Hindawi
Mathematical Problems in Engineering
Volume 2019, Article ID 1564693, 29 pages
https://doi.org/10.1155/2019/1564693

Research Article

Finding Optimal Load Dispatch Solutions by Using a Proposed Cuckoo Search Algorithm

Thang Trung Nguyen ,[1] Cong-Trang Nguyen,[1] Le Van Dai ,[2] and Nguyen Vu Quynh [3]

[1] *Power System Optimization Research Group, Faculty of Electrical and Electronics Engineering, Ton Duc Thang University, Ho Chi Minh City, Vietnam*
[2] *Institute of Research and Development, Duy Tan University, Da Nang, Vietnam*
[3] *Department of Electromechanical and Electronic, Faculty of Mechatronics and Electronics, Lac Hong University, Dong Nai, Vietnam*

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

Received 22 January 2019; Revised 16 April 2019; Accepted 16 May 2019; Published 28 May 2019

Academic Editor: Changzhi Wu

Optimal load dispatch (OLD) is an important engineering problem in power system optimization field due to its significance of reducing the amount of electric generation fuel and increasing benefit. In the paper, an improved cuckoo search algorithm (ICSA) is proposed for determining optimal generation of all available thermal generation units so that all constraints consisting of prohibited power zone (PPZ), real power balance (RPB), power generation limitations (PGL), ramp rate limits (RRL), and real power reserve (RPR) are completely satisfied. The proposed ICSA method performance is more robust than conventional Cuckoo search algorithm (CCSA) by applying new modifications. Compared to CCSA, the proposed ICSA approach can obtain high quality solutions and speed up the solution search ability. The ICSA robustness is verified on different systems with diversification of objective functions as well as the considered constraint set. The results from the proposed ICSA method are compared to other algorithms for comparison. The result comparison analysis indicates that the proposed ICSA approach is more robust than CCSA and other existing optimization approaches in finding solutions with significant quality and shortening simulation time. Consequently, it should lead to a conclusion that the proposed ICSA approach deserves to be applied for finding solutions of OLD problem in power system optimization field.

1. Introduction

Over the past decades, an enormous number of studies have concerned and solved different optimization operation problems in regard to electric grids by utilizing potential search ability of optimization approaches. Many concerned operation problems regarding distribution power network, transmission network, and different types of power plant as well as electric components in the network have been successfully solved. This study focuses on optimal load dispatch (OLD) problem with the task of allocating the generated power of all considered thermal generation units to reduce the cost of burnt fossil fuels. All physical and operational constraints are required to be exactly satisfied. If all generation units in each power plant and all power plants are working under the most appropriate schedule, total fuel cost of all units can be the smallest and consumers can get significant amount of revenue [1]. The achievement is thanks to the meaning of OLD problem.

A huge number of optimization approaches using mathematical programming have been widely applied so far for solving the considered OLD problem such as dynamic programming (DP) [2], lambda iteration method [3], Newton Raphson and Lagrangian multiplier (NRLM) method [4], and linear programming (LP) [5, 6]. Conventional methods focused on the systems with simple constraints and convex objective function where nonlinear constraints and the effects of valve loading process were not considered. The complex level of the constraints and objective has been mentioned in many articles. For example, the authors in [3] could only solve the OLD problem successfully by separating three-curve objective function into three different single curve objective functions. In [7], a more realistic representation of the electric generation fuel function corresponding to multi fossil fuel sources was introduced, in which the authors considered a discontinuous cost function and the effects of valve loading process (EoVLP) of thermal units. More complicated models were also introduced. For example, some thermal generation units were driven by burning multi fossil fuel sources (MFS) to generate electricity [8], or some operating conditions of generators including upper generation boundary, lower generation boundary, and prohibited power zones were added [9]. However, these methods have been only applied to the systems where the generation power-fuel cost characteristic of thermal generation units was mathematically modelled as the second order function and the effects of valve loading process were ignored [10].

Other artificial intelligence-based advanced methods have been recently applied for solving OLD problem. Evolutionary programming-based approaches (EP) [1, 11, 12] and genetic algorithm (GA) [13–15] were considered as fast algorithms because of their parallel search ability. In addition, GA and EP possessed good properties such as finding solutions nearby global optimum, capability of effectively handling nonlinear constraints, reliable search ability, and not many adjustment parameters [10]. Thus, it could become a suitable choice for successfully solving OLD problem. However, GA was prematurely convergent to local optimum solutions [16]. In this regard, simulated annealing approach (SA) [17] was a better probabilistic approach in finding solutions with appropriate fitness but there was a high possibility of easily converging to local optimal zones when coping with complicatedly constrained problem. It converges slower than GA and EP, though. Differential evolution algorithm (DE) [10, 18] also belongs to the same class as GA and EP. However, it is more popular thanks to simple structure with several adjustment parameters and high rate of success. DE has been more widely and successfully applied than SA and GA [19]. But its faster convergence manner led to the same drawback as GA like high rate of falling into local optimum and hardly ever toward promising zones quickly. In fact, these shortcomings could be tackled by setting population size to higher value. But high population size could suffer from long simulation time to calculate fitness function and evaluate quality of solutions [20]. Hopfield neural network (HNN) [21, 22] focused on optimizing energy function and was only successfully applied for optimization problems where objective functions were differentiable. HNN could be an appropriate method for large-scale systems with high number of generation units but it needed long simulation time and may also converge to local optimum solution zones [23]. Particle swam optimization (PSO) [16, 24] is a random search approach developed by behavior of a swarm or flock during food search process. In comparison with GA, PSO owned more advantages such as simpler implementation and few parameters with easy selection. However, the success rate of PSO was highly influenced by adjustment parameters and it copes with high rate to be trapped in many zones with local optimum solutions [25]. Harmony search (HS) [26] is a metaheuristic-based method inspired from music. Instead of using gradient search, HS has employed stochastic random search to exploit its potential ability. Thus, it tended to converge to local optimal zones rather than global optimal zones [27]. Biogeography-based optimization (BBO) [28] could compete with PSO and DE since its solutions were directly updated by migration from other existing solutions and its solutions directly shared their attributes with other solutions [29].

