower than those from DAA [12], APSO [36], RVM-PSO [35], MSFLA [38], GHS [38], and SFLA-GHS [38] and slightly higher than those from three methods, such as IQPSO [32], DEPSO [36], and SDE [38]. However, N_{ger} of MSFS has used 9.000 while IQPSO [32] has used 40.000, and DEPSO [36] has used 25.000. Clearly, N_{ger} of MSFS is many times lower than that of IQPSO [32], DEPSO [36]. This shows that MSFS outperforms the other methods for case 11.

Table 10. Comparison of the implemented results for case 7.

Method	Min. Cost ($/h)	N_{pop}	N_{Iter}	N_{Ger}
DAA [12]	623.952	-	-	-
BBO [14]	605.639	50	400	20,000
APSO [26]	624.015	20	200	4000
IQPSO [28]	623.832	80	500	40,000
RVM-PSO [35]	623.959	30	300	9000
DEPSO [36]	623.830	50	500	25,000
MSFLA [38]	624.116	60	150	9000
GHS [38]	623.849	60	150	9000
SFLA-GHS [38]	623.841	60	150	9000
SDE [38]	623.827	60	150	9000
MSFS	623.834	30	100	9000

The optimal solutions of the system are shown in Table A3 in Appendix A.

5.2.4. Test System 4 with 20 Units

In this paragraph, the proposed method is applied to deal with the ELD for a twenty-unit system with the transmission line losses [9]. The performance of MSFS and a number of other approaches is tabulated in Table 11. The most interesting finding in Table 11, a cost of 62,455.616 ($/h) for HPSO-GSA algorithm is very diminutive but this result is unachievable. Consulting HPSO-GSA [37], we see that the results of the total generated real powers and system power losses do not satisfy the power balance constraints Equation (6). Standing on this view, the cost of MSFS, CSA [25], and ORCSA [25] is only 62,456,633 ($/h) and the value is lower than those of lambda-iteration [9], HM [9], and BBO [31]. Moreover, N_{Ger} of MSFS is lower than those of CSA [25], ORCSA [25] and BBO [31]. Consequently, MSFS is one of the most effective methods with the lowest fitness evaluation.

Table 11. Comparison of the implemented results for case 12.

Methods	Min. Cost ($/h)	N_{pop}	N_{Iter}	N_{Ger}
Lambda-Iteration [9]	62,456.639	-	-	-
HM [9]	62,456.634	-	-	-
CSA [25]	62,456.633	10	500	10,000
ORCSA [25]	62,456.633	10	500	10,000
BBO [31]	62,456.793	50	400	20,000
HPSO-GSA [37]	62,455.616	-	-	-
MSFS	62,456.633	10	100	3000

The optimal solution of the system is shown in Table A4 in Appendix.

6. Conclusions

In this paper, a revamped version of the conventional stochastic fractal search algorithm, called modified stochastic fractal search algorithm, was proposed for solving the smooth or non-smooth ELD problem with different fuel cost function characteristics, different loads and different constraints. Based on two proposed modifications, the proposed MSFS has become a strong tool with a good solution quality, fast convergence, and stabilization of searching ability. The first modification is in charge of exploiting the global search space while the second one undertakes to enhance solution quality. Therefore, the proposed method has solutions better than SFS. As tested on four systems, fuel cost-saving level over SFS could reach 0.04 ($/h) for a three-unit system, 0.409 ($/h) for a six-unit system, 0.0429 ($/h) for a 10-unit system, and 0.06 ($/h) for a 20-unit system. Furthermore, the performance of the proposed method over other mentioned methods has also been scrutinized. The reduction in cost can be up to 16.65 ($/h) for the second test system, 0.282 ($/h) for the third test system and 0.16 ($/h) for the last test system. In addition, other costs of the proposed method, such as mean cost and maximum cost, are also less than those of the other ones. Consequently, it can be concluded that the proposed method is a useful optimization approach for searching solutions of the ELD problem.

