

Research Article

Optimal Dispatch of Reactive Power Using Modified Stochastic Fractal Search Algorithm

Thang Trung Nguyen ¹, **Dieu Ngoc Vo**,² **Hai Van Tran**,³ and **Le Van Dai** ⁴

¹Power System Optimization Research Group, Faculty of Electrical and Electronics Engineering, Ton Duc Thang University, Ho Chi Minh City, Vietnam

²Department of Power Systems, Ho Chi Minh City University of Technology, VNU-HCM, Ho Chi Minh City, Vietnam

³Faculty of Electrical and Electronic Technology, Ho Chi Minh City University of Food Industry, Ho Chi Minh City, Vietnam

⁴Institute of Research and Development, Duy Tan University, Danang, Vietnam

Correspondence should be addressed to Le Van Dai; levandai@duytan.edu.vn

Received 30 January 2019; Revised 1 April 2019; Accepted 23 April 2019; Published 28 May 2019

Academic Editor: Yan-Ling Wei

Copyright © 2019 Thang Trung Nguyen et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper applies a proposed modified stochastic fractal search algorithm (MSFS) for dealing with all constraints of optimal reactive power dispatch (ORPD) and finding optimal solutions for three different cases including power loss optimization, voltage deviation optimization, and L-index optimization. The proposed MSFS method is newly constructed in the paper by modifying three new solution update mechanisms on standard stochastic fractal search algorithm (SSFS). The first modification is to keep only one formula and abandon one formula in the diffusion process while the second modification and the third modification are used in the first update and the second update. In two updates of SSFS, solutions with low quality are updated with high probability while other solutions with high quality do not get chances to be updated. This manner results in the fact that some promising solutions around the high quality solutions can be missed. In order to tackle this restriction, the second modification of MSFS is to newly update the worst solutions in the first update and the best solutions in the second update. In the third modification, all existing formulas of SSFS in the two updates are abandoned and the same new proposed technique is used for updating such solutions in two updates. Compared to SSFS, the three modifications can bring advantages to MSFS such as using smaller number of produced solutions per iteration, spending shorter execution time, finding better optimal solutions, and owning more stable search ability. Furthermore, the proposed method also sees its effectiveness and robustness over SSFS by testing on IEEE 30-bus system and IEEE 118-bus system with three different single objectives for each system. The proposed method can find less minimum, average, and maximum for all the cases in addition to faster search speed. Besides, the proposed method is also compared to other methods such as PSO-based method group, GA-based method group, DE-based method group, and other recent methods. Result comparisons also indicate that the proposed method can be more efficient than almost all these methods with respect to less minimum and smaller values of control parameters. As a result, evaluation of the performance of the proposed method is that it should be used for seeking solutions of ORPD problem.

1. Introduction

Optimal reactive power dispatch (ORPD), a separate problem of optimal power flow (OPF) computation, is one of the material optimization problems for secure and stable operation in modern power systems. The main task of such ORPD problem is to find optimum settings of all control parameters such as reactive power output of generators, on-load tap changer of transformers, and shunt capacitor power outputs.

In addition, dependent parameters as voltage of load buses, apparent power flow through transmission lines, and reactive power of generators must be within allowable specified range [1, 2]. In addition, different objective functions of ORPD problem comprising total transmission line power losses (P_{Loss}), total voltage deviation (TVD), and amelioration of voltage stability index (VSI) are always optimized. When reaching all considered objectives, the considered power systems are working economically and stably.

In the first stage, many traditional numerical methods called deterministic methods such as gradient search (GS) [3], Newton method (NM) [4], interior point method (IPM) [5–7], linear program (LP) [8–10], dynamic programming method (DPM)[11], quadratic programming method (QPM) [12, 13], and Lagrangian method (LM) [14] have found optimal solutions with acceptable quality, but these methods have met many disadvantages such as highly time consuming manner, high number of iterations, huge number of computation processes, incapability of handling nondifferentiable constraints and objective functions, and easily falling into local optimum solution zone. In this regard, the ones should be retired to make room for new algorithms that have better search ability. Recently, many metaheuristic methods inspired from nature phenomenon or behavior of animals have been more widely and successfully applied for solving such ORPD problem. Many methods have been continually grown and become a big family of methods like the variants of genetic algorithm (GA) [15–19], variants of differential evolution (DE) [20–24], variants of particle swarm optimization (PSO) [25–31], variants of gravitational search algorithm (GSA) [32–35], and many new standard methods [36–49]. In adaptive genetic algorithm (AGA) [15], the method changed both mutation probability and crossover probability based on comparison of the maximum fitness value and average fitness value of the population to enhance global search quality and fast convergence speed. Authors in [16] proposed an improved genetic algorithm (IGA) by using real code for continuous variables and binary code for discrete variables to decrease computation time. However, results of IGA method compared to those of real GA and Binary GA showed that its computation time was insignificantly improved. A version of GA called an efficient genetic algorithm (EGA) in [17] has applied Misum fuzzy goal programming model to find important weights of the goal function. This function was considered as the fitness function of GA for solving ORPD problem. As a result, EGA has decreased computation process and avoided approximation error of classical linearization methods. Alam and De proposed hybrid loop-genetic based algorithm (HLGA) in [18]. HLGA has used the global search feature of GA to find the good search space and then implemented the local search process utilizing loop optimization method within the identified spaces. When comparing to GA, the method has been faster convergent to optimal solutions, but there was no improvement of such obtained results. Thus, HLGA has still coped with a disadvantage so that solution quality was low. DE group has been introduced to ORPD problem such as standard differential evolution (DE) [20], modified differential evolution algorithm (MDE) [21], quasi-oppositional differential evolution (QODE) [22], multiobjective differential evolution (MODE) [23], a novel hybrid chaotic artificial bee colony, and differential evolution algorithm (CABC-DE) [24]. These improved DE methods were formed mainly based on the improved mutation, crossover, and selection operation or the combination of DE and other algorithms for surmounting several drawbacks of DE such as easily getting in stuck in local optimal zones and premature convergence. MDE has applied time-varying chaotic mutation rate F and crossover

rate CR to replace standard operations of DE for time saving and reduction of setting the parameters. This method has been also proven on IEEE 30-bus system and compared to three other DE methods such as conventional DE, self-adaptive DE (SaDE), and E with integration of new mutation and crossover strategies (NMCDE). The comparison results of QODE and other methods showed that the standard deviation of active power loss, voltage deviation, and voltage stability index for all test systems was very small. So, readers cannot see the effectiveness and robustness of QODE over other ones. CABC-DE was an improved version of DE based on the combination of basic characteristic of DE and chaotic artificial bee colony algorithm (CABC). In such method, search capacity of DE was significantly improved by applying the foraging behaviors of observation bees and the search operations of detective bees from artificial bee colony algorithm (ABC). CABC-DE has been compared to the standard DE via the results obtained from IEEE 14-bus system and IEEE 30-bus systems. Simple comparison via power loss could see that this method could get good solution with small populations than classical DE. PSO and its improved versions have been applied for ORPD problem such as the improved particle swarm optimization (IPSO) [25], modified particle swarm optimization (MPSO) [26], comprehensive-learning particle swarm optimization (CLPSO) [27], improved pseudo-gradient search-particle swarm optimization (IPGS-PSO) [28], hybrid particle swarm optimization and imperialist competitive algorithms (HPSO-ICA) [29], fuzzy adaptive heterogeneous comprehensive-learning particle swarm optimization (FAHCLPSO) [30], and hybrid cuckoo search and particle swarm optimization (CS-PSO) [31]. In conventional PSO, the update operation of each particle was dependent on the best position of each particle (P_{best}) and the best position of the whole particles (G_{best}). With searching process, most of particles always updated around in specified position. Thus, PSO only explored zones near the best particle, which could be in local optimal zones. In order to tackle the drawback, IPSO used a new update operation that each particle would be updated by the best position of it, the best position of the groups, and the best position of the neighboring particles. By using the mechanism, capacity of finding the global solution of IPSO was enhanced better and its convergence speed was faster than PSO. MPSO applied the Michalewicz's nonuniform mutation theory into PSO to diversify particle position search processes. Adding new strategy, MPSO decreased the possibility of getting stuck in local zones. However, the effectiveness of MPSO has been only investigated for transmission line power loss objective while other cases like total voltage deviation (TVD) and voltage stability index have not been demonstrated. CLPSO presented in [27] has proposed three modifications relying on the structure of PSO. Firstly, the velocity of each particle could be updated by using its P_{best} or potentially other particle G_{best} in the group. Secondly, a new formula has been introduced to adaptively obtain the inertia weight values. Lastly, a new range has been suggested to limit the velocity value. With above modifications, CLPSO has overcome the premature convergence of PSO. Another improved version of PSO, called the improved pseudogradient search-particle

swarm optimization (IPGS-PSO), has been applied to solve ORPD problem with three independent objectives comprising power loss, voltage deviation, and voltage stabilization index. Pseudogradient theory has been added to PSO to find the best velocity direction for particles. So, IPGS-PSO could determine better direction for velocity update. As a result, the method was a more effective method than other PSO methods such as PSO with time-varying inertia weight (PSO-TVIW), PSO with time-varying acceleration coefficients (PSO-TVAC), self-organizing PSO with TVAC (SPO-PSO-TVAC), PSO with constriction factor PSO-CF, pseudogradient search PSO (PGS-PSO), stochastic weight trade-off PSO (SWT-PSO), and pseudogradient search and SWT-PSO (PGSWT-PSO). HPSO-ICA was a combination of the imperialist competitive algorithm and improved PSO. In HPSO-ICA, authors supplemented an independent country into imperialist competitive algorithm to enhance the searching capability of the method. By this work, this method has spent more time than PSO and ICA searching solutions. FAHCLPSO has applied the Mamdani fuzzy rules to adjust the inertia weight of Heterogeneous Comprehensive-Learning Particle Swarm Optimization (HCLPSO) aiming to avoid trapping in local optimal zones. Via the comparison of results obtained from IEEE 30-bus system, IEEE 118-bus system, and IEEE 354-bus system with the objective of power loss minimization and voltage deviation optimization, FAHCLPSO has shown its easy application and its highly computational efficiency when comparing other PSO methods and metaheuristic algorithms. In addition to the presence of the three largest groups, some small groups have been also introduced to tackle ORPD problem such as the gravitational search algorithm (GSA) [32–34], improved GSA with feasible conditional selection strategies (IGSA-FCSS) [35], quasi-oppositional teaching learning based optimization (QOTLBO) [36], teaching learning based optimization (TLBO) [36], modified Gaussian barebones based TLBO (MGBTLBO) [37], and Gaussian barebones based TLBO (GBTLBO) [37]. From 2015 to 2017, a high number of methods were employed for ORPD problem such as the hybrid Nelder–Mead simplex based firefly algorithm (HNMS-FA) [38], Artificial Bee Colony Algorithm (ABC) [39], differential search algorithm (DSA) [40], exchange market algorithm (EMA) [41], chaotic krill herd algorithm (CKHA) [42], gray wolf optimizer (GWO) [43], Gaussian barebones water cycle algorithm (GBBWCA) [44], ant lion optimizer (ALO) [45], moth-flame optimization technique (MFOT) [46], whale optimization algorithm (WOA) [47], Ant Colony Optimization Algorithm (ACOA) [48], and backtracking search algorithm (BTSA) [49]. All in all, most of these methods had a strong search ability and outperformed deterministic algorithms, original metaheuristic algorithms in terms of solution quality, computing time, and convergence speed.

In the paper, we have proposed a modified stochastic fractal search algorithm (MSFS) for handling constraints and finding three different single objectives of ORPD problem. Standard stochastic fractal search algorithm (SSFS) was first introduced in 2015 by Salimi [50]. The author applied SSFS for twenty-three benchmark functions and compared

results with other popular metaheuristic methods such as cuckoo search algorithm, modified cuckoo search algorithm, backtracking search optimization algorithm, derandomized evolution strategy with covariance matrix adaptation (CMA-ES), animal migration optimization, PSO, DE, ABC, and GSA. The minimum of all benchmark functions found by SSFS was less than those from all methods, but it should be pointed out that the author has used the same population size (N_{pop}) and maximum number of iteration (Max_{iter}) for all compared methods to SSFS. It is clear that the selection is not fair for these implemented methods because total number of produced solutions of these methods for each run is not equal. Especially, SSFS has used very high number of diffusions (N_d) equaling to 10 while the total produced solutions per iteration (N_{spi}) by SSFS is ($N_{pop} * N_d + 2 * N_{pop}$). For other methods, N_{spi} is much smaller than that of SSFS; for example, it is approximately equal to ($N_{pop} * 2$) for CSA, ($N_{pop} * 1$) for PSO, and DE. Consequently, the real performance of SSFS could not be demonstrated persuasively. In spite of the limitation, SSFS has been also applied for different optimization problems in electrical engineering fields such as status estimation of power system [51], DC motor control [52], environment and economic dispatch [53], PID voltage regulator [54], wind integrated multiobjective power dispatch [55], and position determination of Distributed Generators in distribution power network [56]. These studies have tried to prove the outstanding performance of SSFS over other ones via comparison of output parameters such as the fuel cost, emission, speed response, and obtained voltage; however, they have forgotten that convergence speed reflected via comparison of N_{spi} and Max_{iter} is also a very important comparison criterion in addition to such objectives. Thus, other authors have investigated the real performance of SSFS and they have identified the algorithm has needed more improvements for avoiding drawbacks such as low effectiveness for large scale problem, complicated and nondifferentiable objective functions, and high probability of finding local optimums. As a result, improved versions of SSFS have been introduced to readers with difference applications in mathematical and electrical engineering fields. These methods are SFS with DE and Gaussian walk-based diffusion (SFS-DE) [57], hybrid stochastic fractal search and pattern search technique (HSFS-PS) [58], hybrid SFS based on different mutation models of DE (HSFS-DE) [59], improved SFS (ISFS) [60], and chaos-based SFS (CSFS) [61]. In general, almost all improved methods have used different mutation models of DE available in [62]. SFS-DE has applied current-to-best mutation model for solutions with a probability of 50%. If random number was lower than 0.5, the current-to-best mutation model has been employed. Otherwise, the standard diffusion process of SSFS has still been applied. This method has been tested on thirty benchmark functions and compared to other six methods including different improved versions of DE and SSFS. The authors have set control parameters for all such implemented methods based on fair comparison with the same number of fitness evaluations. The method has found better results than all the compared methods for half of benchmark functions and it has gotten worse results for another half. Clearly, the

performance of the method is still a question for readers to decide if it is really effective for their studied problems. HSFS-PS was a combination of SSFS and pattern search (PS) algorithm in which SSFS was completely and successfully carried out first and then optimal solution was used to run PS algorithm. Thus, compared to SSFS and PS, the method was more complicated and highly time consuming. HSFS-PS has been employed for designing different controllers based on PI, PID, and cascaded PI-PD. Final evaluation in the study was that cascaded PI-PD controller, which was yielded by applying HSFS-PS, was superior to remaining controllers applying HSFS-PS. So, the performance of HSFS-PS is now still a question for readers because it was not proven to be better than any optimization algorithms, even SSFS and PS. HSFS-DE in [59] has abandoned original diffusion process of SSFS, and added mutation technique based on target-to-best/1 in [62] for new diffusion process. In addition, self-adaptive crossover technique in [63] has been applied to retain new solutions from such new diffusion process. This method together with PSO, DE, and SSFS has been implemented for tuning parameter of chaotic system based on fractional order. HSFS-DE together with such three compared ones has been fairly implemented by thoroughly selecting population size as well as the number of iterations. Simulation results have indicated that HSFS-DE could reach the highest accuracy and fastest response with very low oscillation for Chen, Lorenz, and Chua systems. The proposed modification on ISFS in [60] and CSFS in [61] is totally different from those in [57–60]. In fact, ISFS has suggested using sigmoid, inverse cosine, exponential, hyperbolic cosine, and hyperbolic cosecant function for calculating standard deviation δ in two formulas of diffusion process while CSFS has tried to apply different chaotic maps for replacing uniform random number within zero and one, which was used in the first formula of diffusion process. ISFS has been applied to different IEEE systems for comparison with SSFS, but the exact name of the considered problem has not been mentioned in [60]. The study also pointed out that ISFS was superior to SSFS in terms of better results and much faster execution time; however, control parameters consisting of population size N_{pop} , number of diffusion N_d , and maximum number of iteration Max_{Iter} have not been reported in the study. Clearly, the outstanding performance of ISFS was not demonstrated persuasively and it can be wondered if readers should use it as the most reasonable choice. Ten versions of chaotic maps have been employed, forming ten CSFS methods in [61]. Three main control parameters have been set equally for both SSFS and ten CSFS methods, and result comparisons have shown that the performance of such ten methods has not been stable and superior to SSFS because for some study cases SSFS found worse results than them but for other study cases SSFS found better results than them. The unstable performance between SSFS and such ten CSFS is due to the impact of randomization in formulas in diffusion, the first and the second update processes. Furthermore, the study has stated that such ten methods have been more effective and robust than PSO; however, the authors have set smaller values for control parameters for PSO. For example, for a 33-bus system $N_{pop}=100$ and $Max_{Iter}=100$ were set for PSO