It is clear that each method has advantages as well as disadvantages for different applications for finding OLD problem solutions. Hence, another natural approach is to combine different methods to exploit the advantages of each method and enhance the overall searching capability. Several hybrid methods have been developed in such way including hybrid Genetic algorithm, Pattern Search and Sequential Quadratic Programming (GA–PS–SQP) method [30], hybrid Artificial Cooperative Search algorithm (HACSA) [31], hybrid PSO-SQP [32], and hybrid GA (HGA) [33]. Basically, these hybrid methods could deal with OLD problem more effectively than each member method. On the other hand, they could suffer from the difficulty of selecting many controllable parameters. In addition to such popular original algorithms and hybrid methods, there are many other original and improved methods that have been applied for solving the considered OLD problem. These methods are Symbiotic organisms search algorithm (SOS) [34] and its modified version (MSOS) [34], teaching activity and learning activity-based optimization (TLBO) [35], chemical reaction-based approach (CRBA) [36], enhanced particle swarm optimization (EPSO) [37], sequential quadratic technique-based cross entropy approach (CEA-SQT) [38], traverse search-based optimization approach (TSBO) [39], invasive weed approach (IWA) [40], Improved Differential evolution (IDE) [41], immune algorithm using power redistribution IAPR [42], Colonial competitive differential evolution (CCDE) [43], Chaotic Bat algorithm (CBA) [44], Exchange market algorithm (EMA) [45], adaptive search technique algorithm and differential evolution (GRASP-DE) [46], θ-Modified Bat Algorithm (θ-MBA) [47], Tournament-based harmony search algorithm (TBHSA) [48], New Modified β-Hill Climbing Local Search Algorithm (Mβ-HCLSA) [49], improved version of artificial bee colony algorithm (IABCA) [50], artificial cooperative search algorithm (ACSA) [51], and ameliorated grey wolf optimization algorithm (AGWOA) [52]. Among these methods, ACSA and AGWOA were the two latest methods, which were applied for OLD and published in early 2019. However, the demonstration of real performance of the two methods is still questionable. In fact, ACSA has been tested only on systems with small scale, single fuel and simple constraints such as generation limits and power balance. The largest scale system was considered in [51] to be 40-unit system. Unlike [51], different types of fuel cost function, complicated constraints, and large scale system with 140 units have been taken into account in [52]. Via comparisons with many existing methods, AGWOA has been stated to be the best one with many surprising results. Thus, the validation of reported solutions from the method must be verified and its strong search ability must be reevaluated. In the numerical results section, we will report the verification of the two questionable issues.

In this paper, we have proposed an improved cuckoo search algorithm (ICSA) for dealing with large scale OLD problem with the consideration of complicated constraints together with nondifferentiable fuel cost function. In the proposed ICSA approach, some new modifications have been performed on conventional cuckoo search algorithm (CCSA) to improve the quality of CCSA. The CCSA method was first developed in 2009 [53] for solving a set of popular benchmark functions and its highly superior performance over PSO and GA has attracted a huge number of researchers in learning and applying for different optimization problems in different fields. Furthermore, its improved variants are also an extremely vast number. In relation to OLD problem, CCSA has been applied and presented in [23, 54–57] meanwhile its improved methods consisting of modified cuckoo search algorithm (MCSA) and improved cuckoo search algorithm with one solution evaluation (OSE-CSA) have been, respectively, presented in [58, 59]. In [55], Basu has applied CCSA for solving OLD problem with 40-unit system with single fuel option and the effective of valve loading process, 20-unit system with single fuel and quadratic fuel cost function, and 10-unit system with multiple fuel sources and without the effects of valve loading process. The author has made a big effort in demonstrating the high potential search of CCSA by comparing with many popular metaheuristic algorithms but the shortcoming of the study was neglecting complicated constraints and large scale systems.

Studies in [56, 57] have dealt with OLD problem with two power systems considering simple constraints and small number of units. Only three simplest constraints, such as power balance, limitations of generation, and prohibited power zones, have been taken into account meanwhile the largest system was solved to be 6-unit system. Thus, there were few methods compared to CCSA and the real performance of CCSA was not shown persuasively in the studies. In [23], authors have applied CCSA for solving different systems with very complicated constraints, complicated characteristics of thermal generating units and high number of units. Among the mentioned studies regarding CCSA for OLD problem, authors in [23] could show the best view in evaluating the real performance of CCSA since there were six cases that were carried out and a huge number of methods were compared to CCSA. In spite of the real potential search ability, CCSA has been commented to be low convergence to global optimum and significantly improved better [58, 59]. MCSA in [58] has been proposed by using a new strategy for the second generation technique. The mutation operation in CCSA has been replaced with current-to-best/1 model of DE in [60]. MCSA has been applied for solving four systems with 3, 6, 15, and 40 units in which the most complicated constraint considered was prohibited power zone and only single fuel source was taken into account. MCSA method has been compared to other popular methods such as PSO, GA, and EP. But the comparisons with CCSA have not been carried out. Thus, the improvement of such proposed method in [58] was not proved persuasively. OSE-CSA in [59] has canceled one evaluation time in case that OSE-CSA has continued to improve solution quality. The improvement seemed to be appropriate for CCSA in dealing with OLD problem with complicated systems. CCSA in [23, 55] have been considered for comparison in [59] and they have been proved to be less effective than OSE-CSA. However, OSE-CSA has used one more control parameter, called one rank parameter, and it needed to be tuned thoroughly for obtaining high performance. In the proposed ICSA approach, we have focused on a new strategy of the second new solution generation in CCSA method. As shown in [58, 59], CCSA has become a strong search method thanks to the first new solution generation, which was performed by Levy flight technique while the second new solution generation could not take on local search function well. In the second update progress via mutation operator, two random old solutions are used to generate an increased step size. However, the manner can lead to new low quality solutions because the increased step will be very small when iterative algorithm is carrying out at the last iterations. In fact, current solutions at final several iterations tend to be close together and the different values between each two ones are very small, leading to a very small increased step. In order to tackle the disadvantage of the CCSA, we apply a new adaptive technique for improvement of solution quality. Firstly, we propose two ways for producing the increased step including two-solution-based increased step and four-solution-based increased step. The decision when which step size will be used is dependent on the result of comparison between fitness function ratio (FFR) and a predetermined parameter Tol. FFR is defined as a ratio of deviation between fitness functions of the considered solution and the most promising solution to the fitness value of the best one meanwhile Tol is a boundary to give the final decision for the selection of a used step size. At the beginning, Tol is a fixed value for all solutions and then it will be adaptive based on the comparison between it and FFR. When FFR of a solution is less than Tol, Tol of the solution will be decreased equally to ninety percent of the previous value. Otherwise, the value of Tol remains unchanged in case the FFR is equal to or higher than Tol. The adaptive technique has a significantly important role in enhancing the potential search ability of the proposed method. This proposed method is investigated on six cases with different considered constraints, different types of fuel cost function, and large scale systems. The detail of the six cases is as follows:

Case 1. Four systems with single fuel source (SFS) and power loss (PL) constraint

Case 2. A 110-unit system with SFS

Case 3. Four systems with SFS and the effects of valve loading process (EoVLP)