With advantages of the proposed method, in future research, MSFS can be introduced to deal with ELD problems considering complicated models of thermal generating units [47], the combination of heat generators and power generators [48]. Besides, the considered ELD problems regarding renewable energies, such as wind power plants and solar power plants, are also interesting studies [49].

Author Contributions: L.H.P. and M.Q.D. have coded MSFS for solving systems of ELD problem. L.H.P., M.Q.D. and T.T.N. have written Sections 1, 3 and 4. V.-D.P. and H.-N.N. have been in charge of Section 2, Section 5, Section 6 and other duties.

Funding: This research received no external funding.

Conflicts of Interest: The authors proclaim that the publication of this paper does not conflict with any personal circumstances or interest.

Nomenclature

$AP^{l}_{k,j}$, $AP^{u}_{k,j}$	Lower and upper limits of the *j*th prohibited operation zones of the *k*th thermal generator
$AP^{\min}_{k}$, $AP^{\max}_{k}$	The upper and lower power output of the *k*th thermal generator
$AP^{\min}_{kMk}$, $AP^{\max}_{kMk}$	The upper and lower power output for the fuel type M_k of the *k*th thermal generator
AP_{SRP}	Sum of spinning reserve power of power system
K	Amplified factor for the violation.
M_k	Number of the M_k fuel type of the *k*th thermal generator
N	Number of thermal generators
N_1	Number of prohibited zones for the first thermal generator
N_{Ger}	Number of fitness assessments
N_{iter}	Number of iterations
N_k	Number of prohibited operation zones of the *k*th thermal generating unit
N_{pop}	Population size
P_{LD}	Total load demand
P_{TLL}	Total transmission line losses
PEN_{AP_1}	Penalty term for the violation of thermal generator 1.
PEN_{POZ,AP_1}	Penalty term for violating POZ constraint of the first thermal generator
PEN_{SRP}	Penalty term for the violation of spinning reserve power.
v_k, x_k, y_k, s_k, t_k	Constant fuel cost function coefficients of the *k*th thermal generator
v_{kM_k}, x_{kM_k}, y_{kM_k},s_{kM_k}, t_{kM_k}	Constant fuel cost function coefficients for the M_k fuel type of the *k*th thermal generator
X^{new1}_{d}	Newly updated position of X_d
X_{t1}, X_{t2}, X_{t3}, X_{t4}	Randomly picked solutions from current population
α, β, γ, ε, λ, D	Random number with value in the range [0, 1]

Appendix A

Table A1. The best solutions for system 1 found by MSFS.

Case 1		Case 2	
k	AP_k (MW)	k	AP_k (MW)
2	299.9079	2	149.7331
3	130.9249	3	400

Table A2. The best solutions for system 2 found by MSFS.

Case 3		Case 4		Case 5		Case 6	
k	AP_k (MW)	k	AP_k (MW)	k	AP_k (MW)	k	AP_k (MW)
2	123.8781	2	100	2	248.0103	2	173.2223
3	117.7802	3	50	3	217.7133	3	263.3614
4	50.0000	4	100	4	75.1815	4	139.0457
5	230.1763	5	110	5	335.4618	5	165.4284
6	229.6070	6	127.4212	6	335.6103	6	87.0688

Table A3. The best solutions for system 3 found by MSFS.

k	Case 7	Case 8	Case 9	Case 10	Case 11
	AP_k (MW)	AP_k (MW)	AP_k (MW)	AP_k (MW)	AP_k (MW)
2	202.3427	206.4573	210.9058	211.6626	211.4116
3	253.8953	265.7391	278.5441	280.7228	280.6591
4	233.0456	235.9531	239.0967	239.6315	239.4175
5	241.8297	258.0177	275.5194	278.4973	279.9549
6	233.0456	235.9531	239.0967	239.6315	240.7359
7	253.2750	268.8635	285.7170	288.5845	287.7287
8	233.0456	235.9531	239.0967	239.6315	239.2832
9	320.3832	331.4877	343.4934	428.5216	426.8477
10	239.3969	255.0562	271.9861	274.8667	275.8486