but $N_{pop}=50$, $Max_{Iter}=100$ and $N_d=5$ were set for such ten CSFS methods. Thus, through the whole search CSFS has produced approximately $(50*5+2*50)*100=35,000$ solutions while PSO has produced $(100*100)=10,000$ solutions. Clearly, CSFS could take more advantages for finding better results than PSO. The selection manner continued to be used for 69-bus system and 118-bus system, and CSFS methods continued to show better results than PSO. In summary, authors have proposed the different improved versions of SSFS to get the best performance of their suggested methods, but they have forgotten to select fair comparison criteria with intent to give an accurate evaluation of performance between implemented. In the paper, we have proposed a modified stochastic search algorithm (MSFS) by applying three modifications on three new solution generations of SSFS. In the first generation, we have used only one formula and canceled one formula while in the second and third new solution generations, we have proposed a new technique for generating new solutions. In SSFS, there are two different formulas used in each new solution generation by a condition of comparison between a random number and a predetermined number, which are normally selected to be 0.5. The fact that the second formula in each generation is less effective than the first formula along with use of both formulas leads to time consuming manner as well as high number of computations. Thus, in the new proposed technique we have given only one new formula with greatly effective performance. In addition, we have changed old solutions, which are selected to be newly updated. Namely, SSFS updates approximately all solutions for the first and the second update processes, but we have suggested updating some worst solutions for the first update process but updating other best solutions for the second update. The sum of updated solutions in the first and the second updates is equal to N_{pop} while that number of SSFS is $2 * N_{pop}$. By applying the three modified techniques, our contributions are as follows:

- (i) Indicate disadvantages of formulas used in SSFS.
- (ii) Propose to keep high efficient formulas and abandon ineffective ones.
- (iii) Cancel the update of N_{pop} solutions and calculation of N_{pop} evaluation functions, leading to reduction of high number of computations as well as execution time.
- (iv) Propose new techniques for producing new solutions in the first and the second update processes.
- (v) Find better results for different systems of ORPD problem.

In the paper, the proposed method together with SSFS using Gaussian random walk (SSFS-Gauu) and SSFS with Levy random walk (SSFS-Levy) have been applied for seeking three different objectives of two standard IEEE systems with 30 and 118 buses of ORPD problem. The results obtained by the three methods consisting of minimum, mean, maximum, and standard deviation in addition to control parameters are compared with each other. Furthermore, the results of the proposed method are also compared to those from

other existing methods mentioning introduction for further investigation of the proposed method performance.

2. Optimal Reactive Power Dispatch Problem Formulation

ORPD problem can be mathematically expressed in terms of objective functions and a set of constraints. Normally, ORPD considers three different targets consisting of minimization of total active power loss, minimization of total voltage deviation of all load buses, and minimization of voltage stability L-index. Besides, all constraints in transmission power network such as voltage bounds of all load buses, upper bound of apparent power flow through transmission branches, reactive power bounds and voltage bounds of generators, and real and reactive power flow balance at each bus. The detail of objectives and constraints is as follows.

2.1. Objective Functions

- (1) *Reduction of total active power loss*: total active power loss P_{Loss} in all branches of high voltage transmission power networks is significant, causing ineffectiveness for power system. Thus, reducing total reactive power loss is one of the most important targets when operating power system. The objective is described by [20]:

$$\begin{aligned} \text{Minimize } P_{Loss} & \\ &= \sum_{i=1}^{N_{bus}} \sum_{\substack{j=1 \\ j \neq i}}^{N_{bus}} G_{l_{inell}} [V_i^2 \\ &+ V_j^2 - 2V_i V_j \cos(\varphi_i - \varphi_j)] \end{aligned} \quad (1)$$

- (2) *Reduction of total voltage deviation of all load buses*: power quality consists of two most important factors, voltage and frequency, in which frequency can be controlled by tuning reactive power balance while voltage control is dependent on many factors such as transformer, reactive power of generator, and total reactive power loss in branches and capacitors. Voltage control is effective when total voltage deviation is very small, nearby zero. The objective can be described by the following model [28]:

$$\text{Minimize } TVD = \sum_{i=1}^{N_{load}} |V_{loadi} - 1| \quad (2)$$

where V_{loadi} is voltage of load bus i and 1 Pu is the expected value of each load bus.

- (3) *Reduction of L-index*: voltage instability is one of the most harmful phenomenon for power system, which can cause voltage collapse gradually even immediately. Thus, the enhancement of voltage stability is one of the most important targets as two objective mentioned above. The enhancement of voltage stability is equivalent to minimization of voltage stability

indicator, normally called L-index at each bus in power system. The enhancement of voltage stability is carried out by reducing the highest value of L-index at one bus in the power system. Namely, L-index at each bus i and the voltage stability enhancement are mathematically formulated by [11]:

$$L_i = \left| 1 - \frac{\sum_{j=1}^{N_G} Y_{ij} V_j}{V_i} \right|; \quad i = 1, 2, \dots, N_{bus} \quad (3)$$

$$\text{Minimize } \max(L_i); \quad i = 1, \dots, N_{bus} \quad (4)$$

where L_i is the L-index value of bus i and Y_{ij} is the mutual admittance between buses i and j .

2.2. *Constraints. Active and reactive power flow balance at each bus*: at each bus, active power flow is the sum of three terms, generator, load, and lines while reactive power flow at each bus is the sum of four terms, generator, load, lines, and capacitors. The balance of all the terms must be exactly met by the following models [27]:

$$\begin{aligned} P_{Gi} - P_{di} & \\ &= V_i \sum_{j=1}^{N_{bus}} V_j [G_{l_{inell}} \cos(\varphi_i - \varphi_j) + B_{l_{inell}} \sin(\varphi_i - \varphi_j)]; \end{aligned} \quad (5)$$

$$i = 1, \dots, N_{bus}$$

$$\begin{aligned} Q_{Gi} + Q_{ci} - Q_{di} & \\ &= V_i \sum_{j=1}^{N_{bus}} V_j [G_{l_{inell}} \sin(\varphi_i - \varphi_j) - B_{l_{inell}} \cos(\varphi_i - \varphi_j)]; \end{aligned} \quad (6)$$

$$i = 1, \dots, N_{bus}$$

In addition, all electric components in power systems such as generators, transformers, conductors, capacitor banks, and loads are also constrained by security work capability. Thus, electrical bounds of these components are also considered seriously during operation process. In fact, they are subject to the following rules [27]:

$$Q_{Gi, \min} \leq Q_{Gi} \leq Q_{Gi, \max}; \quad i = 1, \dots, N_G \quad (7)$$

$$V_{G, \min} \leq V_{Gi} \leq V_{G, \max}; \quad i = 1, \dots, N_G \quad (8)$$

$$T_{\min} \leq T_i \leq T_{\max}; \quad i = 1, \dots, N_t \quad (9)$$

$$S_l \leq S_{l, \max}; \quad l = 1, \dots, N_{line} \quad (10)$$

$$Q_{ci, \min} \leq Q_{ci} \leq Q_{ci, \max}; \quad i = 1, \dots, N_c \quad (11)$$

$$V_{load, \min} \leq V_{loadi} \leq V_{load, \max}; \quad i = 1, \dots, N_{load} \quad (12)$$

3. The Proposed Modified Stochastic Fractal Search Algorithm (MSFS)

3.1. *Overview on Standard Stochastic Fractal Search Algorithm (SSFS)*. SSFS comprised three new solution generations

corresponding to three main stages, diffusion process, the first update process, and the second update process in which diffusion process is the first generation, the first update is the second new solution generation and the second update is the third new solution generation. In the diffusion process, each point will be diffused into N_d different points. Therefore, there are N_d times number of points, which are newly

updated in the diffusion. SSFS uses Gaussian walk to produce new solutions for diffusion process in which there are two options for each solution. If random number of the d th solution $rand_d$ are less than $walk$, which is selected within 0 and 1, the first option is selected. Otherwise, the second option is selected. The diffusion process is described by the following model:

$$Point_d^{new} = \begin{cases} Gaussian(Point_{best}, \alpha) + \varepsilon \cdot (Point_{best} - Point_d) & \text{if } rand_d < walk \\ Gaussian(Point_d, \alpha) & \text{else} \end{cases} \quad (13)$$

where $point_{best}$ is the best point among the current population; $rand_d$ is a random number for $point_d$; and α is obtained by

$$\alpha = \left| \frac{\log(CI)}{CI} \times (Point_d - Point_{best}) \right| \quad (14)$$

where CI is current iteration.

The first update process is carried out by using only formula (15) and the second update process is based on (16). However, not every solution in the whole population is updated but depends on another condition. Before applying the first and second update processes, the whole population is evaluated and ranked in descending order of fitness function value. Then, the order of solutions is set to their rank, solution d corresponding to $rank_d$. For example, the worst solution is with $rank_j=1$ and the best solution is with $rank_{N_{pop}}=N_{pop}$. If a random number λ_d within 0 and 1 of solution d is higher than $rank_d/N_{pop}$, solution d is newly updated by using (15) or (16). Otherwise, it remains unchanged.

$$Point_{d1}^{new} = Point_1 - rand \times (Point_2 - Point_d) \quad (15)$$

$$Point_{d2}^{new} = \begin{cases} Point_d - \varepsilon \times (Point_3 - Point_{best}) & \text{if } \beta_d \leq 0.5 \\ Point_d + \varepsilon \times (Point_3 - Point_d) & \text{else } \beta_d > 0.5 \end{cases} \quad (16)$$

The whole search process of SSFS can be described in flowchart of Figure 1.

3.2. The Proposed Modified Stochastic Fractal Search Algorithm. In the proposed MSFS, we carry out three modifications on diffusion process, the first update and the second update. The modifications are described in detail as follows:

3.2.1. The First Modification. In the first modification, we suggest using the first model for all points by setting $walk$ to 1. It is clearly seen that the first model considers the impact of each point and the best point for producing updated step size while the second model only utilizes Gaussian random walk. Control variables in each point are dependent on different considered optimization problem, so Gaussian random walk

cannot result in promising solutions with high quality. As a result, diffusion is carried out by using the following formula:

$$Point_d^{new} = Gaussian(Point_{best}, \alpha) + \varepsilon \cdot (Point_{best} - Point_d) \quad (17)$$

Furthermore, using only the proposed formula can reduce the selection of values for parameter $walk$ and shorten simulation time that SSFS has coped with so far.

3.2.2. The Second Modification. In the second modification, we change update strategy in the first and second update processes. SSFS updates almost all solutions depending on a comparison of random number for the two update processes shown in step 10 and step 16 in Figure 1. Thanks to the steps, it can be seen that almost all the worst solutions are newly updated with very high probability while the best solutions are hardly ever newly updated for the two processes. Clearly, the manner can miss promising solutions, which may be around the best solutions. Furthermore, the search ability of (15) in the first update, and (16) in the second update is less effective. Thus, the generation of N_{pop} solutions in the first update and the generation of N_{pop} solutions in the second update are time consuming and ineffective. In order to avoid overlapping the disadvantages of SSFS, the second proposed modification is applied. We choose a fraction P_a of the population, $(P_a * N_{pop})$ solutions with the low quality to be updated in the first update process while $[(1 - P_a) * N_{pop}]$ solutions with high quality will be updated for the second update process. Besides, we also propose other formulas, which are different from formulas (15) and (16), for updating new solutions in the first and the second update processes. The change will be presented in the third modification below.

3.2.3. The Third Modification. In the third modification, we propose a new technique for updating new solutions of the first and the second update processes. SSFS uses formulas (15) and (16), which are similar to mutation operation of classical DE in [20] and coping with different limitations such as easily trapping in local optimal zones and hardly ever converging to global optimum. Furthermore, both formula (15) and the first model in formula (16) are totally different from other metaheuristic since new solutions are updated

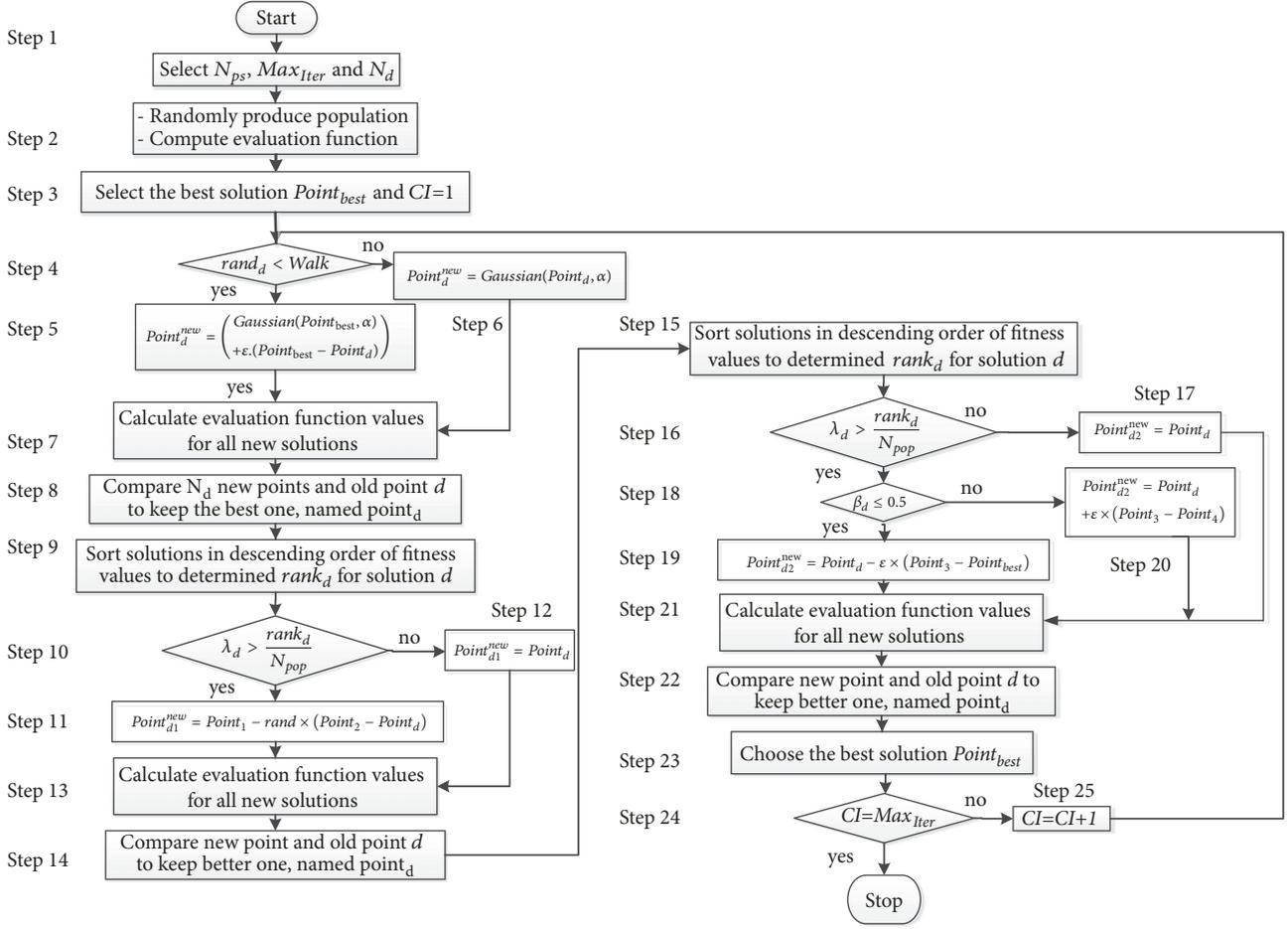


FIGURE 1: Flowchart of using SSFS for solving a typical optimization problem.

by subtracting an updated step size from old solutions while approximately all methods add an updated step size to old solutions for finding new solutions, namely, PSO [25–31] with velocity and position calculation formulas, Cuckoo search algorithm [66] with Levy flight formula and strange egg discovery formula, Bat algorithm [67] with formulas calculating frequency, velocity and position, DE [20] with mutation formula, and GA [65] with polynomial mutation formula, etc. Improved social spider optimization (ISSO) was proposed in [64] by performing several modifications on social spider optimization (SSO) in which one main modification was to abandon one formula with subtracting updated step sizes and keep another with adding updated step size. ISSO was more effective than SSO in terms of capability of finding the best solutions and shortening execution time. The review on many metaheuristic algorithms indicates that SSFS can miss the promising solutions with high quality. As a result, in the proposed technique, we suggest formulas like (15) and the first model in formula (16) should be canceled and replaced with more effective formulas.