Case 4. Two systems with SFS, and PPZ and RPR constraints

Case 5. A 15-unit system with SFS, and RRL, PPZ, and PL constraints

Case 6. Three systems with multiple fuel sources (MFS) and EoVLP

The achieved results in terms of minimum fuel cost, average fuel cost, maximum fuel cost, and standard deviation found by the proposed method compared to those obtained by others reveal that the method is very efficient for the OLD problem. In addition, the performance improvement of the proposed method over CCSA is also investigated via the comparison of the best solution and all trial runs. In summary, the main advantages of the proposed ICSA approach over CCSA as well as the main contribution of the study are as follows:

(i) Based on fitness function of each considered solution, local search or global search is decided to be applied more effectively

(ii) Find better solutions with smaller number of iterations and shorter execution time for each run

(iii) Shorten simulation time for the whole search of each study case

However, the proposed method also copes with the same shortcomings as CSA. Although the shortcomings do not cause bad results for the proposed method, they make the proposed method be time consuming in tuning optimal parameter. The shortcomings are analyzed as follows:

(i) Control parameter, probability of replacing control variables in each old solution, must be tuned in range between 0 and 1. There is no proper theory for determining the most effective values of the parameter. Thus, the performance of the proposed method must be tried by setting the parameter to values from 0.1 to 1.

(ii) The method uses more computation steps for search process. Thus, the proposed method uses higher number of computation steps for each iteration. However, due to more effective search ability for each iteration, the proposed method can use smaller number of iterations but it finds more effective solutions.

The remaining parts of the paper are arranged as follows: Section 2 shows the objective and constraints of the considered OLD problem. CCSA and the proposed method are clearly explained in Section 3. Section 4 is in charge of presenting the implementation of ICSA method for the studied problem. The simulation results together with analysis and discussions are given in Section 5. Finally, conclusion is summarized in Section 6. In addition, appendix is also added for showing found solutions by the proposed ICSA approach for test cases.

2. Optimal Load Dispatch Problem Description

2.1. Fuel Cost Function Forms with Single Fuel Source. In the considered OLD problem, the optimal operation of a set of thermal generation units is concerned as the duty of reducing total cost of all the units, which can be seen by the following model:

$$\text{Reduce } F = \sum_{i=1}^{N} F_i\left(P_i\right) \tag{1}$$

In traditional OLD problem, fuel cost function of the *ith* generation unit $F_i(P_i)$ is represented as the second order function with respect to real power output and coefficients as the model below [2]:

$$F_i\left(P_i\right) = \alpha_i P_i^2 + \lambda_i P_i + \delta_i \tag{2}$$

In addition, for the case considering the effects of valve loading process on thermal generation units, fuel cost becomes more complicated by adding sinusoidal term as below [12]:

$$\begin{aligned} F_i\left(P_i\right) &= \alpha_i P_i^2 + \lambda_i P_i + \delta_i \\ &\quad + \left|\beta_i \times \sin\left(\gamma_i \times \left(P_{i,\min} - P_i\right)\right)\right| \end{aligned} \tag{3}$$

Real Power Balance Constraint. Total real power demand of all loads in power system together with real power loss in all conductors must be equal to the generation from all available thermal generation units. The requirement is constrained by the following equality:

$$\sum_{i=1}^{N} P_i = P_D + P_L \tag{4}$$

where total real power loss, P_L, is determined by Kron's equation below:

$$P_L = B_{00} + \sum_{j=1}^{N} B_{0j} P_j + \sum_{j=1}^{N}\sum_{i=1}^{N} P_j B_{ji} P_i \tag{5}$$

Generation Boundary Constraint. For the purpose of economy and safe operation, each thermal generation unit is constrained by the lower generation bound and upper generation bound as the following model:

$$P_{i,\min} \le P_i \le P_{i,\max} \tag{6}$$

2.2. Fuel Cost Function Forms with Multi-Fuel Sources. In this section, fuel cost function of thermal generation units is mathematically modeled in terms of different forms from that in the section above due to the consideration of multi-fuel sources. Each type of fuel source is formed as each second order function and the fuel cost function form is the sum of different second order functions for the case of neglecting the effects of valve loading progress. But for the consideration case of the effects, the form is more complex with the presence of sinusoidal terms [15]. As a result, the forms of cost function can be expressed in Equation (7) [21] and Equation (8) [15]:

$$F_i(P_i) = \begin{cases} \delta_{i1} + \lambda_{i1} P_i + \alpha_{i1} P_i^2, & \text{fuel } 1, P_{i,\min} \le P_i \le P_{i1,max} \\ \delta_{i2} + \lambda_{i2} P_i + \alpha_{i2} P_i^2, & \text{fuel } 2, P_{i2,\min} \le P_i \le P_{i2,\max} \\ \vdots \\ \delta_{ij} + \lambda_{ij} P_i + \alpha_{ij} P_i^2, & \text{fuel } j, P_{ij,\min} \le P_i \le P_{ij,\max} \end{cases} \quad (7)$$

$$F_i(P_i) = \begin{cases} \delta_{i1} + \lambda_{i1} P_i + \alpha_{i1} P_i^2 + \left|\beta_{i1} \times \sin\left(\gamma_{i1} \times \left(P_{i,\min} - P_i\right)\right)\right|, & \text{for fuel } 1, P_{i,\min} \le P_i \le P_{i1,max} \\ \delta_{i2} + \lambda_{i2} P_i + \alpha_{i2} P_i^2 + \left|\beta_{i2} \times \sin\left(\gamma_{i2} \times \left(P_{i,\min} - P_i\right)\right)\right|, & \text{for fuel } 2, P_{i2,\min} \le P_i \le P_{i2,max}, \ j = 1, \ldots, m_i \\ \vdots \\ \delta_{ij} + \lambda_{ij} P_i + \alpha_{ij} P_i^2 + \left|\beta_{ij} \times \sin\left(\gamma_{ij} \times \left(P_{i,\min} - P_i\right)\right)\right|, & \text{for fuel } j, P_{ij,\min} \le P_i \le P_{ij,\max} \end{cases} \quad (8)$$

Cost function forms in Equations (7) and (8) are only included in objective function (1) meanwhile main constraints in formulas (4) and (6) must be always satisfied.