Table A4. The best solution for system 4 found by MSFS.

k	AP_k (MW)	k	AP_k (MW)	k	AP_k (MW)	k	AP_k (MW)
2	169.3077	7	115.2917	12	292.7916	17	66.8422
3	126.8774	8	116.4401	13	119.1725	18	88.0168
4	102.8520	9	100.3887	14	30.8405	19	100.7899
5	113.6378	10	105.8272	15	115.8406	20	54.3104
6	73.5494	11	150.2158	16	36.2444		

References

1. Balamurugan, R. Application of shuffled frog leaping algorithm for economic dispatch with multiple fuel options. In Proceedings of the 2012 International Conference on Emerging Trends in Electrical Engineering and Energy Management (ICETEEEM), Chennai, India, 13–15 December 2012; pp. 191–197. [CrossRef]
2. Vo, D.N.; Schegner, P.; Ongsakul, W. Cuckoo search algorithm for non-convex economic dispatch. *IET Gener. Transm. Distrib.* **2013**, *7*, 645–654. [CrossRef]
3. James, J.Q.; Li, V.O. A social spider algorithm for solving the non-convex economic load dispatch problem. *Neurocomputing* **2016**, *171*, 955–965. [CrossRef]
4. Baskar, S.; Subbaraj, P.; Rao, M.V.C. Hybrid real coded genetic algorithm solution to economic dispatch problem. *Comput. Electr. Eng.* **2003**, *29*, 407–419. [CrossRef]