At the beginning, we introduce two new formulas (18) and (19), which are used to measure quality of each solution

among the current population. In the two formulas, ΔFF_d is fitness function deviation between fitness function of the d th point $FF(Point_d)$ and that of best point $FF(Point_{best})$ while ΔFF_{mean} is the fitness function deviation between mean fitness function of population FF_{mean} and fitness function of the best point $FF(Point_{best})$. We use the two objects to measure the quality of each considered solution wherein solution d with $\Delta FF_d > \Delta FF_{mean}$ is considered to be bad solution and solution d with another case is evaluated to be good solution. A strategy of finding new optimal solutions is plotted in Figure 2 by the presence of five current points (consisting of two good points in blue, two bad points in black, and the best point in red) and four newly updated points (two of bad points in yellow and two of good point in green). The figure shows that good points are newly updated around themselves while newly updated bad points can be found nearby the best point. Thus, search space around many good points is expanded while narrow space around the current best solution is also exploited strongly. The search strategy can explore global search zones and also exploit local search zones effectively and quickly. For simplicity of implementing the search technique, a pseudocode is shown in Algorithm 1 and

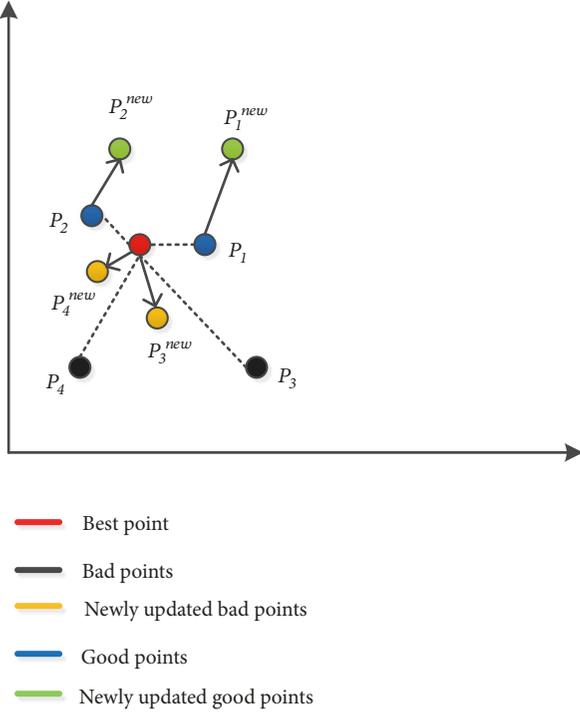


FIGURE 2: Strategy of searching new solutions of the proposed method.

the whole search of the proposed MSFS method is described in Figure 3.

$$\Delta FF_d = FF(\text{point}_d) - FF(\text{point}_{best}) \quad (18)$$

$$\Delta FF_{mean} = FF_{mean} - FF(\text{Point}_{best}) \quad (19)$$

Algorithm 1. New solution search strategy for the first and the second update processes of MSFS

For $d=1:N_{pop}$

If $\Delta FF_d > \Delta FF_{mean}$

$$\begin{aligned} \text{Point}_d^{\text{new}} &= \text{Point}_{best} + \varepsilon \\ &\times (\text{Point}_{r_1} - \text{Point}_{r_2} + \text{Point}_{r_3} - \text{Point}_{r_4}) \end{aligned} \quad (20)$$

else

$$\begin{aligned} \text{Point}_d^{\text{new}} &= \text{Point}_d + \varepsilon \\ &\times (\text{Point}_{r_1} - \text{Point}_{r_2} + \text{Point}_{r_3} - \text{Point}_{r_4}) \end{aligned} \quad (21)$$

end

end

4. The Application of the Proposed Method for Solving ORPD Problem

4.1. Selection of Control Variables and Producing a Set of Points. Similarly to optimal power flow problem, ORPD problem

is also supported by using Matpower program for obtaining the remaining variables in a considered transmission power network. Matpower needs input data consisting of active power of generators, which is given data, and a set of control variables such as generator voltage V_{Gi} , transformers tap setting T_i , and reactive power of capacitors Q_{ci} . Thus, control variables included in each point d (solution d) are as follows:

$$\begin{aligned} \text{Point}_d &= \{V_{G1,d}, \dots, V_{GN_G,d}, T_{1,d}, \dots, T_{N_t,d}, Q_{c1,d}, \dots, \\ &Q_{cN_c,d}\}^T; \quad d = 1, \dots, N_{pop} \end{aligned} \quad (22)$$

The first step is to randomly produce a set of points so that control variables must be within their lower limit and upper limit. The following model expresses initialization step.

$$\begin{aligned} \text{Point}_d &= CV_{min} + \text{rand.}(CV_{max} - CV_{min}); \\ &d = 1, \dots, N_{pop} \end{aligned} \quad (23)$$

where CV_{min} and CV_{max} are respectively lower limit and upper limit of control variables shown in (22).

4.2. Construction of Solution Quality Evaluation Function. Solution quality evaluation function is normally called fitness function when applying metaheuristic algorithms. Basically, the function covers two main terms that are objective function and penalty function wherein objective function is in charge of evaluating effectiveness of control variable regarding considered objective and penalty function undertakes evaluating violation of dependent variables. In problem formulation section, there are two equality constraints and six other inequality constraints; however, not all of these constraints are included in fitness function for evaluating quality. In fact, V_{Gi} , T_i together with Q_c are control variables, so they are always verified and corrected within their limits. On the other hand, formulas (5) and (6) are equality constraints, which have already been taken into account and solved by Matpower program. Thus, inequality constraints (8), (9), and (11) and equality constraints (5) and (6) are not terms in fitness function as a result. Predetermined active power of all generators and control variables are added to Matpower program for running, and then dependent variables consisting of reactive power of generator Q_{Gi} , voltage of load buses V_{loadi} , and apparent power flow of transmission lines S_l are determined in addition to three single objectives including power loss, total voltage deviation, and L-index. Finally, quality evaluation function of solution d or point d is expressed by

$$\begin{aligned} FF(\text{point}_d) &= \text{Objective} + K_Q \left(\sum_{i=1}^{N_G} \Delta Q_{Gi,d} \right) \\ &+ K_V \left(\sum_{i=1}^{N_{load}} \Delta V_{loadi,d} \right) \\ &+ K_S \left(\sum_{l=1}^{N_{line}} \Delta S_{l,d} \right) \end{aligned} \quad (24)$$

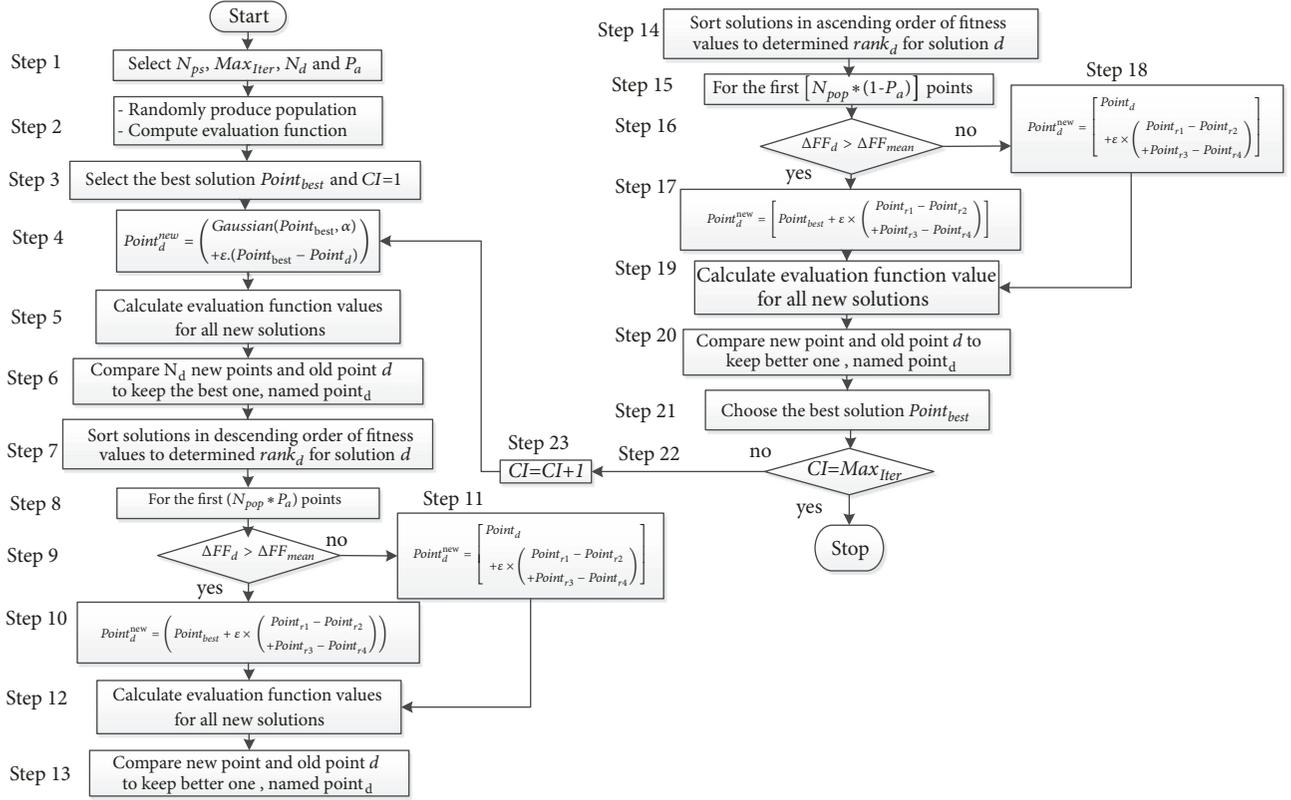


FIGURE 3: Flowchart of using MSFS for solving a typical optimization problem.

where K_Q , K_V , and K_S are penalty factors corresponding to violation of generator reactive power, voltage of load buses, and apparent power flow of transmission lines; $\Delta Q_{Gi,d}$, $\Delta V_{loadi,d}$, and $\Delta S_{l,d}$ are violation of reactive power of generator i , voltage of load bus i , and apparent power flow of transmission line l corresponding to solution d . Normally, the selection of penalty factors is based on experiment while violation objects are exactly determined by the set of following models:

$$\Delta Q_{Gi,d} = \begin{cases} 0 & \text{if } Q_{Gi,\min} \leq Q_{Gi,d} \leq Q_{Gi,\max} \\ Q_{Gi,\min} - Q_{Gi,d} & \text{if } Q_{Gi,d} < Q_{Gi,\min} \\ Q_{Gi,d} - Q_{Gi,\max} & \text{else} \end{cases} \quad (25)$$

$$\Delta V_{loadi,d} = \begin{cases} 0 & \text{if } V_{loadi,\min} \leq V_{loadi,d} \leq V_{loadi,\max} \\ V_{loadi,\min} - V_{loadi,d} & \text{if } V_{loadi,d} < V_{loadi,\min} \\ V_{loadi,d} - V_{loadi,\max} & \text{else} \end{cases} \quad (26)$$

$$\Delta S_{l,d} = \begin{cases} 0 & \text{if } S_{l,d} \leq S_{l,\max} \\ S_{l,d} - S_{l,\max} & \text{else} \end{cases} \quad (27)$$

4.3. Stopping Condition for Iterative Algorithm. The whole search process of the proposed method is stopped since

the predetermined maximum iteration number Max_{Iter} is reached. At the time, optimal solution for the current run is stored in addition to its evaluation function value. Similarly, after each 100 runs for optimizing one objective function, the best optimal solution is determined and then results are analyzed based on the best solution, the mean solution, the worst solution as well as standard deviation of all solutions.

4.4. The Entire Implementation of MSFS for ORPD Problem. The implementation of the proposed method for finding solutions of ORPD problem can be presented in the following.

Algorithm 2. The application of MSFS for solving ORPD problem:

Step 1. Determine control parameters consisting of population size N_{pop} , the number of diffusion N_d , the maximum number of iterations Max_{Iter} and P_a .

Step 2. Produce a set of initial points as shown in (23).

Step 3. Start running Matpower 4.1 for finding dependent variables.

Step 4. Calculate solution quality evaluation function value as (24)-(27).

(i) Determine solution with the lowest value and name it the best point $Point_{best}$.

TABLE 1: The selection of main control parameters for SSFS and the proposed method.

Power system	Objective	Max_{Iter}	N_{pop}	N_d	N_{nspr} of SSFS	N_{nspr} of MSFS
IEEE 30-bus system	Power loss	50	10	2	2,000	1,500
	Voltage deviation	100	10	2	4,000	3,000
	L-index	50	10	2	2,000	1,500
IEEE 118-bus system	Power loss	200	15	2	12,000	9,000
	Voltage deviation	200	15	2	12,000	9,000
	L-index	200	15	2	12,000	9,000

(ii) Start loop search algorithm by setting current iteration number CI to 1.

Step 5. Carry out diffusion process by running (17).

Step 6. Start running Matpower 4.1 for finding dependent variables.

Step 7. Calculate solution quality evaluation function value as (24)-(27).

(i) Compare such value of each old solution and that of new solutions to keep only one best solution.

Step 8. Sort all kept solutions in descending order of solution quality evaluation function value.

Step 9. Choose the first ($P_a * N_{pop}$) solutions.

(i) Carry out the first update process by using Algorithm 1 to produce new solutions of the selected solutions.

Step 10. Start running Matpower 4.1 for finding dependent variables.

Step 11. Calculate solution quality evaluation function value as (24)-(27).

(i) Compare such value of each old solution and that of its new sole solution to keep better one.

Step 12. Sort all kept solutions in ascending order of solution quality evaluation function value.

Step 13. Choose the first ($P_a * N_{pop}$) solutions.

(i) Carry out the second update process by using Algorithm 1 to produce new solutions of the selected solutions.

Step 14. Start running Matpower 4.1 for finding dependent variables.

Step 15. Calculate solution quality evaluation function value as (24)-(27).

(i) Compare such value of each old solution and that of new solution to keep better one.

Step 16. In all current solutions, determine one solution with the lowest solution quality evaluation function value and name it the best solution $Point_{best}$.

Step 17. Check stopping condition: if current iteration equals the maximum iteration number Max_{Iter} , stop iterative algorithm. Otherwise, back to step 5 and set $CI=CI+1$.