2.3. Prohibited Power Zone, Real Power Reserve, and Ramp Rate Limit Constraints. Prohibited power zones (PPZ) are different ranges of power in fuel cost function that thermal generation units are not allowed to work due to operation process of steam or gas valves in their shaft bearing. The power generation of units in the violated zones is harmful to gas or steam turbines even destroyed shaft bearing. Thus, the constraint is strictly observed. In the fuel-power characteristic curve of generation units, PPZ causes small violation zones and such curves become discontinuous. As considering PPZ constraint, the determination of power generation of units is more complex and equal to either lower bound or upper bound. Unlike PPZ constraint, RPR constraint is not related to fuel-power feature curve but it causes difficulty for optimization approaches in satisfying one more inequality constraint. Each generation unit among the set of available generation units must reserve real power so that the sum of real power from all generation units can be higher or equal to the requirement of power system for the purpose of stabilizing power system in case that there are some units stopping producing electricity. On the contrary to PPZ constraint, ramp rate limit (RRL) constraint does not allow power output of thermal generating units outside a predetermined range. The constraint considers maximum power change of each thermal generating unit as compared to the previous power value. Thus, optimal generation must satisfy the RRL constraint. The PPZ constraint, RPR constraint, and RRL constraint can be presented as follows:

Prohibited Power Zones. As considering PPZ constraint, valid working zones of each thermal generating unit are not continuous and its generation must be outside the violated zones as the following mathematical description:

$$P_i \in \begin{cases} P_{i,\min} \le P_i \le P_{i1}^l \\ P_{ik-1}^u \le P_i \le P_{ik}^l; \quad \mathrm{k} = 2, \ldots, \mathrm{n_i}; \ \forall i \in \Omega \\ P_{in_i}^u \le P_i \le P_{i,\max} \end{cases} \quad (9)$$

As observing Equation (9), generation units cannot be operated within the violated zones except for starting point and end point. Consequently, the verification of PPZ constraint violation should be carried out first and then the correction should be done before dealing with other constraints such as real power reserve constraint and real power balance. Besides, if power output of all units can satisfy the PPZ constraint, generation limits in Equation (6) are also exactly met.

Real Power Reserve Constraint. Real power reserve in power system aims to enhance the ability of stability recovery of power system and avoid blackout. In order to get high enough power for requirement, all available units are constrained by the following inequality:

$$\sum_{i=1}^{N} S_i \ge S_R \quad (10)$$

where S_i is the real power reserve contribution of the *ith* thermal generation unit and the determination of S_i can be done by employing the two models below:

$$S_i = \begin{cases} P_{i,\max} - P_i & if \ S_{i,\max} > \left(P_{i,\max} - P_i\right) \\ S_{i,\max} & else \end{cases}; \quad \forall i \notin \Omega \quad (11)$$

$$S_i = 0; \quad \forall i \in \Omega \quad (12)$$

Equation (10) shows that the constraint of prohibited power zones is not included in the real power reserve constraints; however, prohibited power zones are always strictly considered and must be exactly satisfied.

Ramp Rate Limit (RRL) Constraint. In OLD problem, all considered thermal generating units are supposed to be under working status but previous active power of each thermal generating unit is not taken into account. Thus, increased or decreased power is not constrained. This assumption seems to be not practical until RRL constraint is considered. RRL constraint considers initial power output and the power change is supervised. Regulated power can be higher or lower than the initial value as long as it is within a predetermined range. Increased step size (ISS) and decreased step size (DSS) are given as input data and they are used to limit the change of power output of each thermal generating unit. The constraint can be mathematically expressed as the following formula [7]:

$$P_{i,0} - DSS_i \leq P_i \leq P_{i,0} + ISS_i \tag{13}$$

where $P_{i,0}$ is the initial power output of the ith thermal generating unit before its power output is regulated; ISS_i and DSS_i are, respectively, maximum increased and decreased step sizes of the ith thermal generating unit.

3. The Proposed Cuckoo Search Algorithm

3.1. Classical Cuckoo Search Algorithm. In search technique of CCSA [53], a set of solutions is randomly generated within a predetermined range in the first step and then the quality of each one is ranked by computing value of fitness function. The most effective solution corresponding to the smallest value of fitness function is determined and then search procedure comes into a loop algorithm until the maximum iteration is reached. In the loop algorithm, two techniques updating new solutions two times (corresponding to two generations) are Lévy flights and mutation technique, which is called strange eggs identification technique. The two generations can produce promising quality solutions for CCSA. After each generation, CCSA will carry out comparing fitness of newly updated solutions and initial solutions for keeping better ones and abandoning worse ones. The most effective solution at last step of the loop search algorithm is determined and it is restored as one candidate solution for a study case. The detail of the two stages is as follows.

3.1.1. Lévy Flights Stage. This is the first calculation step in the loop algorithm and it also produces new solutions in the first generation for CCSA. New solution Sol_{newx} is created by the following model:

$$Sol_{newx} = Sol_x + \alpha \left(Sol_x - Sol_{Gbest}\right) \oplus L\acute{e}vy\left(\beta\right) \tag{14}$$

where α is the positive scaling factor and it is nearly set to different values for different problems in the studies [53, 62]. In the work, the most appropriate values for such factor can be chosen to be 0.25/0.5 for different systems.

3.1.2. Discovery of Alien Eggs Stage. The step plays a very important role for updating new solutions Sol_{newx} of the whole population. However, not every control variable in each old solution is newly updated and the decision of replacement is dependent on comparison criteria as the following equation:

$$Sol_{newx} = \begin{cases} Sol_x + \varepsilon_1 . \left(Sol_{rand1} - Sol_{rand2}\right) & if\ \varepsilon_2 < P_a \\ Sol_x & otherwise \end{cases} \tag{15}$$

3.2. Proposed Algorithm. In the part, a new variant of CCSA (ICSA) is constructed by applying three effective changes on the main functions of CCSA in order to shorten simulation time corresponding to reduction of iterations and find more promising solutions. The proposed amendments are explained in detail as follows:

(i) Suggest one more equation producing updated step size in addition to existing one in CCSA

(ii) Create a new selection standard by computing fitness function ratio ΔFFR_x and comparing ΔFFR_x with a predetermined parameter Tol_x. Thus, thanks to the standard, the existing updated step size and additional update step size will be chosen more effectively

(iii) Automatically change value of Tol_x for the xth solution based on the result of comparing ΔFFR_x with the previous Tol_x

Such three points are clarified by observing the following sections:

3.2.1. Strange Eggs Identification Technique (Mutation Technique). The first proposed improvement in our proposed ICSA approach is to select a more suitable formula for producing new solutions with better fitness function value. In CCSA, Equation (16) below is used to produce a changing step nearby old solutions for all current solutions.

$$\Delta Sol_{newx,1} = \varepsilon_3 . \left(Sol_{rand1} - Sol_{rand2}\right) \tag{16}$$

The use of Equation (16) aims to produce a random walk around old solutions in search zones with intent to find out promising solutions. In order to reduce the possibility of suffering the local trap and approach to other favorable zones for searching, we propose a new Equation (17). The formula is built by the idea of enlarging search zone with the use of two more available solutions. Obviously, the larger changing step can own higher performance in moving to other search spaces that the classical approach used in CCSA. The suggestion is mathematically expressed by the formula below:

$$\Delta Sol_{newx2} = \varepsilon_4 . \left(Sol_{rand1} - Sol_{rand2} + Sol_{rand3} - Sol_{rand4}\right) \tag{17}$$

The changing step obtained by using Eq. (17) is named four-point changing step. Now, two solutions, which are newly formed by using two different changing steps shown in formulas (16) and (17), are found by the two following methods:

$$Sol_{newx1} = Sol_x + \Delta Sol_{newx1} \tag{18}$$

$$Sol_{newx2} = Sol_x + \Delta Sol_{newx2} \tag{19}$$

It can be clearly observed that the distance between Sol_x (old solution) and Sol_{newx1} (new solution) is lower than that between Sol_x and Sol_{newx2}. This difference can contribute a highly efficient improvement to the proposed ICSA approach search ability.