5. Granville, S. Optimal reactive dispatch through interior point methods. *IEEE Trans. Power Syst.* **1994**, *9*, 136–146. [CrossRef]
6. Dodu, J.C.; Martin, P.; Merlin, A.; Pouget, J. An optimal formulation and solution of short-range operating problems for a power system with flow constraints. *Proc. IEEE* **1972**, *60*, 54–63. [CrossRef]
7. Aravindhababu, P.; Nayar, K.R. Economic dispatch based on optimal lambda using radial basis function network. *Intern. J. Electr. Power Energy Syst.* **2002**, *24*, 551–556. [CrossRef]
8. Fan, J.Y.; Zhang, L. Real-time economic dispatch with line flow and emission constraints using quadratic programming. *IEEE Trans. Power Syst.* **1998**, *13*, 320–325. [CrossRef]
9. Su, C.T.; Lin, C.T. New approach with a Hopfield modeling framework to economic dispatch. *IEEE Trans. Power Syst.* **2000**, *15*, 541–545. [CrossRef]
10. Lee, S.C.; Kim, Y.H. An enhanced Lagrangian neural network for the ELD problems with piecewise quadratic cost functions and nonlinear constraints. *Electr. Power Syst. Res.* **2002**, *60*, 167–177. [CrossRef]
11. Vo, D.N.; Ongsakul, W. Economic dispatch with multiple fuel types by enhanced augmented Lagrange Hopfield network. *Appl. Energy* **2012**, *91*, 281–289. [CrossRef]
12. Binetti, G.; Davoudi, A.; Naso, D.; Turchiano, B.; Lewis, F.L. A distributed auction-based algorithm for the nonconvex economic dispatch problem. *IEEE Trans. Ind. Inf.* **2014**, *10*, 1124–1132. [CrossRef]
13. Panigrahi, B.K.; Yadav, S.R.; Agrawal, S.; Tiwari, M.K. A clonal algorithm to solve economic load dispatch. *Electr. Power Syst. Res.* **2007**, *77*, 1381–1389. [CrossRef]
14. Bhattacharya, A.; Chattopadhyay, P.K. Biogeography-based optimization for different economic load dispatch problems. *IEEE Trans. Power Syst.* **2010**, *25*, 1064–1077. [CrossRef]
15. Bhattacharya, A.; Chattopadhyay, P.K. Application of biogeography-based optimization for solving multi-objective economic emission load dispatch problems. *Electr. Power Compon. Syst.* **2010**, *38*, 340–365. [CrossRef]
16. Yalcinoz, T.; Altun, H.; Uzam, M. Economic dispatch solution using a genetic algorithm based on arithmetic crossover. In Proceedings of the 2001 IEEE Porto Power Tech, Porto, Portugal, 10–13 September 2001. [CrossRef]
17. Jeyakumar, D.N.; Venkatesh, P.; Lee, K.Y. Application of multi objective evolutionary programming to combined economic emission dispatch problem. In Proceedings of the 2007 International Joint Conference on Neural Networks, Orlando, FL, USA, 12–17 August 2007; pp. 1162–1167. [CrossRef]
18. Yang, X.S.; Hosseini, S.S.S.; Gandomi, A.H. Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect. *Appl. Soft Comput.* **2012**, *12*, 1180–1186. [CrossRef]
19. Noman, N.; Iba, H. Differential evolution for economic load dispatch problems. *Electr. Power Syst. Res.* **2008**, *78*, 1322–1331. [CrossRef]
20. Mandal, B.; Roy, P.K.; Mandal, S. Economic load dispatch using krill herd algorithm. *Intern. J. Electr. Power Energy Syst.* **2014**, *57*, 1–10. [CrossRef]
21. Basu, M.; Chowdhury, A. Cuckoo search algorithm for economic dispatch. *Energy* **2013**, *60*, 99–108. [CrossRef]
22. Thao, N.T.P.; Thang, N.T. Environmental economic load dispatch with quadratic fuel cost function using cuckoo search algorithm. *Intern. J. U E–Serv. Sci. Technol.* **2014**, *7*, 199–210. [CrossRef]
23. Lin, W.M.; Cheng, F.S.; Tsay, M.T. An improved tabu search for economic dispatch with multiple minima. *IEEE Trans. Power Syst.* **2002**, *17*, 108–112. [CrossRef]
24. Nguyen, T.T.; Vo, D.N. The application of one rank cuckoo search algorithm for solving economic load dispatch problems. *Appl. Soft Comput.* **2015**, *37*, 763–773. [CrossRef]
25. Tran, C.D.; Nguyen, T.T.; Hoang, H.M.; Nguyen, B.Q. One Rank Cuckoo Search Algorithm for Bi-Objective Load Dispatch Problem. *Intern. J. Grid Distrib. Comput.* **2016**, *9*, 13–26. [CrossRef]
26. Selvakumar, A.I.; Thanushkodi, K. Anti–predatory particle swarm optimization: Solution to nonconvex economic dispatch problems. *Electr. Power Syst. Res.* **2008**, *78*, 2–10. [CrossRef]
27. Park, J.B.; Lee, K.S.; Shin, J.R.; Lee, K.Y. A particle swarm optimization for economic dispatch with nonsmooth cost functions. *IEEE Trans. Power Syst.* **2005**, *20*, 34–42. [CrossRef]
28. Niu, Q.; Zhou, Z.; Zhang, H.Y.; Deng, J. An improved quantum–behaved particle swarm optimization method for economic dispatch problems with multiple fuel options and valve–points effects. *Energies* **2012**, *5*, 3655–3673. [CrossRef]
29. Secui, D.C. A new modified artificial bee colony algorithm for the economic dispatch problem. *Energy Convers. Manag.* **2015**, *89*, 43–62. [CrossRef]