5. Numerical Results

In the section, we apply both SSFS and the proposed MSFS methods for finding solutions of two sample power systems, IEEE 30-bus system and IEEE 118-bus system with three different single objectives, power loss, voltage deviation, and L-index. The two methods are coded on Matlab 2016a and a personal laptop with 2.4 Ghz processor and 4 Gb of ram. For finding the best solution for each objective functions among three considered objectives, each method is run 100 and then results in terms of the best objective, the mean objective, the worst objective, and standard deviation and simulation time on average are also reported. Three main parameters consisting of the maximum number of iterations, the maximum number of diffusion, and the population size for each case are selected as shown in Table 1. Besides, the number of produced solutions per run (N_{nspr}) of SSFS and the proposed method is also reported in the table by using the following formulas:

$$N_{nspr} \text{ of SSFS} = (N_{pop} \times N_d + 2 \times N_{pop}) \times Max_{Iter} \quad (28)$$

$$N_{nspr} \text{ of MSFS} = (N_{pop} \times N_d + N_{pop}) \times Max_{Iter} \quad (29)$$

5.1. Testing Performance of SSFS and MSFS by Using Different Random Walks and Control Parameters. In the first subsection, we test performance of SSFS method when applying two different random walks consisting of Gaussian random walk and Levy random walk in which Gaussian random walk model was presented in (13) and Levy random walk was presented in [50]. Besides, we also perform performance

TABLE 2: Results obtained by the proposed method for IEEE 30-bus system by using different values of P_a .

Objective	P_a	0.8	0.6	0.4	0.2
P_{Loss}	Min.	4.5154	4.5143	4.524	4.5408
	Aver.	5.0158	5.0932	5.3127	5.185
	Max.	6.8004	6.721	7.2655	7.2549
	Std. dev.	0.5389	0.5259	0.7147	0.5179
TVD	Min.	0.0897	0.0874	0.0894	0.0894
	Aver.	0.1202	0.1106	0.114	0.113
	Max.	0.2214	0.1306	0.2382	0.2322
	Std. dev.	0.0338	0.012	0.0344	0.0294
L-index	Min.	0.12482	0.1244	0.12484	0.1253
	Aver.	0.1263	0.1267	0.12721	0.1267
	Max.	0.1274	0.1389	0.14318	0.128
	Std. dev.	0.0009	0.0016	0.00233	0.0007

TABLE 3: Comparisons of result obtained by the proposed method, SSFS-Gauss and SSFS-Levy for different objectives of IEEE 30-bus system.

Results	P_{Loss}			TVD			L-index		
	SSFS-Gauss	SSFS-Levy	MSFS	SSFS-Gauss	SSFS-Levy	MSFS	SSFS-Gauss	SSFS-Levy	MSFS
Min.	4.584	4.6619	4.5143	0.093	0.0974	0.0874	0.12484	0.1258	0.1244
Aver.	4.9846	4.9989	5.0932	0.1294	0.1214	0.1106	0.1270	0.1271	0.1267
Max.	5.6154	5.6399	6.721	0.2614	0.2544	0.1306	0.1372	0.141	0.1389
Std. dev.	0.2158	0.2025	0.5259	0.0374	0.0259	0.012	0.0017	0.002	0.0016
Cpu time (s)	4.1858	4.2047	3.2	8.0961	8.1008	6.3	4.2	4.3	3.1

investigation of the proposed MSFS method by using different P_a values. We set four P_a values consisting of 0.8, 0.6, 0.4, and 0.2.

For implementation of the survey, IEEE 30-bus system and IEEE 118-bus system, which can be reached by referring to [68], are employed. The first system consists of 6 generator buses, 24 load buses, and 41 transmission lines, 9 capacitor banks, and 4 transformers. The second system is constructed by more number of generators, load buses, transmission lines, capacitor banks, and transformers. The number of control variables for the two different systems is also different wherein 19 control variables are included for the first system including 6 voltage values of 6 generators, 9 reactive power output values of 9 shunt capacitor banks, and 4 tap setting values of 4 transformers, and 73 control variables are included for the second system including 54 voltage values of 54 generators, 14 reactive power output values of 14 shunt capacitor banks, and 9 tap setting values of 9 transformers.

5.1.1. Result Survey Obtained from IEEE 30-Bus System. The results obtained by the proposed method with the setting of different P_a are reported in Table 2. In addition, the results from the proposed method, SSFS with Gaussian random walk (SSFS-Gauss) and SSFS with Levy random walk, are also summarized in Table 3. Bold numbers, which are the best minimum, mean, maximum, and standard deviation values in Table 2, reveal that the proposed method can find the best optimal solutions for all three objectives by setting $P_a = 0.6$; however, the search stability of $P_a = 0.6$ is not the highest stable among four values since it cannot reach the

best average and standard deviation for power loss, and the best average, mean, maximum, and standard deviation for L-index objective. In spite of the limitation, the best P_a is still 0.6.

In Table 3, we also make the best minimum, average, maximum, and standard deviation numbers in bold for better observation and performance comparison. It is clearly visible that most bold numbers, especially all the best minimum values for three objectives, are belonging to the proposed MSFS method. Between two SSFS methods, SSFS with Gaussian random walk can get better results for approximately all three cases excluding standard deviation for power loss objective. The improvement level of best solutions obtained by the proposed method over SSFS-Gauss and SSFS-Levy are calculated by using (30) to be 1.52% and 3.17% for power loss optimization, 6.02% and 10.27% for voltage deviation optimization, and 0.35% and 1.11% for L-index optimization. Obviously, the improvement is significant and necessary for operating power system economically and stably because cost saving in \$ yearly is a very high amount, and better voltage stabilization can enable loads such as industrial zones and manufactures work effectively.

In addition, we also compare the worst and the second worst minimum numbers of the proposed method in Table 2 to the best minimum value of SSFS-Gauss. The worst minimums for power loss, voltage deviation, and L-index obtained by the proposed method are, respectively, 4.5408, 0.0897, and 0.1253 while those of SSFS-Gauss are, respectively, 4.584, 0.093, and 0.12484, and those of SSFS-Levy are, respectively, 4.6619, 0.0974, and 0.1258.

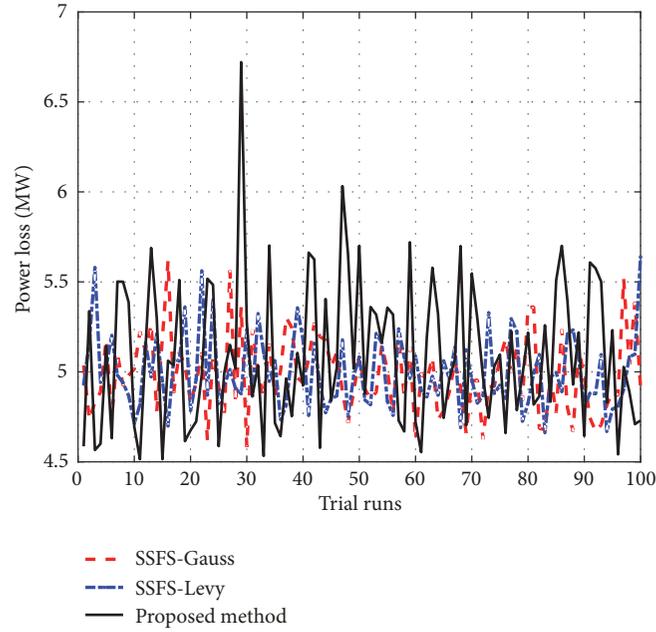


FIGURE 4: Comparisons of 100 runs for power loss objective of IEEE 30-bus system obtained by three methods.

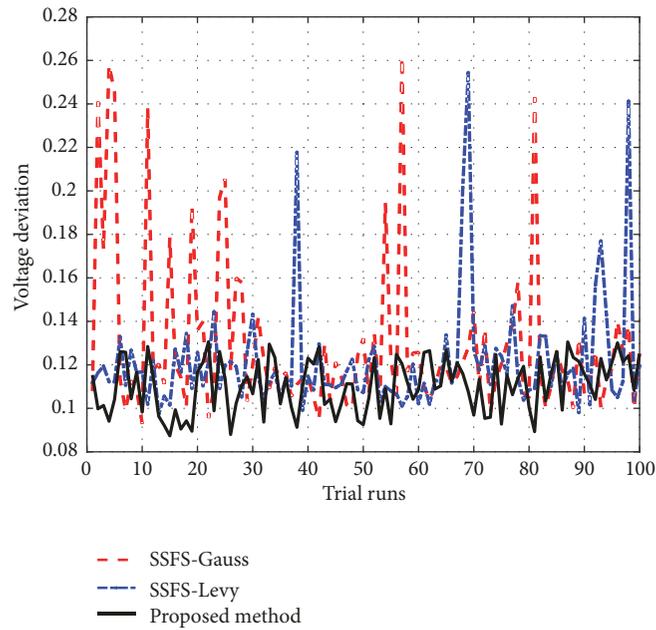


FIGURE 5: Comparisons of 100 runs for voltage deviation objective of IEEE 30-bus system obtained by three methods.

Clearly, the worst performance of the proposed method is also better than the best performance of SSFS-Levy for all cases and better than the best performance of SSFS-Gauss for power loss and voltage deviation although the worst minimum of L-index from the proposed method is higher than the minimum of SSFS-Gauss, but the second worst minimum of the proposed method is equal to such value of SSFS-Gauss. Furthermore, we also plot 100 values of quality solution evaluation function and optimal solution search

characteristic of the best run among 100 runs obtained by the three methods for the three single objectives in Figures 4–9. Figures 4, 5, and 6 show that the proposed method can find many solutions with lower evaluation function value than SSFS-Gauss and SSFS-Levy for all three objectives. For fluctuation comparison, the proposed method can get the best stabilization over the two SSFS methods only for voltage deviation objective, but it has high fluctuation for power loss and L-index objectives. The limits of MSFS can be easily

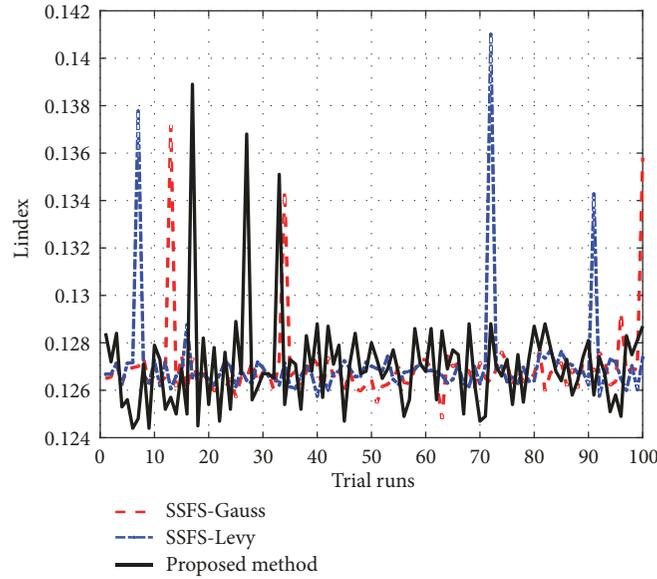


FIGURE 6: Comparisons of 100 runs for L-index objective of IEEE 30-bus system obtained by three methods.

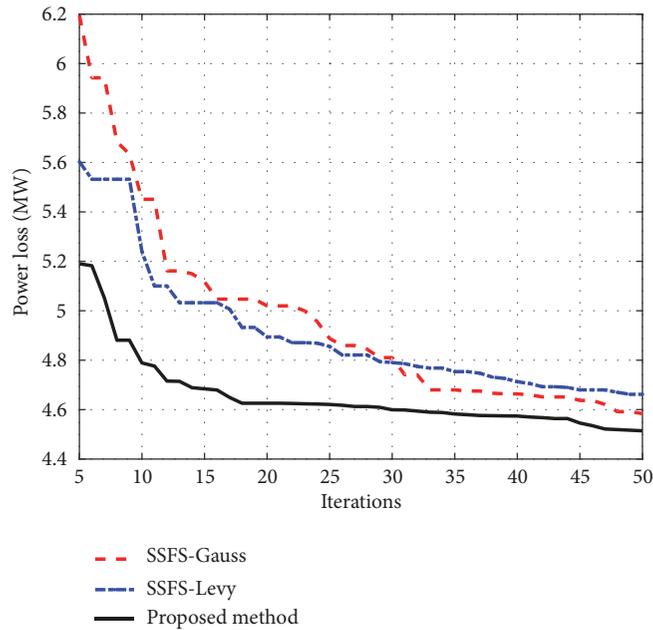


FIGURE 7: The best run obtained by three methods for power loss optimization of IEEE 30-bus system.

understood because it has used smaller N_{nspr} , which was shown in Table 1. In fact, SSFS methods have employed approximately $N_{nspr}=2,000$, but the proposed method has used $N_{nspr}=1,500$ for power loss and L-index optimization cases. It took the proposed method around 3 seconds but SSFS method around 4 seconds. On the other hand, the performance of applied methods is always evaluated by quality of the best optimal solutions and capability of finding global optimum or near global optimum. SSFS-Gauss can reach lower fluctuation than the proposed method, but it cannot

find near global optimum; it falls into local optimal zone and it hardly very jumps out the local zones for moving to global optimum or near global optimum. One more time, faster convergence ability to optimum of the proposed method compared to SSFS methods is also observed in Figures 7, 8, and 9.

In summary, the proposed method is superior to SSFS methods in terms of capability of finding higher quality solutions and capability of jumping out search zones with local optimum for moving to promising search space with

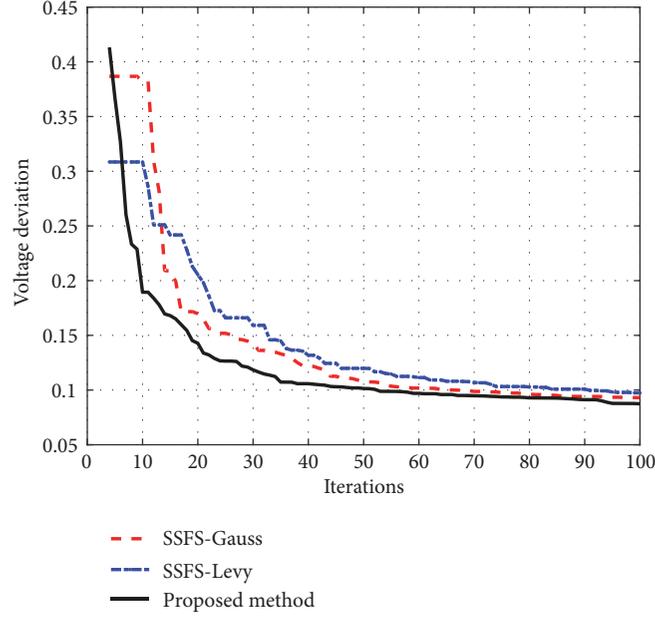


FIGURE 8: The best run obtained by three methods for voltage deviation optimization of IEEE 30-bus system.

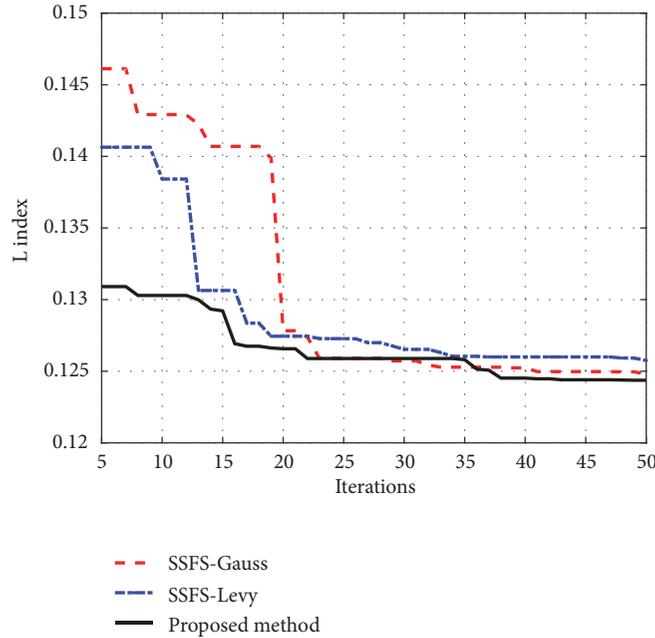


FIGURE 9: The best run obtained by three methods for L-index optimization of IEEE 30-bus system.

more effective solutions in addition to faster search ability. Thus, it can be indicated that the proposed modifications in the paper are reasonable for finding solutions of IEEE 30-bus system.