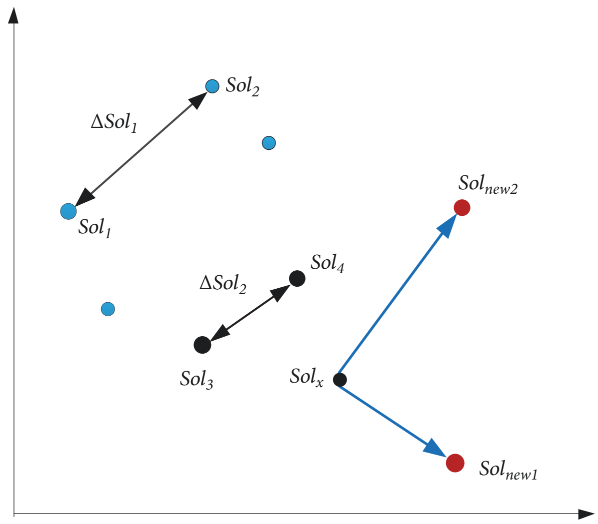

Figure 1: Simulation of solutions corresponding to the first iterations of the loop algorithm.

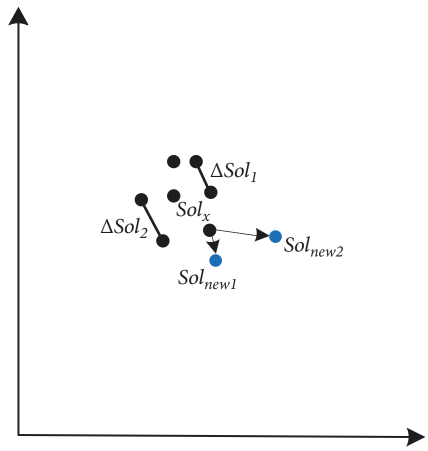

Figure 2: Simulation of solutions corresponding to the last iterations of the loop algorithm.

For the CCSA case, if two solutions Sol_{rand1} and Sol_{rand2} are either slightly different or completely coincident, such newly updated solution Sol_{newx1} does not have good chance to leave the current zone and approach to more promising zones. In another word, the new one is approximately coincident with the old one. As the search task is taking place at some last iterations, this phenomenon becomes much worse because all current solutions are lumped in a small zone and the capability of moving to other zones is impossible. As a result, the CCSA approach will work ineffectively and search strategy is time consuming until other runs are started.

Contrary to the two-point step size, the new proposed formula may produce a large enough length to escape the local optimum zone and reach new favorable zones. It explains why the four-point changing step has positive impact on the considered random walk rather than the two-point changing step.

3.2.2. New Standard for Choosing the Most Appropriate Changing Step. In this section, we extend our analysis to answer the question when to use the four-point step size. From Equations (18) and (19), two new solutions which are represented as the results of the two-point-based factor and the four-point step size can be illustrated by using Figure 1 corresponding to the search process at the first some iterations and Figure 2 corresponding to the last some iterations. For the sake of simplicity, we rewrite the two equations as follows:

$$Sol_{new1} = Sol_x + \Delta Sol_1 \quad (20)$$

$$Sol_{new2} = Sol_x + \Delta Sol_1 + \Delta Sol_2 \quad (21)$$

Here we suppose that ΔSol_1 and ΔSol_2 are obtained by four exact solutions, Sol_1, Sol_2, Sol_3 and Sol_4, and calculated as follows:

$$\Delta Sol_1 = Sol_1 - Sol_2 \quad (22)$$

$$\Delta Sol_2 = Sol_3 - Sol_4 \quad (23)$$

As mentioned above, the high changing step between new solution and old solution can help to explore new favorable zones. However, in optimization algorithms, searching steps cannot be arbitrarily large; otherwise the algorithm may diverge, in particular, for the cases that the considered solutions Sol_x are not close together in solution search space. For example, at the beginning of loop algorithm with the first iterations in Figure 1, Sol_{new1} is a better choice than Sol_{new2} because it is kept in a sufficient limit and does not lead to a risk of divergence. In contrast, as many of current solutions are in different positions but their distance is not very short or approximately coincident such as at the last iterations in Figure 2, Sol_{new1} and Sol_x have a very short distance but Sol_{new2} and Sol_x have higher distance. According to the phenomenon in Figure 2, the proposed ICSA approach needs to produce a high changing step to move solutions to other search zones without local optimum. Hence, Sol_{new2} would be preferred to Sol_{new1}.

Based on the argument above, the determination of the condition for using either two-point changing step or four-point changing step is really crucial to the performance of the proposed ICSA approach in searching solutions of OLD problem. Here, the ratio of ΔFFR_x, which can be found by Equation (24), is suggested to be a suitable measurement for the selection of two options.

$$\Delta FFR_x = \frac{FF_x - FF_{best}}{FF_{best}} \quad (24)$$

For a particular set of the current solutions, each individual depending on its ΔFFR_x will create a corresponding new solution by using either Equation (18) or (19). If the value of one current solution is smaller than the predetermined parameter Tol, Equation (19) is applied for updating such considered solution x. Otherwise, Equation (18) is a better option. The steps of the modified algorithm are similar to the

```
If   ε5 < Pa
        If   ΔFFRx < Tolx
            Solnewx = Solx + ε4.(Solrand1 − Solrand2 + Solrand3 − Solrand4)
        else
            Solnewx = Solx + ε4.(Solrand1 − Solrand2)
        end
else
    Solnewx = Solx
    End
End
```

Algorithm 1: New mutation technique applied in the proposed ICSA approach.

conventional CSA, except that an additional step should be added at each iteration. In this step, the ΔFF_d of all individual solutions should be calculated by utilizing Equation (24) and then the result of comparing the ratio with Tol will be used to decide which updating formula should be selected. The whole description of the proposed standard and new mutation technique can be coded in Matlab program language by using Algorithm 1.

3.2.3. Adjustment of Tolerance for Each Solution. As pointed out above, the proposed method needs assistances to determine the most appropriate step size for finding out favorable solution zones. The given aim can be reached if the selection of Tol_x is reasonable; however, the range of this parameter is infinite and hard to select. Thus, the adaptation of tuning the parameter is really necessary. First of all, the comparison between Tol_x and ΔFFR_x is carried out and then the adaptation will be determined based on the obtained result from the comparison. Results of comparison between the two parameters can be either ΔFFR_x is less than Tol_x or ΔFFR_x is higher than Tol_x. The case that two parameters are equal hardly ever occurs.