30. Safari, A.; Shayeghi, H. Iteration particle swarm optimization procedure for economic load dispatch with generator constraints. *Expert Syst. Appl.* **2011**, *38*, 6043–6048. [CrossRef]
31. Secui, D.C. A modified symbiotic organism's search algorithm for large scale economic dispatch problem with valve–point effects. *Energy* **2016**, *113*, 366–384. [CrossRef]
32. Rughooputh, H.C.; King, R.A. Environmental/economic dispatch of thermal units using an elitist multiobjective evolutionary algorithm. In Proceedings of the IEEE International Conference on Industrial Technology, Maribor, Slovenia, 10–12 December 2003; Volume 1, pp. 48–53. [CrossRef]
33. Niknam, T.; Mojarrad, H.D.; Meymand, H.Z. Non–smooth economic dispatch computation by fuzzy and self–adaptive particle swarm optimization. *Appl. Soft Comput.* **2011**, *11*, 2805–2817. [CrossRef]
34. Song, Y.H.; Wang, G.S.; Wang, P.Y.; Johns, A.T. Environmental/economic dispatch using fuzzy logic controlled genetic algorithms. *IEE Proc.–Gener. Transm. Distrib.* **1997**, *144*, 377–382. [CrossRef]
35. Lu, H.; Sriyanyong, P.; Song, Y.H.; Dillon, T. Experimental study of a new hybrid PSO with mutation for economic dispatch with non–smooth cost function. *Intern. J. Electr. Power Energy Syst.* **2010**, *32*, 921–935. [CrossRef]
36. Sayah, S.; Hamouda, A. A hybrid differential evolution algorithm based on particle swarm optimization for nonconvex economic dispatch problems. *Appl. Soft Comput.* **2013**, *13*, 1608–1619. [CrossRef]
37. Iqbal, H.Z.; Shafique, A. A hybrid evolutionary algorithm for economic load dispatch problem considering transmission losses and various operational constraints. In Proceedings of the 2016 International Conference on Intelligent Systems Engineering (ICISE), Islamabad, Pakistan, 15–17 January 2016; pp. 202–209. [CrossRef]
38. Vaisakh, K.; Reddy, A.S. MSFLA/GHS/SFLA–GHS/SDE algorithms for economic dispatch problem considering multiple fuels and valve point loadings. *Appl. Soft Comput.* **2013**, *13*, 4281–4291. [CrossRef]
39. Cai, J.; Li, Q.; Li, L.; Peng, H.; Yang, Y. A hybrid FCASO–SQP method for solving the economic dispatch problems with valve–point effects. *Energy* **2012**, *38*, 346–353. [CrossRef]
40. Chen, Y.P.; Peng, W.C.; Jian, M.C. Particle swarm optimization with recombination and dynamic linkage discovery. *IEEE Trans. Syst. Man Cybern. Part B* **2007**, *37*, 1460–1470. [CrossRef]
41. Ciornei, I.; Kyriakides, E. A GA–API solution for the economic dispatch of generation in power system operation. *IEEE Trans. Power Syst.* **2012**, *27*, 233–242. [CrossRef]
42. Reddy, A.S.; Vaisakh, K. Shuffled differential evolution for economic dispatch with valve point loading effects. *Intern. J. Electr. Power Energy Syst.* **2013**, *46*, 342–352. [CrossRef]
43. Salimi, H. Stochastic fractal search: a powerful metaheuristic algorithm. *Knowl.–Based Syst.* **2015**, *75*, 1–18. [CrossRef]
44. Saadat, H. *Power System Analysis*, 1st ed.; McGraw–Hill: New York, NY, USA, 1999.
45. Shang, Y. Lie algebraic discussion for affinity based information diffusion in social networks. *Open Phys.* **2017**, *151*, 705–711. [CrossRef]
46. Shang, Y. An agent based model for opinion dynamics with random confidence threshold. *Commun. Nonlinear Sci. Numer Simul.* **2014**, *19*, 3766–3777. [CrossRef]
47. Meng, A.; Li, J.; Yin, H. An efficient crisscross optimization solution to large–scale non–convex economic load dispatch with multiple fuel types and valve–point effects. *Energy* **2016**, *113*, 1147–1161. [CrossRef]
48. Nguyen, T.T.; Nguyen, T.T.; Vo, D.N. An effective cuckoo search algorithm for large–scale combined heat and power economic dispatch problem. *Neural Comput. Appl.* **2018**, *30*, 3545–3564. [CrossRef]
49. Xiao, Y.; Wang, Y.; Sun, Y. Reactive Power Optimal Control of a Wind Farm for Minimizing Collector System Losses. *Energies* **2018**, *11*, 3177. [CrossRef]