Improvement level (%)

$$= \frac{\text{minimum of compared method} - \text{minimum of MSFS}}{\text{minimum of compared method}} \quad (30)$$

· 100%

5.1.2. Result Survey Obtained from IEEE 118-Bus System. In order to execute the proposed method for the system, $P_a=0.6$, which was the best performance value for the first system above, is used. The results obtained by the three methods are reported in Table 4 for comparison. In the table, the best minimum, average, maximum, and standard deviation numbers are also remarked in bold for better observation. Bold numbers indicate that the proposed MSFS method can find the best optimal solutions with the lowest values of objective function for three optimization cases. Furthermore, the

TABLE 4: Comparisons of result obtained by the proposed method, SSFS-Gauss and SSFS-Levy for three different objectives of IEEE 118-bus system.

Results	P_{Loss}			TVD			L-index		
	SSFS-Gauss	SSFS-Levy	MSFS	SSFS-Gauss	SSFS-Levy	MSFS	SSFS-Gauss	SSFS-Levy	MSFS
Min.	118.5537	121.54876	114.6251	0.170952	0.197606	0.1486	0.06063	0.06065	0.06060
Aver.	121.42841	124.48965	115.4278	0.213546	0.253021	0.1604	0.06067	0.06072	0.06067
Max.	125.09423	127.07634	116.6677	0.278253	0.296876	0.1788	0.06077	0.06115	0.0611
Std. dev.	1.2841	1.1745	0.4678	0.0181	0.018	0.0068	0.00002	0.000059	0.00007
Cpu time (s)	82.4	85.4	63.7	83.6	86.2	62.8	84.1	86.2	62.7

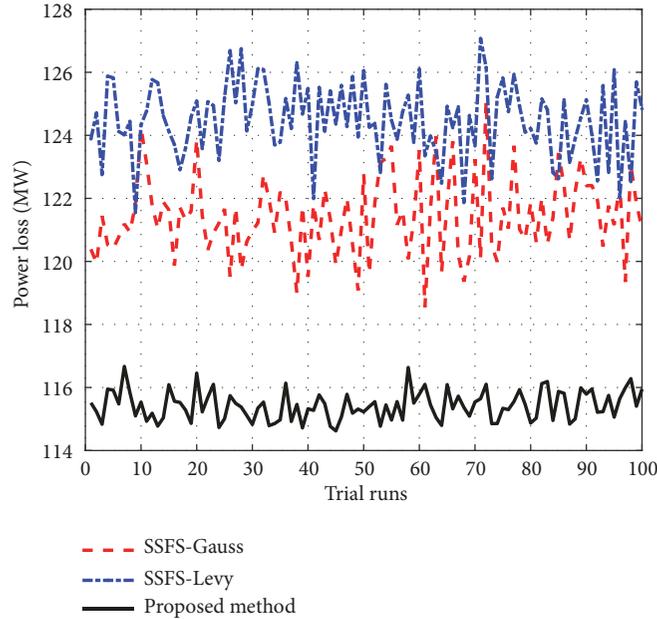


FIGURE 10: Comparisons of 100 runs for power loss objective of IEEE 118-bus system obtained by three methods.

proposed method also reaches the best average, maximum, and standard deviation for power loss optimization case and voltage deviation optimization case in addition to the best mean for L-index optimization case. Clearly, only for L-index optimization case, the proposed method gets the second best maximum and the worst standard deviation. Compared to SSFS-Gauss and SSFS-Levy, the minimum reduction of L-index from the proposed method is not much but that for power loss and voltage deviation is significant. Namely, the improvement level of the proposed method over SSFS-Gauss and SSFS-Levy are, respectively, 3.31% and 5.7% for power loss optimization, 13.08% and 24.8% for voltage deviation optimization, and 0.05% and 0.082% for L-index optimization. In addition to tables and numbers, Figures 10–15 are also used as evidence for proving the superiority of the proposed MSFS over SSFS methods. Unlike Figures 4–6, Figures 10–12 confirm outstanding performance of the proposed MSFS compared to SSFS methods. The figures show that all runs of the proposed method for power loss and voltage deviation optimization cases can find better solutions than both SSFS-Gauss and SSFS-Levy. Furthermore, the proposed method also copes with very low fluctuations for the two cases while that of SSFS methods is much worse. Although results of

the proposed method for L-index optimization shown in Figure 12 are not as good as those for two other cases, the proposed method can find many solutions with lower evaluation function value and its fluctuation is also smaller. Figures 13–15 can indicate that solutions of the proposed method at the 100th iteration are much more effective than those of SSFS methods at the last iteration, the 200th iteration. On the other hand, it is also noted that the proposed method used $N_{nspr}=9,000$, but SSFS methods used approximately $N_{nspr}=12,000$. The simulation execution time of the proposed method was around 60 seconds while that of SSFS method was around 80 seconds. It is clearly visible that the proposed method is much more effective and robust than SSFS methods for the IEEE 118-bus system.

Optimal solutions found by the proposed method are reported in Appendix.

5.2. Comparison of Results Obtained from IEEE 30-Bus Power System. In this subsection, the best results of the proposed MSFS method in terms of the best minimum, average, maximum and standard deviation in addition to other important parameters such as maximum number of iterations Max_{Iter} and number of newly produced solutions

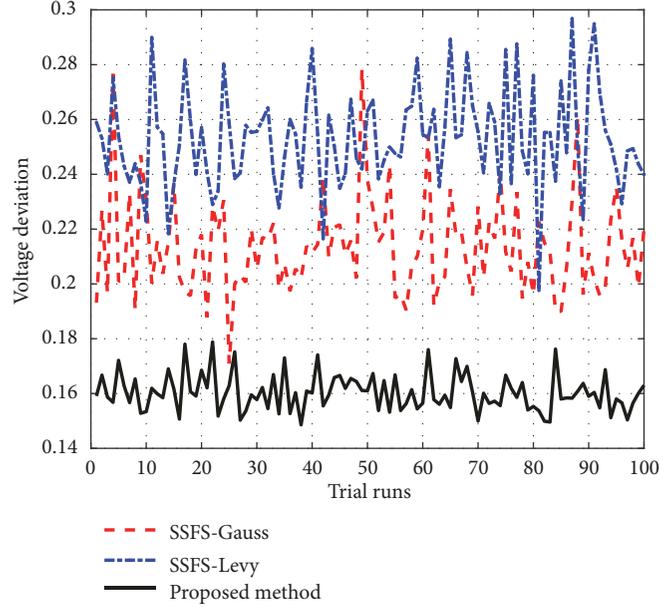


FIGURE 11: Comparisons of 100 runs for voltage deviation objective of IEEE 118-bus system obtained by three methods.

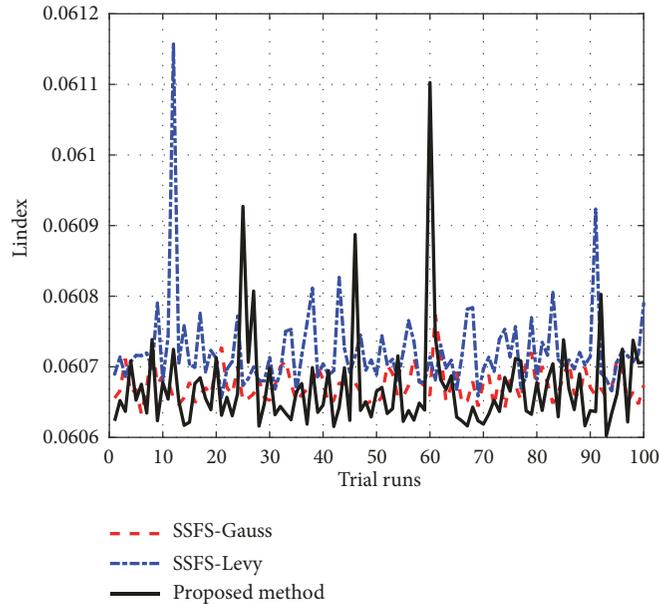


FIGURE 12: Comparisons of 100 runs for L-index objective of IEEE 118-bus system obtained by three methods.

per iteration N_{spi} are compared to other methods. Tables 5, 6, and 7 summarize comparisons of power loss, voltage deviation, and L-index obtained by different methods. For better comparison of the minimum, we also use (30) to calculate improvement level (ImLe) of the proposed method over other compared methods. For observation from ImLe, it can be seen that some values are represented by symbol “+” and some values are represented by symbol “-”. The two symbols have different meaning and different evaluation of the proposed method performance. Symbol “+” means that the proposed method has found better solution than such compared methods while symbol “-” points out that

such compared methods could find better solutions than the proposed method. However, there have been some cases reporting solutions with constraint violation or incorrect objective. Thus, for the case we substitute reported solutions of compared methods for running Matpower program and check solution as well as real objectives. In Table 5, the symbol “-” can be seen for comparison with FAHCLPSO [30] and MFOT [46] wherein the minimum of the two methods is respectively 4.4877 MW and 4.5128 MW corresponding to ImLe of -0.59% and -0.03%. We have substituted optimal solutions reported in [30, 46] to Matpower for running and obtaining all remaining variables. The recalculated power loss

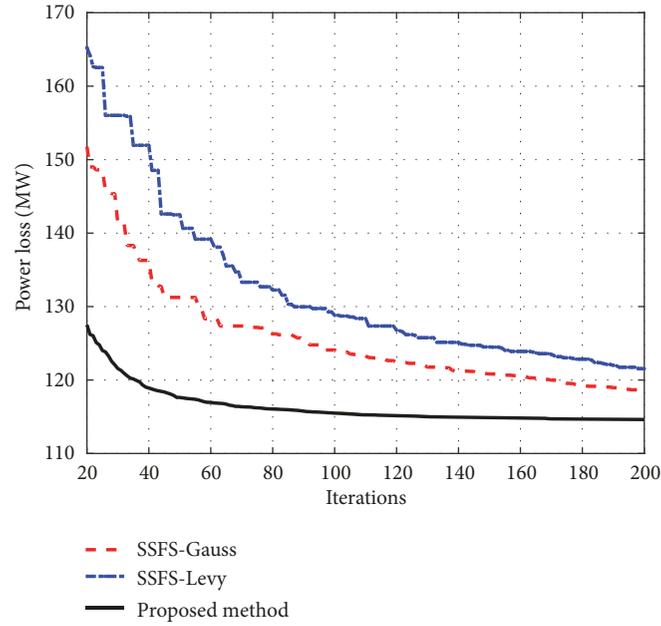


FIGURE 13: The best run obtained by three methods for power loss optimization of IEEE 118-bus system.

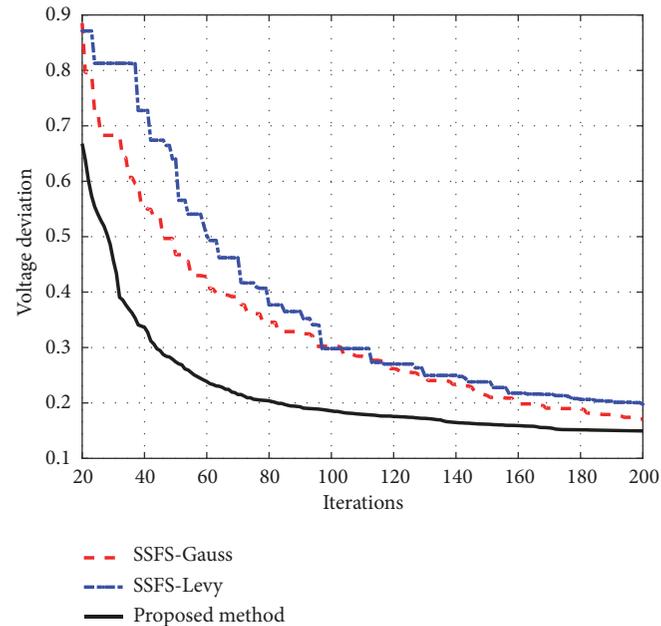


FIGURE 14: The best run obtained by three methods for voltage deviation optimization of IEEE 118-bus system.

of FAHCLPSO is 6.8372MW but that of MFOT is still 4.5128 MW. Thus, it indicates that the proposed method found better solution than FAHCLPSO but less effective optimal solution than MFOT for the case of minimizing power loss. However, MFOT is not the strongest method because it has used $Max_{Iter}=300$ and $N_{psi}=30$ while the proposed method has employed $Max_{Iter}=50$ and $N_{psi}=30$. Clearly, MFOT had to take approximately 6 times execution time of the proposed method. For further investigation of performance in comparison with MFOT, we also set $Max_{Iter}=300$ and $N_{psi}=30$

for running the proposed method and results obtained are also reported in the last row in the table. The minimum obtained by the setting is also 4.5128 MW equaling that of MFOT method. Thus, it can conclude that MFOT is not more effective than the proposed method for the case. As compared to other methods with symbol “+” of ImLe, it is clear that the number of such methods is high. They are GA variants, PSO variants, DE variants, TLBO variants, standard GSA, and standard ALO. The improvement level of the proposed method over these methods can be up to 7.37%. Furthermore,

TABLE 5: Comparison of results obtained from IEEE 30-bus system with power loss optimization.

Method	Min.	Aver.	Max.	Std. dev.	CPU time (s)	Max_{Iter}	N_{pop}	N_{spi}	ImLe (%)
EGA[17]	4.55	-	-	-	-	100	50	50	+0.78
DE[20]	4.555	-	-	-	-	500	150	150	+0.89
MDE[21]	4.87359	-	-	-	-	1500	15	15	+7.37
PSO-TVIW[28]	4.8458	4.8761	5.2292	0.0562	8.762	200	20	20	+6.84
PSO-TVAC[28]	4.8449	4.8702	4.9655	0.0263	8.68	200	20	20	+6.82
SPSO-TVAC[28]	4.5262	4.5564	4.7716	0.0554	9.092	200	20	20	+0.26
PSO-CF[28]	4.5258	4.5711	4.999	0.0815	8.48	200	20	20	+0.25
PGS-PSO[28]	4.6425	4.732	4.7972	0.1124	8.216	200	20	20	+2.76
SWT-PSO[28]	4.6578	4.9413	5.2521	0.1221	7.995	200	20	20	+3.08
PGSWT-PSO[28]	4.7914	5.2349	6.0512	0.2131	7.912	200	20	20	+5.78
IPGS-PSO[28]	4.5256	4.5508	4.9493	0.0592	7.852	200	20	20	+0.25
FAHCLPSO[30]	4.4877	4.6811	4.8732	0.1935	-	-	-	-	-0.59
GSA[32]	4.51431	-	-	-	-	200	100	100	0.00
GSA[34]	4.616657	-	-	-	-	200	-	-	+2.22
QOTLBO[36]	4.5594	4.5601	4.5617	0.037	-	100	50	100	+0.99
TLBO[36]	4.5629	4.5695	4.5748	0.0564	-	100	50	100	+1.07
ALO[45]	4.59	-	-	-	-	100	-	-	+1.65
MFOT[46]	4.5128	-	-	-	-	300	30	30	-0.03
SSO [64]	4.86029	5.2259	5.5975	0.19162	4.7	50	25	~30	+7.119
ISSO [64]	4.51445	4.69215	4.52546	0.02799	4.5	50	25	~30	+0.037
MSFS	4.5143	5.0932	6.721	0.5259	3.2	50	10	30	0
	4.5128	4.6928	5.7126	0.2268	15.6	300	10	30	-

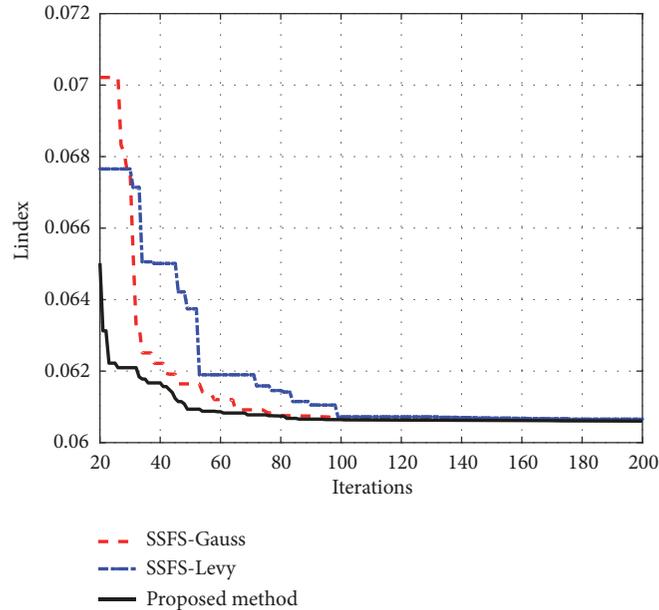


FIGURE 15: The best run obtained by three methods for L-index optimization of IEEE I18-bus system.

the proposed method is also the fastest tool because it uses the smallest values of $Max_{Iter} = 50$ and $N_{spi} = 30$ while those from others are much higher, especially the comparison with MDE in [21] using $Max_{Iter} = 1500$ and $N_{psi} = 15$, PSO variants in [28] using $Max_{Iter} = 200$ and $N_{psi} = 30$, and GSA in [32] using $Max_{Iter} = 200$ and $N_{psi} = 100$. In summary, for the case of

power loss optimization the proposed method can find better optimal solution than all methods and it is also the fastest optimization tool using the smallest number of iterations and the smallest population size.