As the comer assumption happens (i.e., ΔFFR_x is less than Tol_x) at the considered time, the four-point step size will be employed for the xth solution. If Tol_x remains unchanged at the previous value, the identification of improvement from such four-point step size or two-point step size is vague. Consequently, value of Tol_x must be automatically reduced to a lower value in case that it has significant contribution to found promising solution of previous iteration. Clearly, the decrease of Tol_x can enable the proposed method to jump out local optimal zone and approach more effective zones. By trial and error method, Tol_x is selected to be a function of itself that is 0.9 of the previous value. Finally, the implementation of the proposed ICSA approach is presented in Algorithm 2.

4. The Application of the Proposed ICSA for OLD Problem

The whole computation steps of the proposed ICSA approach for solving OLD problem are explained as follows.

4.1. Handling Constraints and Randomly Producing Initial Population. As shown in Section 2, the considered OLD problem takes five following constraints into account:

(i) Power balance constraint is shown in Equation (4)

(ii) Power output limitation constraint is shown in Equation (6)

(iii) Prohibited power zone constraint is shown in Equation (9)

(iv) Real power reserve constraint is shown in Equation (10)

(v) Ramp rate limit constraint is shown in Equation (13)

Among the five constraints, ramp rate limit, generation limit, and prohibited power zone seem to be more complicated than power balance and power reserve constraints. However, the three constraints can be solved more easily because each unit is constrained independently in the three constraints whereas power balance constraint and power reserve constraint consider all the thermal generating units simultaneously. Power reserve constraint can be handled by penalizing the total generation of all units while power balance constraint can be solved by penalizing one violated thermal generating unit. The whole computation procedure for solving all constraints and calculating fitness function of solutions is described in detail as follows:

Step 1. Redefine maximum and minimum power output of each thermal generating unit as considering PPZ and RRL constraints by using the following formulas:

$$P_{i,\max} = \begin{cases} P_{i,\max} & \text{if } P_{i,\max} \le P_{i,0} + ISS_i \\ P_{i,0} + ISS_i & \text{if } P_{i,\max} > P_{i,0} + ISS_i; \end{cases} \quad i = 1, \ldots, N \tag{25}$$

$$P_{i,\min} = \begin{cases} P_{i,\min} & \text{if } P_{i,\min} \ge P_{i,0} - DSS_i \\ P_{i,0} - DSS_i & \text{else}; \end{cases} \quad i = 1, \ldots, N \tag{26}$$

Produce initial population with N_{ps} solutions ($Sol_1, Sol_2, \ldots, Sol_x, \ldots, Sol_{Nps}$)
Calculate fitness function ($FF_1, FF_2, \ldots, FF_x, \ldots, FF_{Np}$)
Go to the loop algorithm by setting $G = 1$
While ($G_{max} > G$)
% *(i) The first newly produced solutions*
$$Sol_{newx} = Sol_x + \alpha(Sol_x - Sol_{Gbest}) \oplus Lévy(\beta)$$
% *(ii) Perform selection approach*
$$Sol_x = \begin{cases} Sol_x & if\ FF_x < FF_{newx} \\ Sol_x^{new} & else \end{cases}; \quad x = 1, \ldots, N_{ps}$$
$$FF_x = \begin{cases} FF_x & if\ FF_x \le FF_{newx} \\ FF_{newx} & otherwise \end{cases}; \quad x = 1, \ldots, N_{ps}$$
% *(iii) The second newly produced solutions*
If $\varepsilon_6 < P_a$
If $\Delta FFR_x < Tol_x$
$Tol_x = Tol_x.0.9;$
$Sol_{newx} = Sol_x + \varepsilon_4.(Sol_{rand1} - Sol_{rand2} + Sol_{rand3} - Sol_{rand4})$
Else
$Tol_x = Tol_x$ & $Sol_{newx} = Sol_x + \varepsilon_4.(Sol_{rand1} - Sol_{rand2})$
End
Else
$Sol_{newx} = Sol_x$
End
% *(iv) Perform selection approach*
$$Sol_x = \begin{cases} Sol_x & if\ FF_x < FF_{newx} \\ Sol_x^{new} & else \end{cases}; \quad x = 1, \ldots, N_{ps}$$
$$FF_x = \begin{cases} FF_x & if\ FF_x \le FF_{newx} \\ FF_{newx} & otherwise \end{cases}; \quad x = 1, \ldots, N_{ps}$$
% *(v) Determine the most effective solution and its fitness*
Determine FF_x with the smallest value and assign Sol_x to Sol_{Gbest}
If $G_{max} > G$, perform step (i) and increase G to $G + 1$. Otherwise, stop the loop algorithm and report both the smallest fitness together with Sol_{Gbest}.
End while

Algorithm 2: The proposed ICSA approach.

$$P_{i,\max} = \begin{cases} P_{i,k}^l & if\ P_{i,k}^l < P_{i,\max} < P_{i,k}^u\ \&\ P_{i,\max} < P_{i,k}^u \\ P_{i,\max} & else; \end{cases} \quad i = 1, \ldots, N \tag{27}$$

$$P_{i,\min} = \begin{cases} P_{i,k}^u & if\ P_{i,k}^l < P_{i,\max} < P_{i,k}^u\ \&\ P_{i,\max} > P_{i,k}^l \\ P_{i,\min} & else; \end{cases} \quad i = 1, \ldots, N \tag{28}$$

Among the four Equations, (25) and (26) are used first in order to redefine upper bound and lower bound for all thermal generating units as considering RRL constraint. The, the redefined bounds continue to be redefined for the second time by using (27) and (28) as considering PPZ constraints.

Step 2 (randomly produce initial population). For dealing with the power balance constraint, all available units are separated into two groups in which the first group with decision variables consists of the power output from the second unit to the last unit ($P_2, P_3, \ldots, P_N$) meanwhile only the power output of the first unit (P_1) belongs to the second group with dependent variable. So, upper bound solution Sol_{max} and lower bound solution Sol_{min} must be defined as follows:

$$\begin{aligned} Sol_{\min} &= [P_{2,\min}, P_{3,\min}, \ldots, P_{N,\min}] \\ Sol_{\max} &= [P_{2,\max}, P_{3,\max}, \ldots, P_{N,\max}] \end{aligned} \tag{29}$$

Based on the upper bound solution and lower bound solution, each solution Sol_x is initially produced by the following model:

$$Sol_x = Sol_{\min} + \text{rand}\left(Sol_{\max} - Sol_{\min}\right), \quad x = 1, \ldots, N_{ps} \tag{30}$$

Step 3. Handle prohibited power zone constraint for decision variables $P_2, P_3, \ldots, P_N$

After being randomly produced, there is a high possibility that decision variables fall into PPZ and they violate PPZ constraint. So, the verification of falling into PPZ and correction of the violation should be accomplished by using the following formula:

$$P_i = \begin{cases} P_{ik}^l & \text{if } P_{ik}^l < P_i \le \dfrac{P_{ik}^l + P_{ik}^k}{2} \\ P_{ik}^u & \text{if } \left(P_i > \dfrac{P_{ik}^l + P_{ik}^k}{2}\right) \;\&\; \left(P_i < P_{ik}^u\right) \\ P_i & else \end{cases}; \qquad (31)$$

$$i = 2, \ldots, N \;\&\; k = 1, \ldots, n_i$$

Step 4. Handle RPB constraint by calculating P_1 and penalizing P_1 if it violates constraints.

In this step, power balance constraint is exactly handled by calculating and penalizing dependent variable (P_1). P_1 is obtained by using formulas (4) and (5) as follows:

$$P_1 = \frac{-\left(B_{01} - 1 + 2\sum_{i=2}^{N} B_{1i} P_i\right) \pm \sqrt{\Delta}}{2B_{11}} \qquad (32)$$

where

$$\Delta = \left(B_{01} - 1 + 2\sum_{i=2}^{N} B_{1i} P_n\right)^2 - 4B_{11}\left(P_D - \sum_{i=2}^{N} P_i + B_{00} + \sum_{i=2}^{N} B_{0i} P_i + \sum_{i=2}^{N}\sum_{j=2}^{N} P_i B_{ij} P_j\right); \quad \& \; \Delta \ge 0 \qquad (33)$$

In Equation (32), P_1 has been determined for the purpose of dealing with real power balance constraint. However, it is not sure that P_1 can satisfy upper bound and lower bound constraints and prohibited power zone constraints. So, P_1 must be checked and penalized.

Firstly, P_1 is checked and penalized for upper and lower bound constraints by the following model:

$$\Delta P_{1,x} = \begin{cases} 0 & if\; P_{1,\min} \le P_{1,x} \le P_{1,\max} \\ P_{1,\min} - P_{1,x} & if\; P_{1,\min} > P_{1,x} \\ P_{1,x} - P_{1,\max} & if\; P_{1,\max} < P_{1,x} \end{cases} \qquad (34)$$

In Equation (34), if the second case or the third case occurs, it means P_1 has violated either lower bound or upper bound and it would be penalized by using either ($\Delta P_{1,x} = P_{1,min} - P_{1,x}$) or ($\Delta P_{1,x} = P_{1,x} - P_{1,max}$). Otherwise, if P_1 has not violated the bound constraints (i.e., the first case in (34) happened), P_1 would continue to be checked for PPZ constraint by the following model:

$$\Delta P_{1,x} = \begin{cases} P_1 - P_{1k}^l & if\; P_{1k}^l < P_1 \le \dfrac{P_{1k}^l + P_{1k}^k}{2} \\ P_{1k}^u - P_1 & if\; \left(P_1 > \dfrac{P_{1k}^l + P_{1k}^k}{2}\right) \;\&\; \left(P_1 < P_{1k}^u\right) \\ 0 & else \end{cases} \qquad (35)$$

Step 5. Handle real power reserve constraint (10).

First of all, S_i is determined by using (11) and (12) and then the xth solution will be checked and penalized if power output of all thermal generating units cannot satisfy RPR constraint. The penalty for violation of the constraint can be calculated by using equation (36).

$$\Delta S_{R,x} = \begin{cases} 0 & \text{if } \sum_{i=1}^{N} S_{i,x} \ge S_R \\ S_R - \sum_{i=1}^{N} S_{i,x} & else \end{cases} \qquad (36)$$

As a result, real power reserve constraint can be solved by using the penalty method.

4.2. Calculate Fitness Function for Solutions. Fitness function of each solution is used to evaluate quality of solution. Normally, the function is the sum of objective function and penalty of violating constraints and is obtained by

$$FF_x = \sum_{i=1}^{N} F_i\left(P_{i,x}\right) + K.\left(\Delta S_{R,x}\right)^2 + K.\left(\Delta P_{1,x}\right)^2 \qquad (37)$$

4.3. The First Newly Updated Solutions by Lévy Flights Technique. In this section, the first newly updated solutions are performed by employing Lévy flights technique using Equation (14). However, each new solution can be out of their feasible operating zone such as PPZ and upper and lower limitations. When the power output violates its PPZ constraints, Equation (31) will be applied to tackle the constraint. Besides, the following equation will be employed when power output is higher or lower than their limitations.

$$Sol_x = \begin{cases} Sol_{\max} & \text{if } Sol_{\max} < Sol_x \\ Sol_{\min} & \text{if } Sol_{\min} > Sol_x \\ Sol_x & \text{Otherwise} \end{cases}; \quad x = 1, \ldots, N_p \qquad (38)$$

After that, Equations (32)-(37) are performed for determining all variables and penalty terms. Finally, Equation (38) is employed to calculate fitness function.

4.4. The Second Newly Updated Solutions by Using Mutation Technique. The second newly updated solutions are accomplished as presented in Section 3 above. Similar to the task after doing the first update, each solution in the new population must satisfy PPZ constraint and upper and lower boundaries by considering Equations (31) and (38). Then, Equations (32)-(37) are performed for determining all variables and penalty terms. Finally, Equation (38) is employed to calculate fitness function and the solution with the best value is assigned to the best one, Sol_{Gbest}.

4.5. Criterion of Stopping the Loop Algorithm. In the loop algorithm of using the proposed ICSA approach, the solution search work is stopped in case that the predetermined maximum iterations G_{max} is reached. For each search termination, the most effective solution is stored and another run continues to be accomplished until the predetermined number of runs is reached. After finishing the runs, the best one is found and reported. In addition, other values such as the fitness of the worst solution and average fitness of all solutions are also reported for comparing with other methods.

4.6. The Whole Iterative Process. The whole iterative algorithm for implementing the proposed ICSA approach for coping with OLD problem is described in detail in Figure 3.

5. Results and Discussions

The proposed ICSA approach performance has been investigated on six cases with different fuel options, different fuel characteristics, and complicated constraints. The details of the studied cases are presented as follows.