For the voltage deviation minimization reported in Table 6, it should be noted that FAHCLPSO [30], GSA

TABLE 6: Comparison of results obtained from IEEE 30-bus system with voltage deviation optimization.

Method	Min.	Aver.	Max.	Std. dev.	CPU time (s)	Max_{Iter}	N_{pop}	N_{spi}	ImLe (%)
EGA[17]	0.0911	-	-	-	-	100	50	50	+4.06
DE[20]	0.0911	-	-	-	-	500	150	150	+4.06
MDE[21]	0.0882	-	-	-	-	1500	15	15	+0.91
PSO-TVIW[28]	0.1038	0.1597	0.5791	0.1112	12.25	200	20	20	+15.80
PSO-TVAC[28]	0.2064	0.2376	0.5796	0.0153	12.88	200	20	20	+57.66
SPSO-TVAC[28]	0.1354	0.1558	0.1833	0.0103	12.59	200	20	20	+35.45
PSO-CF[28]	0.1287	0.1557	0.4041	0.0404	12.94	200	20	20	+32.09
PGS-PSO[28]	0.1202	0.144	0.2593	0.0222	12.45	200	20	20	+27.29
SWT-PSO[28]	0.1614	0.1814	0.2296	0.133	22.57	200	20	20	+45.85
PGSWT-PSO[28]	0.1539	0.2189	0.5532	0.0656	22.32	200	20	20	+43.21
IPGS-PSO[28]	0.0892	0.1078	0.2518	0.0298	9.724	200	20	20	+2.02
FAHCLPSO[30]	0.04315	0.04705	0.05081	0.004	-	100	-	-	-102.55
GSA[32]	0.0676	-	-	-	-	200	100	100	-29.29
GSA[34]	0.1065	-	-	-	-	200	-	-	+17.93
QOTLBO[36]	0.0856	0.0872	0.0907	0.0314	-	100	50	100	-2.10
TLBO[36]	0.0913	0.0934	0.0988	0.0403	-	100	50	100	+4.27
MFOT[46]	0.12154	0.132961	0.15612	-	-	-	-	-	+28.09
SSO [64]	0.19304	0.2863	0.42681	0.05357	5.3	50	25	~30	+54.724
ISSO [64]	0.08847	0.11603	0.14938	0.02277	4.6	50	25	~30	+1.209
MSFS	0.0874	0.1106	0.1306	0.012	6.3	100	10	30	0

[32], and QOTLBO [36] have reported very good voltage deviation corresponding to ImLe of -102.55%, -29.29%, and -2.1%. However, as substituting their optimal solution for running Matpower, we have obtained 1.3442, 0.1862, and 0.1031 for FAHCLPSO, GSA, and QOTLBO, respectively. Clearly, the true values are much higher than real value of the proposed method, which is 0.0874. For comparison with other methods, the proposed method could find better results with faster manner. Namely, the improvement level of the proposed method over other ones can be from approximately 1% to 57% and its search process is very fast with $Max_{Iter}=100$ and $N_{psi}=30$, but those from others are 100 and 50, 1500 and 15, 200, and 20, 100 and 100. In summary, for voltage deviation objective, the proposed method is the best one for finding optimal solution with the fastest speed.

For comparison with L-index obtained by other methods shown in Table 7, the proposed method can find better optimal solution with improvement level up to +17.01% excluding the comparison with IPGS-PSO in [28], GSA in [32], QOTLBO in [36], and BA, GWO, ABC and ALO in [45]. For more exact comparison, we have also checked solutions and recalculated real L-index of these methods. As a result, L-index of optimal solutions reported by IPGS-PSO and QOTLBO in [36] remains unchanged and all dependent variables from IPGS-PSO are within lower and upper limits but reactive power of generator 4 of QOTLBO is 79.1274 MVAR, which is higher than upper limit of 60 MVAR. On the contrary, recalculated L-index obtained by other methods changes and is much higher than reported values. Namely, those of GSA, ALO, BA, GWO, and ABC are, respectively, 0.1247, 0.1244, 0.1264, 0.1257, and 0.1264. Clearly, only IPGS-PSO has better solution than the proposed method while

ALO in [45] has the same solution quality as the proposed method. However, IPGS-PSO and ALO have used very high values of Max_{Iter} and N_{spi} , namely, 200 and 20 for IPGS-PSO, and 100 and 40 for ALO. Clearly, ALO is slower than the proposed method for converging to the solution with the same quality. For further comparison with IPGS-PSO in [28], we set Max_{Iter} and N_{spi} to 130 and 30 for running the proposed method and obtained results reported in the last row of Table 7 indicate that the proposed method can find approximate quality solution with IPGS-PSO but mean and standard deviation from the proposed method are lower than those from IPGS-PSO.

In summary, the proposed method can find better results than most methods for the case excluding IPGS-PSO with better solution and ALO with the same solution quality, but the proposed method has faster convergence than all methods excluding IPGS-PSO. Thus, we can conclude that the proposed method is very effective for the case of L-index optimization.

5.3. Comparison of Result Obtained from IEEE 118-Bus Power System. In comparison with other methods for IEEE 118-bus system, obtained results for finding solutions of power loss optimization, voltage deviation and L-index are included in Tables 8, 9, and 10, respectively. The obtained result indicates that the proposed method can find better results than most methods but its results are also worse than several ones. Namely, for power loss optimization reported in Table 8, the proposed method has lower minimum, mean, and maximum power loss than two PSO methods in [27] and eight PSO methods in [28], TLBO in [36], HEP in [65] and MFOT in [46] corresponding to the improvement level from 0.04 to

TABLE 7: Comparison of results obtained from IEEE 30-bus system with voltage stabilization index optimization.

Method	Min.	Aver.	Max.	Std. dev.	CPU time (s)	Max_{Iter}	N_{pop}	N_{spi}	ImLe (%)
EGA [17]	0.1246	-	-	-	-	100	50	50	+0.16
DE [20]	0.1246	-	-	-	-	500	50	50	+0.16
PSO-TVIW [28]	0.1258	0.127	0.1289	0.0008	14.42	200	20	20	+1.11
PSO-TVAC [28]	0.1499	0.1513	0.1544	0.0009	14.53	200	20	20	+17.01
SPSO-TVAC [28]	0.1271	0.1285	0.1297	0.0006	14.05	200	20	20	+2.12
PSO-CF [28]	0.1261	0.1279	0.1295	0.0008	14.39	200	20	20	+1.35
PGS-PSO [28]	0.1264	0.1297	0.1313	0.0008	14.84	200	20	20	+1.58
SWT-PSO [28]	0.1488	0.1634	0.1806	0.0074	18.99	200	20	20	+16.40
PGSWT-PSO [28]	0.1394	0.1537	0.1749	0.0081	19.107	200	20	20	+10.76
IPGS-PSO [28]	0.1241	0.1266	0.1298	0.001	13.75	200	20	20	-0.24
GSA [32]	0.1161	-	-	-	-	200	100	100	-7.15
QOTLBO [36]	0.1242	0.1245	0.1247	0.0452	-	100	50	100	-0.16
TLBO [36]	0.1252	0.1254	0.1258	0.0454	-	100	50	100	+0.64
BA [45]	0.1191	-	-	-	94.65	100	40	80	-4.45
GWO [45]	0.118	-	-	-	104.29	100	40	40	-5.42
ABC [45]	0.1161	-	-	-	105.04	100	40	40	-7.15
ALO [45]	0.1161	-	-	-	97.92	100	40	40	-7.15
SSO [64]	0.12738	0.1338	0.1446	0.00384	5.6	50	25	~30	+2.34
ISSO [64]	0.12466	0.1253	0.1271	0.0004	4.9	50	25	~30	+0.37
MSFS	0.1244	0.1267	0.1389	0.0016	3.1	50	10	30	0
	0.1242	0.1257	0.1314	0.0007	7.5	130	10	30	-

TABLE 8: Comparison of results obtained from IEEE 118-bus system with power loss optimization.

Method	Min.	Aver.	Max.	Std. dev.	CPU time (s)	Max_{Iter}	N_{pop}	N_{spi}	ImLe (%)
CLPSO[27]	130.96	-	-	-	-	200	120	120	+0.12
PSO[27]	131.99	-	-	-	-	200	120	120	+0.13
PSO-TVIW[28]	116.8976	118.2344	126.6222	1.6009	109.645	200	40	40	+0.02
PSO-TVAC[28]	124.3335	129.7494	134.1254	2.156	96.32	200	40	40	+0.08
SPSO-TVAC[28]	116.2026	117.3553	118.139	0.4696	96.45	200	40	40	+0.01
PSO-CF[28]	115.6469	116.9863	119.8378	0.8655	95.86	200	40	40	+0.01
PGS-PSO[28]	116.6075	119.3968	127.0772	2.107	96.11	200	40	40	+0.02
SWT-PSO[28]	124.1476	129.371	141.6147	3.309	91.58	200	40	40	+0.08
PGSWT-PSO [28]	119.427	122.781	125.762	1.2455	95.17	200	40	40	+0.04
IPGS-PSO[28]	115.06	116.462	118.35	0.528	91.07	200	40	40	+0.004
QOTLBO[36]	112.2789	113.7693	115.4516	0.0244	-	100	50	100	-0.02
TLBO[36]	116.4003	118.4427	121.3902	0.0482	-	100	50	100	+0.02
SARCGA[65]	113.12	113.968	-	0.0002	-	300	15	15	-0.01
HEP[65]	115.58	115.8	-	0.0018	-	300	15	15	+0.01
MFOT[46]	116.4254	123.9207	135.96	-	-	1,000	30	30	+0.02
MSFS	114.6251	115.4278	116.6677	0.4678	63.7	200	15	45	0

0.13%. But the proposed method has higher power loss than QOTLBO in [36] and SARCGA in [65] corresponding to the ImLe of -0.02% and -0.01%. The verification of available solutions shows that QOTLBO in [36] reported correct power loss and valid solution, but there was no solution given in [65] for SARCGA. For the case of optimizing voltage deviation in Table 9, the proposed method is greatly superior to all methods including ten PSO methods in both [27, 28], and TLBO and QOTLBO in [36] corresponding to the ImLe from 0.1% to 0.93%.

For L-index comparison in Table 10, the proposed method has better optimal solution quality than two PSO methods in [27], four PSO methods in [28], and two TLBO methods in [36], but it gets worse solutions than three other PSO methods in [28] and approximate solution with another PSO method in [28]. The best performance of the proposed method is more effective than the worst method with the improvement level of 0.57% while the best method IGP-PSO can improve result over the proposed method only with 0.07%. For comparison with other methods in terms of convergence speed, it can be

TABLE 9: Comparison of results obtained from IEEE 118-bus system with TVD optimization.

Method	Min.	Aver.	Max.	Std. dev.	CPU time (s)	Max_{Iter}	N_{pop}	N_{spi}	ImLe (%)
CLPSO[27]	1.6177	-	-	-	-	200	120	120	+0.91
PSO[27]	2.2359	-	-	-	-	200	120	120	+0.93
PSO-TVIW[28]	0.1935	0.2291	0.2809	0.0206	78.49	200	40	40	+0.23
PSO-TVAC[28]	0.3921	0.4724	0.5407	0.0316	78.7	200	40	40	+0.62
SPSO-TVAC[28]	0.2074	0.2498	0.3012	0.0215	74.9	200	40	40	+0.28
PSO-CF[28]	0.1801	0.2143	0.3384	0.0286	78.13	200	40	40	+0.17
PGS-PSO[28]	0.1658	0.2084	0.2354	0.021	51.24	200	40	40	+0.10
SWT-PSO[28]	0.1658	0.2084	0.2354	0.021	51.24	200	40	40	+0.10
PGSWT-PSO[28]	0.2355	0.2755	0.3239	0.0205	114.5	200	40	40	+0.37
IPGS-PSO[28]	0.162	0.1923	0.2147	0.0184	47.86	200	40	40	+0.08
QOTLBO[36]	0.191	0.2043	0.2267	0.0356	-	100	50	100	+0.22
TLBO[36]	0.2237	0.2306	0.2543	0.0384	-	100	50	100	+0.34
MSFS	0.1486	0.1604	0.1788	0.0068	62.8	200	15	45	0

TABLE 10: Comparison of results obtained from IEEE 118-bus system with L-index optimization.

Method	Min.	Aver.	Max.	Std. dev.	CPU time (s)	Max_{Iter}	N_{pop}	N_{spi}	ImLe (%)
CLPSO[27]	0.0965	-	-	-	-	200	120	120	+0.37
PSO[27]	0.1388	-	-	-	-	200	120	120	+0.56
PSO-TVIW[28]	0.0606	0.0607	0.0612	0.0001	119.66	200	40	40	+0.00
PSO-TVAC[28]	0.0607	0.0609	0.0613	0.0001	119.22	200	40	40	+0.00
SPSO-TVAC[28]	0.0607	0.0608	0.0612	0.0001	119.16	200	40	40	+0.00
PSO-CF[28]	0.0606	0.0607	0.061	0.0001	119.86	200	40	40	+0.00
PGS-PSO[28]	0.0654	0.0656	0.0721	0.0002	119.65	200	40	40	+0.07
SWT-PSO[28]	0.0587	0.0608	0.0641	0.0012	58.45	200	40	40	-0.03
PGSWT-PSO[28]	0.0574	0.0605	0.0683	0.0018	56.43	200	40	40	-0.06
IPGS-PSO[28]	0.0568	0.0569	0.0583	0.0002	55.62	200	40	40	-0.07
QOTLBO[36]	0.0608	0.0616	0.0631	0.0476	-	100	50	100	+0.00
TLBO[36]	0.0613	0.0626	0.0646	0.0488	-	100	50	100	0.01
MSFS	0.0606	0.0607	0.0611	0.0001	62.7	200	15	45	0

noted that the proposed method is always faster than two PSO methods in [27] for all cases, MFOT in [46] for power loss optimization case and two TLBO methods in [36] for the last two cases because the proposed method has used less values of control parameters than these methods, but it could get better results than these methods. Namely, the numbers of iterations and produced solutions per iteration of the proposed method are 200 and 45 but those are 200 and 120 for methods in [27], 1,000 and 30 for MFOT in [46], and 100 and 100 for TLBO methods in [36]. For comparison with methods in [28], the proposed method has used equal number of iterations but slightly higher number of produced solutions, namely, 45 compared to 40 but it can improve results significantly for power loss and voltage deviation optimization cases and worse result for L-index optimization. Thus, the proposed method is slower to be convergent to high quality solution than three PSO methods in [28] for L-index optimization case.

In summary, the proposed method can find better results than approximately all methods excluding a few ones and it

also converges to better results faster by using smaller number of ($Max_{Iter} * N_{spi}$). Thus, it can conclude that the proposed method is very efficient for searching optimal solutions of ORPD problem. Optimal solutions corresponding to their objective are given in Appendix.

6. Conclusions

This paper has proposed an effective improved version of standard stochastic fractal search algorithm for seeking optimal solutions of ORPD problem considering two standard IEEE systems as study cases. The proposed MSFS method is a metaheuristics based powerful optimization search tool with applying modifications on three generations of producing new solutions in the iterative technique. In the first generation, the proposed method has chosen only one most effective formula among two available ones for producing high quality solutions. Similarly, the second and the third generations have also been carried out by keeping a more effective formula and abandoning a less effective formula. In addition, the

TABLE 11: The information of IEEE 30-bus system and IEEE 118-bus system.

Power system	IEEE 30-bus system	IEEE 118-bus system
Number of transmission lines	41	186
Number of generator buses	6	54
Number of transformers	4	9
Number of capacitor banks	9	14
Number of control variables	19	77
Total active power of loads (MW)	283.4	4,242
Total reactive power of loads (MVAR)	126.2	1,438
Total active power of generators (MW)	288.67	132.863

TABLE 12: Setting limits for IEEE 30-bus system.

Limits of capacitor banks (MVAR)			Voltage limits of generator buses (Pu)		Reactive power limits of generators (MVAR)		
Bus	$Q_{ci,min}$	$Q_{ci,max}$	$V_{G,min}$	$V_{G,max}$	Bus	$Q_{Gi,min}$	$Q_{Gi,max}$
10	0	5	0.95	1.10	1	-20	200
12	0	5	Voltage limits of load buses (Pu)		2	-20	100
15	0	5	$V_{load,min}$	$V_{load,max}$	13	-15	80
17	0	5	0.95	1.05	22	-15	60
20	0	5	Transformer tap setting limits (Pu)		23	-10	50
21	0	5	T_{min}	T_{max}	27	-15	60
23	0	5	0.9	1.10			
24	0	5					
29	0	5					

kept formula in the last generations has been modified with intent to exploit performance of highly efficient space and avoid spending time searching ineffective zones. By using the proposed modification, the proposed method could reduce a number of computations such as canceling three formulas in three generations, canceling selection of parameter “walk” and comparison steps between random number and 0.5 in the second and the third generations. On the other hand, we have also tried the performance of Gaussian random walk and Levy random walk for SSFS, and we have concluded that the proposed method needed Gaussian random walk rather Levy random walk. The obtained result comparisons with SSFS have given persuasive evidences for conclusion that the proposed method could find much better solutions, owned more stable search capability and converged to better results with faster speed. Further evaluation of the proposed method has been done by comparing with a high number of existing methods including PSO family, DE family, GA family, TLBO family, and other recent methods. Comparisons have indicated that the proposed method could find solutions with less power loss, voltage deviation, and L-index than those from nearly all methods. Furthermore, search speed of the proposed method was also faster than all methods excluding a few improved versions of PSO for L-index optimization case. As a result, it can be pointed out that the proposed method deserves to be applied for ORPD problem as one of the most effective optimization tools.

TABLE 13: Setting limits for IEEE 118-bus system.

Limits of Capacitor banks (MVAR)			Reactive power limits of generators (MVAR)	
Bus	$Q_{ci,min}$	$Q_{ci,max}$	$Q_{Gi,min}$	$Q_{Gi,max}$
5	-40	0	-500	500
34	0	14	Voltage limits of generator buses (Pu)	
37	25	0	$V_{G,min}$	$V_{G,max}$
44	0	10	0.95	1.1
45	0	10	Voltage limits of load buses (Pu)	
46	0	10	$V_{load,min}$	$V_{load,max}$
48	0	15	0.95	1.05
74	0	12	Transformer tap setting limits (Pu)	
79	0	20	T_{min}	T_{max}
82	0	20	0.9	1.1
83	0	10		
105	0	20		
107	0	6		
110	0	6		

Appendix

See Tables 11–17.

TABLE 14: Optimal solutions obtained by the proposed method for IEEE 30-bus system.

Control Variables	Power loss objective		Vol. dev. objective	L-index objective	
V_{G1} (pu)	1.1000	1.1000	1.0097	1.1000	1.0999
V_{G2}	1.0939	1.0943	1.0042	1.0918	1.0958
V_{G5}	1.0739	1.0747	1.0182	1.0968	1.1000
V_{G8}	1.0754	1.0765	1.0099	1.0905	1.0920
V_{G11}	1.1000	1.1000	1.0070	1.1000	1.1000
V_{G13}	1.1000	1.1000	1.0150	1.0962	1.1000
Q_{C10} (MVAR)	5.0000	5.0000	5.0000	5.0000	5.0000
Q_{C12}	5.0000	5.0000	4.7776	4.9371	0
Q_{C15}	5.0000	4.8050	5.0000	0.0000	5.0000
Q_{C17}	5.0000	5.0000	0.0222	4.7321	4.9993
Q_{C20}	5.0000	4.0091	4.9976	1.5651	0
Q_{C21}	4.9990	4.9999	4.9986	5.0000	3.2232
Q_{C23}	2.3155	2.5889	4.9997	5.0000	0
Q_{C24}	5.0000	5.0000	4.9699	0.0000	0.0005
Q_{C29}	2.0180	2.1893	2.9033	0.0000	0.0061
T_{11}	1.0473	1.0425	1.0218	0.9835	1.0036
T_{12}	0.9000	0.9000	0.9000	0.9439	0.9014
T_{15}	0.9790	0.9793	0.9973	0.9680	0.9652
T_{36}	0.9634	0.9647	0.9698	0.9557	0.9552
<i>Objective function</i>	<i>4.5143</i>	<i>4.5128</i>	<i>0.0874</i>	<i>0.1244</i>	<i>0.1242</i>

TABLE 15: Optimal solution obtained by MSFS for the IEEE 118-bus power system with minimization of power loss.

Control variables	Value	Control variables	Value
V_{G1} (pu)	1.0276	V_{G90}	1.0433
V_{G4}	1.0488	V_{G91}	1.0466
V_{G6}	1.0402	V_{G92}	1.0562
V_{G8}	1.0300	V_{G99}	1.0472
V_{G10}	1.0482	V_{G100}	1.0510
V_{G12}	1.0372	V_{G103}	1.0401
V_{G15}	1.0331	V_{G104}	1.0293
V_{G18}	1.0355	V_{G105}	1.0261
V_{G19}	1.0327	V_{G107}	1.0134
V_{G24}	1.0441	V_{G110}	1.0219
V_{G25}	1.0779	V_{G111}	1.0305
V_{G26}	1.0927	V_{G112}	1.0057
V_{G27}	1.0384	V_{G113}	1.0421
V_{G31}	1.0304	V_{G116}	1.0497
V_{G32}	1.0332	Q_{C5} (MVAR)	-14.2920
V_{G34}	1.0433	Q_{C34}	12.1356
V_{G36}	1.0403	Q_{C37}	-12.0608
V_{G40}	1.0227	Q_{C44}	9.9086
V_{G42}	1.0271	Q_{C45}	9.9938
V_{G46}	1.0408	Q_{C46}	10.0000
V_{G49}	1.0556	Q_{C48}	6.7524
V_{G54}	1.0349	Q_{C74}	3.0938
V_{G55}	1.0336	Q_{C79}	19.9643
V_{G56}	1.0340	Q_{C82}	19.9731
V_{G59}	1.0565	Q_{C83}	9.9826
V_{G61}	1.0516	Q_{C105}	1.0080
V_{G62}	1.0432	Q_{C107}	2.0549
V_{G65}	1.0592	Q_{C110}	0.8772

TABLE 15: Continued.

Control variables	Value	Control variables	Value
V_{G66}	1.0669	T_8 (pu)	0.9741
V_{G69}	1.0686	T_{32}	1.0910
V_{G70}	1.0316	T_{36}	0.9861
V_{G72}	1.0306	T_{51}	0.9762
V_{G73}	1.0290	T_{93}	0.9748
V_{G74}	1.0257	T_{95}	0.9952
V_{G76}	1.0247	T_{102}	0.9115
V_{G77}	1.0462	T_{107}	0.9392
V_{G80}	1.0584	T_{127}	0.9767
V_{G85}	1.0504		
V_{G87}	1.0393		
V_{G89}	1.0642		

Power loss = 114.6251 MW

TABLE 16: Optimal solution obtained by MSFS for the IEEE 118-bus power system with minimization of voltage deviation.

Control Variables	Value	Control Variables	Value
V_{G1} (pu)	1.0037	V_{G90}	1.0089
V_{G4}	1.0179	V_{G91}	1.0951
V_{G6}	0.9942	V_{G92}	0.9584
V_{G8}	1.0160	V_{G99}	1.0087
V_{G10}	0.9657	V_{G100}	0.9679
V_{G12}	1.0110	V_{G103}	1.0296
V_{G15}	1.0110	V_{G104}	0.9648
V_{G18}	1.0247	V_{G105}	1.0869
V_{G19}	1.0351	V_{G107}	0.9950
V_{G24}	0.9932	V_{G110}	1.0412
V_{G25}	1.0117	V_{G111}	1.0089
V_{G26}	0.9733	V_{G112}	1.0539
V_{G27}	1.0068	V_{G113}	1.0928
V_{G31}	1.0038	V_{G116}	0.9515
V_{G32}	1.0036	Q_{C5} (MVAR)	0.9954
V_{G34}	1.0024	Q_{C34}	-32.2857
V_{G36}	1.0006	Q_{C37}	8.0559
V_{G40}	1.0086	Q_{C44}	-23.4580
V_{G42}	1.0020	Q_{C45}	9.8744
V_{G46}	1.0447	Q_{C46}	9.9382
V_{G49}	0.9996	Q_{C48}	8.1557
V_{G54}	1.0237	Q_{C74}	0.2848
V_{G55}	1.0453	Q_{C79}	7.5632
V_{G56}	1.0165	Q_{C82}	3.9702
V_{G59}	1.0946	Q_{C83}	19.9644
V_{G61}	0.9921	Q_{C105}	9.8840
V_{G62}	1.0055	Q_{C107}	11.6216
V_{G65}	0.9889	Q_{C110}	1.5550
V_{G66}	1.0075	T_8 (pu)	1.1667
V_{G69}	0.9517	T_{32}	1.0868
V_{G70}	0.9752	T_{36}	1.0957
V_{G72}	1.0106	T_{51}	0.9865
V_{G73}	1.0280	T_{93}	0.9915
V_{G74}	1.0337	T_{95}	0.9177
V_{G76}	1.0037	T_{102}	1.0140
V_{G77}	1.0134	T_{107}	0.9088
V_{G80}	1.0109	T_{127}	1.0955
V_{G85}	1.0155		
V_{G87}	1.0092		
V_{G89}	1.0091		

Voltage deviation= 0.1486 pu

TABLE 17: Optimal solution obtained by MSFS for the IEEE 118-bus power system with minimization of L-index.

Control Variables	Value	Control Variables	Value
V_{G1} (pu)	0.9943	V_{G90}	0.9514
V_{G4}	1.0404	V_{G91}	1.0320
V_{G6}	1.0141	V_{G92}	1.0335
V_{G8}	1.0224	V_{G99}	0.9886
V_{G10}	1.0286	V_{G100}	1.0168
V_{G12}	1.0243	V_{G103}	1.0488
V_{G15}	1.0281	V_{G104}	1.1000
V_{G18}	1.0111	V_{G105}	0.9731
V_{G19}	1.0206	V_{G107}	1.0008
V_{G24}	1.0375	V_{G110}	0.9506
V_{G25}	1.0229	V_{G111}	0.9948
V_{G26}	0.9939	V_{G112}	1.0011
V_{G27}	1.0151	V_{G113}	0.9538
V_{G31}	1.0013	V_{G116}	1.0366
V_{G32}	1.0072	Q_{C5} (MVAR)	1.0478
V_{G34}	1.0596	Q_{C34}	-10.3586
V_{G36}	1.0289	Q_{C37}	4.8660
V_{G40}	0.9805	Q_{C44}	-16.2333
V_{G42}	1.0996	Q_{C45}	7.1117
V_{G46}	1.1000	Q_{C46}	3.9834
V_{G49}	1.0444	Q_{C48}	0.0266
V_{G54}	1.0135	Q_{C74}	0.0021
V_{G55}	1.0783	Q_{C79}	4.6679
V_{G56}	1.0549	Q_{C82}	19.5441
V_{G59}	0.9501	Q_{C83}	3.6342
V_{G61}	0.9694	Q_{C105}	9.8996
V_{G62}	1.0340	Q_{C107}	5.4045
V_{G65}	1.0719	Q_{C110}	0.1113
V_{G66}	0.9506	T_8 (pu)	0.0068
V_{G69}	0.9505	T_{32}	1.0000
V_{G70}	1.0449	T_{36}	1.0981
V_{G72}	1.0240	T_{51}	0.9516
V_{G73}	1.0137	T_{93}	1.0033
V_{G74}	1.0998	T_{95}	1.0602
V_{G76}	0.9943	T_{102}	0.9892
V_{G77}	0.9983	T_{107}	1.0547
V_{G80}	0.9612	T_{127}	1.0218
V_{G85}	0.9873		
V_{G87}	1.0066		
V_{G89}	0.9500		

$L\text{-index} = 0.0606 \text{ pu}$

Nomenclature

$1-P_a$: The fraction of the best solutions selected in the second update
 Max_{Iter} : Maximum number of iterations
 N_{bus} : Number of buses in the considered power system
 N_c : Number of buses with the presence of capacitor banks
 N_d : Maximum number of diffusion

N_G : Number of generators with the presence of generators
 N_{line} : Number of transmission branches
 N_{load} : Number of load buses
 N_{pop} : Population size or a set of points
 N_{spi} : Number of new solutions produced per iteration
 N_t : Number of buses with transformer
 P_a : The fraction of the worst solutions selected in the first update

P_{di}, Q_{di} :	Active and reactive power of load installed at bus i , respectively
P_{Loss} :	Total power loss in all transmission branches
$Point_1, Point_2, Point_3, Point_4$:	Randomly picked points in the current population
Q_{ci} :	Reactive power output of capacitor banks installed at bus i
$Q_{ci,min}, Q_{ci,max}$:	Lower and upper reactive power output limitations of capacitor bank installed at bus i , respectively
$Q_{Gi,min}, Q_{Gi,max}$:	Lower and upper reactive power output limitations of generator at bus i , respectively
$rank_d$:	The rank of solution d
$S_{l,max}$:	Maximum apparent power flow of branch l
T_{min}, T_{max} :	Lower and upper tap changer limitations of all transformers, respectively
TVD:	Total voltage deviation of all load buses
$V_{G,min}, V_{G,max}$:	Lower and upper voltage magnitude limitations of all generators, respectively
V_i, V_j :	Working voltage magnitude of buses i and j , respectively
$V_{load,min}, V_{load,max}$:	Lower and upper voltage magnitude limitations of all loads, respectively
λ_d, β_d :	Two random number within 0 and 1 of $point_d$, respectively
G_{linel}, B_{linel} :	Real and imaginary terms of admittance of branch l , which is connecting buses i and j , respectively
φ_i, φ_j :	Angles of voltage at buses and j , respectively.

Data Availability

No data were used to support this study. All data are provided in full in the results section of this paper.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] W. Villa-Acevedo, J. López-Lezama, and J. Valencia-Velásquez, "A novel constraint handling approach for the optimal reactive power dispatch problem," *Energies*, vol. 11, no. 9, p. 2352, 2018.
- [2] T. Nguyen, D. Vo, N. Vu Quynh, and L. Van Dai, "Modified cuckoo search algorithm: a novel method to minimize the fuel cost," *Energies*, vol. 11, no. 6, p. 1328, 2018.
- [3] D. C. Yu, J. E. Fagan, B. Foote, and A. A. Aly, "An optimal load flow study by the generalized reduced gradient approach," *Electric Power Systems Research*, vol. 10, no. 1, pp. 47–53, 1986.
- [4] D. I. Sun, B. Ashley, B. Brewer, A. Hughes, and W. F. Tinney, "Optimal power flow by Newton approach," *IEEE Transactions on Power Apparatus and Systems*, vol. 103, no. 10, pp. 2864–2880, 1984.
- [5] S. Granville, "Optimal reactive dispatch through interior point methods," *IEEE Transactions on Power Systems*, vol. 9, no. 1, pp. 136–146, 1994.
- [6] E. Rezaia and S. M. Shahidehpour, "Real power loss minimization using interior point method," *International Journal of Electrical Power & Energy Systems*, vol. 23, no. 1, pp. 45–56, 2001.
- [7] J. Zhu and X. Xiong, "Optimal reactive power control using modified interior point method," *Electric Power Systems Research*, vol. 66, no. 2, pp. 187–192, 2003.
- [8] O. Alsaç, J. Bright, M. Prais, and B. Stott, "Further developments in LP-based optimal power flow," *IEEE Transactions on Power Systems*, vol. 5, no. 3, pp. 697–711, 1990.
- [9] B. Stott and J. L. Marinho, "Linear programming for power-system network security applications," *IEEE Transactions on Power Apparatus and Systems*, vol. 98, no. 3, pp. 837–848, 1979.
- [10] A. M. Chebbo and M. R. Irving, "Combined active and reactive dispatch. i. problem formulation and solution algorithm," *IEE Proceedings-Generation, Transmission and Distribution*, vol. 142, no. 4, pp. 393–400, 1995.
- [11] F.-C. Lu and Y.-Y. Hsu, "Reactive power/voltage control in a distribution substation using dynamic programming," *IEE Proceedings Generation, Transmission and Distribution*, vol. 142, no. 6, pp. 639–645, 1995.
- [12] V. H. Quintana and M. Santos-Nieto, "Reactive-power dispatch by successive quadratic programming," *IEEE Transactions on Energy Conversion*, vol. 4, no. 3, pp. 425–435, 1989.
- [13] N. Grudin, "Reactive power optimization using successive quadratic programming method," *IEEE Transactions on Power Systems*, vol. 13, no. 4, pp. 1219–1225, 1998.
- [14] V. de Sousa, E. Baptista, and G. da Costa, "Optimal reactive power flow via the modified barrier Lagrangian function approach," *Electric Power Systems Research*, vol. 84, no. 1, pp. 159–164, 2012.
- [15] Y. J. Cao and Q. H. Wu, "Optimal reactive power dispatch using an adaptive genetic algorithm," in *Proceedings of the Second International Conference on Genetic Algorithms in Engineering Systems*, pp. 117–122, Glasgow, UK, 1997.
- [16] D. T. Khanmiri, N. Nasiri, and T. Abedinzadeh, "Optimal reactive power dispatch using an improved genetic algorithm," *International Journal of Computer and Electrical Engineering*, vol. 4, no. 4, pp. 463–466, 2012.
- [17] B. B. Pal, P. Biswas, and A. Mukhopadhyay, "GA based FGP approach for optimal reactive power dispatch," *Procedia Technology*, vol. 10, pp. 464–473, 2013.
- [18] M. S. Alam and M. De, "Optimal reactive power dispatch using hybrid loop-genetic based algorithm," in *Proceedings of the 2016 19th National Power Systems Conference (NPSC)*, pp. 1–6, Bhubaneswar, India, December 2016.
- [19] K. Rayudu, G. Yesuratnam, and A. Jayalaxmi, "Improving voltage stability by optimal reactive power dispatch based on genetic algorithm and linear programming technique," in

- Proceedings of the 2016 International Conference on Electrical, Electronics, and Optimization Techniques, ICEEOT 2016*, pp. 1357–1362, March 2016.
- [20] A. A. E. Ela, M. A. Abido, and S. R. Spea, “Differential evolution algorithm for optimal reactive power dispatch,” *Electric Power Systems Research*, vol. 81, no. 2, pp. 458–464, 2011.
- [21] H. Singh and L. Srivastava, “Modified differential evolution algorithm for multi-objective VAR management,” *International Journal of Electrical Power & Energy Systems*, vol. 55, pp. 731–740, 2014.
- [22] M. Basu, “Quasi-oppositional differential evolution for optimal reactive power dispatch,” *International Journal of Electrical Power & Energy Systems*, vol. 78, pp. 29–40, 2016.
- [23] M. Basu, “Multi-objective optimal reactive power dispatch using multi-objective differential evolution,” *International Journal of Electrical Power & Energy Systems*, vol. 82, pp. 213–224, 2016.
- [24] Y. Li, X. Li, and Z. Li, “Reactive power optimization using hybrid CAB-DE algorithm,” *Electric Power Components and Systems*, vol. 45, no. 9, pp. 980–989, 2017.
- [25] M. Kumari and M. Sydulu, “Improved Particle Swarm Algorithm Applied to Optimal Reactive Power Control,” in *Proceedings of the 2006 IEEE International Conference on Industrial Technology*, pp. 1873–1878, Mumbai, India, December 2006.
- [26] G. Cai, Z. Ren, and T. Yu, “Optimal reactive power dispatch based on modified particle swarm optimization considering voltage stability,” in *Proceedings of the 2007 IEEE Power Engineering Society General Meeting*, pp. 1–5, Tampa, FL, USA, June 2007.
- [27] K. Mahadevan and P. S. Kannan, “Comprehensive learning particle swarm optimization for reactive power dispatch,” *Applied Soft Computing*, vol. 10, no. 2, pp. 641–652, 2010.
- [28] J. Polprasert, W. Ongsakul, and V. N. Dieu, “Optimal reactive power dispatch using improved pseudo-gradient search particle swarm optimization,” *Electric Power Components and Systems*, vol. 44, no. 5, pp. 518–532, 2016.
- [29] M. Mehdinejad, B. Mohammadi-Ivatloo, R. Dadashzadeh-Bonab, and K. Zare, “Solution of optimal reactive power dispatch of power systems using hybrid particle swarm optimization and imperialist competitive algorithms,” *International Journal of Electrical Power & Energy Systems*, vol. 83, pp. 104–116, 2016.
- [30] E. Naderi, H. Narimani, M. Fathi, and M. R. Narimani, “A novel fuzzy adaptive configuration of particle swarm optimization to solve large-scale optimal reactive power dispatch,” *Applied Soft Computing*, vol. 53, pp. 441–456, 2017.
- [31] J. Ding, Q. Zhang, and Y. Ma, “Optimal reactive power dispatch based on the CS-PSO algorithm,” in *Proceedings of the 13th Conference on Industrial Electronics and Applications (ICIEA)*, IEEE, Wuhan, China, 2018.
- [32] S. Duman, Y. Sönmez, U. Güvenç, and N. Yörükeren, “Optimal reactive power dispatch using a gravitational search algorithm,” *IET Generation, Transmission & Distribution*, vol. 6, no. 6, pp. 563–576, 2012.
- [33] P. K. Roy, B. Mandal, and K. Bhattacharya, “Gravitational search algorithm based optimal reactive power dispatch for voltage stability enhancement,” *Electric Power Components and Systems*, vol. 40, no. 9, pp. 956–976, 2012.
- [34] S. Duman, Y. Sonmez, U. Guvenc, and N. Yorukeren, “Application of gravitational search algorithm for optimal reactive power dispatch problem,” in *Proceedings of the International Symposium on INnovations in Intelligent SysTems and Applications (INISTA '11)*, pp. 519–523, June 2011.
- [35] G. Chen, L. Liu, Z. Zhang, and S. Huang, “Optimal reactive power dispatch by improved GSA-based algorithm with the novel strategies to handle constraints,” *Applied Soft Computing*, vol. 50, pp. 58–70, 2017.
- [36] B. Mandal and P. K. Roy, “Optimal reactive power dispatch using quasi-oppositional teaching learning based optimization,” *International Journal of Electrical Power & Energy Systems*, vol. 53, pp. 123–134, 2013.
- [37] M. Ghasemi, M. Taghizadeh, S. Ghavidel, J. Aghaei, and A. Abbasian, “Solving optimal reactive power dispatch problem using a novel teaching-learning-based optimization algorithm,” *Engineering Applications of Artificial Intelligence*, vol. 39, pp. 100–108, 2015.
- [38] A. Rajan and T. Malakar, “Optimal reactive power dispatch using hybrid Nelder–Mead simplex based firefly algorithm,” *International Journal of Electrical Power & Energy Systems*, vol. 66, pp. 9–24, 2015.
- [39] K. Rayudu, G. Yesuratnam, and A. Jayalaxmi, “Artificial Bee Colony algorithm for optimal reactive power dispatch to improve voltage stability,” in *Proceedings of the 2016 IEEE International Conference on Circuit, Power and Computing Technologies, ICCPCT 2016*, pp. 1–7, March 2016.
- [40] K. Abaci and V. Yamaçlı, “Optimal reactive-power dispatch using differential search algorithm,” *Electrical Engineering*, vol. 99, no. 1, pp. 213–225, 2017.
- [41] A. Rajan and T. Malakar, “Exchange market algorithm based optimum reactive power dispatch,” *Applied Soft Computing*, vol. 43, pp. 320–336, 2016.
- [42] A. Mukherjee and V. Mukherjee, “Chaos embedded krill herd algorithm for optimal VAR dispatch problem of power system,” *International Journal of Electrical Power & Energy Systems*, vol. 82, pp. 37–48, 2016.
- [43] M. H. Sulaiman, Z. Mustaffa, M. R. Mohamed, and O. Aliman, “Using the gray wolf optimizer for solving optimal reactive power dispatch problem,” *Applied Soft Computing*, vol. 32, pp. 286–292, 2015.
- [44] A. A. Heidari, R. Ali Abbaspour, and A. Rezaee Jordehi, “Gaussian bare-bones water cycle algorithm for optimal reactive power dispatch in electrical power systems,” *Applied Soft Computing*, vol. 57, pp. 657–671, 2017.
- [45] S. Mouassa, T. Bouktir, and A. Salhi, “Ant lion optimizer for solving optimal reactive power dispatch problem in power systems,” *Engineering Science and Technology, an International Journal*, vol. 20, no. 3, pp. 885–895, 2017.
- [46] R. Ng Shin Mei, M. H. Sulaiman, Z. Mustaffa, and H. Daniyal, “Optimal reactive power dispatch solution by loss minimization using moth-flame optimization technique,” *Applied Soft Computing*, vol. 59, pp. 210–222, 2017.
- [47] K. B. O. Medani, S. Sayah, and A. Bekrar, “Whale optimization algorithm based optimal reactive power dispatch: A case study of the Algerian power system,” *Electric Power Systems Research*, vol. 163, pp. 696–705, 2018.
- [48] K. Rayudu, G. Yesuratnam, and A. Jayalaxmi, “Ant colony optimization algorithm based optimal reactive power dispatch to improve voltage stability,” in *Proceedings of the 2017 IEEE International Conference on Circuit, Power and Computing Technologies, ICCPCT 2017*, pp. 1–6, April 2017.
- [49] A. M. Shaheen, R. A. El-Sehiemy, and S. M. Farrag, “Optimal reactive power dispatch using backtracking search algorithm,”

- Australian Journal of Electrical and Electronics Engineering*, vol. 13, no. 3, pp. 200–210, 2016.
- [50] H. Salimi, “Stochastic fractal search: a powerful metaheuristic algorithm,” *Knowledge-Based Systems*, vol. 75, pp. 1–18, 2015.
- [51] H. Mosbah and M. El-Hawary, “Power system tracking state estimation based on stochastic fractal search technique under sudden load changing conditions,” in *Proceedings of the 2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, pp. 1–6, Vancouver, Canada, May 2016.
- [52] I. Khanam and G. Parmar, “Application of SFS algorithm in control of DC motor and comparative analysis,” in *Proceedings of the 2017 4th IEEE Uttar Pradesh Section International Conference on Electrical, Computer and Electronics (UPCON)*, pp. 256–261, Mathura, October 2017.
- [53] M. I. Alomoush and Z. B. Oweis, “Environmental-economic dispatch using stochastic fractal search algorithm,” *International Transactions on Electrical Energy Systems*, vol. 28, no. 5, article no. e2530, 2018.
- [54] E. Çelik, “Incorporation of stochastic fractal search algorithm into efficient design of PID controller for an automatic voltage regulator system,” *Neural Computing and Applications*, vol. 30, no. 6, pp. 1991–2002, 2018.
- [55] H. M. Dubey, M. Pandit, and B. K. Panigrahi, “An overview and comparative analysis of recent bio-inspired optimization techniques for wind integrated multi-objective power dispatch,” *Swarm and Evolutionary Computation*, vol. 38, pp. 12–34, 2018.
- [56] T. P. Nguyen and D. N. Vo, “A novel stochastic fractal search algorithm for optimal allocation of distributed generators in radial distribution systems,” *Applied Soft Computing*, vol. 70, pp. 773–796, 2018.
- [57] N. H. Awad, M. Z. Ali, P. N. Suganthan, and E. Jaser, “Differential evolution with stochastic fractal search algorithm for global numerical optimization,” in *Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC)*, pp. 3154–3161, Vancouver, Canada, July 2016.
- [58] S. Padhy and S. Panda, “A hybrid stochastic fractal search and pattern search technique based cascade PI-PD controller for automatic generation control of multi-source power systems in presence of plug in electric vehicles,” *CAAI Transactions on Intelligence Technology*, vol. 2, no. 1, pp. 12–25, 2017.
- [59] J. Lin and Z.-J. Wang, “Parameter identification for fractional-order chaotic systems using a hybrid stochastic fractal search algorithm,” *Nonlinear Dynamics*, vol. 90, no. 2, pp. 1243–1255, 2017.
- [60] H. Mosbah and M. El-Hawary, “Power system static state estimation using modified stochastic fractal search technique,” in *Proceedings of the IEEE Canadian Conference on Electrical & Computer Engineering (CCECE)*, pp. 1–4, 2018.
- [61] T. P. Nguyen, T. T. Tran, and D. N. Vo, “Improved stochastic fractal search algorithm with chaos for optimal determination of location, size, and quantity of distributed generators in distribution systems,” *Neural Computing and Applications*, pp. 1–26, 2018.
- [62] S. Das and P. N. Suganthan, “Differential evolution: a survey of the state-of-the-art,” *IEEE Transactions on Evolutionary Computation*, vol. 15, no. 1, pp. 4–31, 2011.
- [63] D. Zou, S. Li, G.-G. Wang, Z. Li, and H. Ouyang, “An improved differential evolution algorithm for the economic load dispatch problems with or without valve-point effects,” *Applied Energy*, vol. 181, pp. 375–390, 2016.
- [64] T. T. Nguyen and D. N. Vo, “Improved social spider optimization algorithm for optimal reactive power dispatch problem with different objectives,” *Neural Computing and Applications*, pp. 1–32, 2019.
- [65] P. Subbaraj and P. N. Rajnarayanan, “Optimal reactive power dispatch using self-adaptive real coded genetic algorithm,” *Electric Power Systems Research*, vol. 79, no. 2, pp. 374–381, 2009.
- [66] T. Kang, J. Yao, M. Jin, S. Yang, and T. Duong, “A Novel Improved Cuckoo Search Algorithm for Parameter Estimation of Photovoltaic (PV) Models,” *Energies*, vol. 11, no. 5, pp. 1–31, 2018.
- [67] M. Fayaz and D. Kim, “Energy consumption optimization and user comfort management in residential buildings using a bat algorithm and fuzzy logic,” *Energies*, vol. 11, no. 1, p. 161, 2018.
- [68] “MATPOWER 4.1 IEEE 30-bus and 118-bus test system,” <http://www.pserc.cornell.edu/matpower>.