Case 1. Four systems with single fuel source (SFS) and power loss (PL) constraint

Subcase 1.1: A 3-unit system [57]

Subcase 1.2: A 6-unit system [57]

Subcase 1.3: A 3-unit system [56]

Subcase 1.4: A 6-unit system [56]

Case 2. A 110-unit system with SFS [57]

Case 3. Four systems with SFS and the effects of valve loading process (EoVLP)

Subcase 3.1: A 3-unit system supplying to a load of 850 MW [58]

Subcase 3.2.: A 13-unit system supplying to a load of 1,800 MW [1]

Subcase 3.3: A 13-unit system supplying to a load of 2,520 MW [1]

Subcase 3.4: A 40-unit system supplying to a load of 2,500 MW [1]

Subcase 3.5: An 80-unit system supplying to a load of 4,100 MW [49]

Case 4. Two systems with SFS and PPZ and RPR constraints

Subcase 4.1: A 60-unit system supplying to a 10,600 MW load [9]

Subcase 4.2: A 90-unit system supplying to a 15,900 MW load [9]

Case 5. A 15-unit system with SFS and RRL, PPZ, and PL constraints [61]

Case 6. Three systems with multiple fuel sources (MFS) and EoVLP

Subcase 6.1: An 80-unit system supplying to a 21,600 MW load [15]

Subcase 6.2: A 160-unit system supplying to a 43,200 MW load [15]

Subcase 6.3: A 320-unit system supplying to an 86,400 MW load [54]

For each considered case with each load case, the proposed ICSA approach is run 50 times on the program language of Matlab and a PC with 4 GB of RAM and 2.4 GHz processor. The selection of adjustment parameters including P_a and Tol_x is carefully considered to obtain the best optimal solutions meanwhile two others such as N_{ps} and G_{max} are chosen corresponding to the scale of particular test system. 9 values with the change of 0.1 in the range $[0.1, 0.9]$ are in turn selected for P_a while Tol_x is 0.01 at the beginning. The information including load demand, N_{ps}, G_{max}, and the best P_a is reported in Table 1.

5.1. Obtained Results on Case 1 considering Four Systems with SFS and PL Constraint. In this section, we have implemented the proposed ICSA approach for solving four systems divided into four subcases. Tables 2 and 3 show the comparisons of obtained results from Subcases 1.1 and 1.2 and Subcases 1.3 and 1.4, respectively. As listed in Table 2, the proposed ICSA method and CCSA can find equal fuel cost for Subcases 1.1 whereas the reduction of fuel cost from the proposed ICSA method as compared to CCSA is clearer for Subcase 1.2. As shown in Table 3 for comparing the proposed ICSA and three methods consisting of CCSA, ABC, and FA, the minimum fuel cost of the proposed ICSA is approximately equal to that of these methods for Subcases 1.3 but much less than that of these methods for Subcase 1.4. Furthermore, the proposed ICSA has been run by setting N_{ps} and G_{max} to 5 and 20 but these values were much higher for CCSA, ABC, and FA. They are 20 and 5000 for CCSA, 40 and 100 for ABC, and 20 and 5000 for FA. Consequently, the proposed method is very efficient for Case 1 with four subcases.

Optimal solutions obtained by ICSA for Case 1 are shown in Tables 16–18.

5.2. Obtained Results on Case 2 considering 110-Unit System with SFS. In this section, we have employed a very large scale system with 110 units but there were not challenges for objective function and complex constraints since EoVLP and constraints were not taken into account. Both CCSA and the proposed ICSA methods have been run for comparing with BBO, hybrid BBO and DE (DE/BBO), and Oppositional real coded chemical reaction optimization algorithm (ORCCROA) in [36], IWA in [40], and AGWO in [52]. As

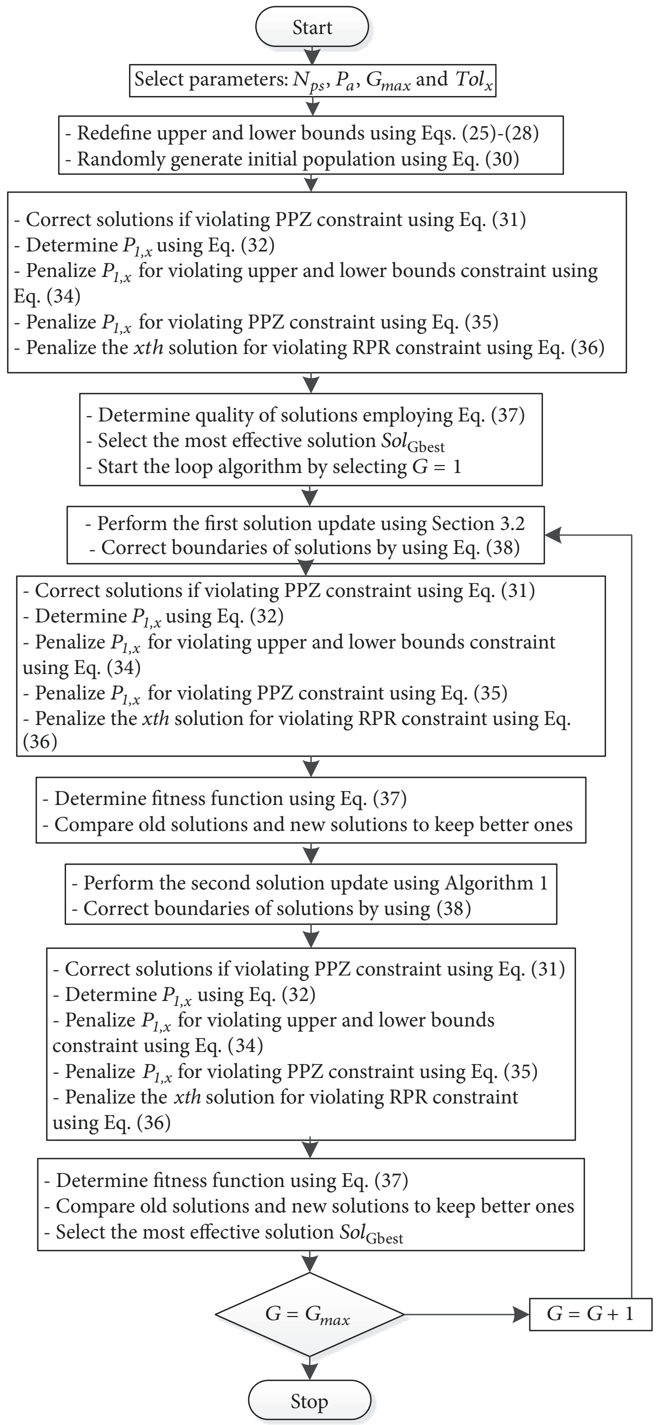

Figure 3: All computation steps for solving OLD problem by employing the proposed ICSA approach.

shown in Table 4, AGWO [52] has reached less fuel cost than ICSA; however, the exact fuel cost, which was recalculated by using reported solution, pointed out that the method has reached a very high fuel cost of \$215740.4250. For comparison with other methods, ICSA has found less fuel cost than all these methods. Particularly, the reduction of generation fuel cost is significant as compared to BBO, DE/BBO, and CCSA. Execution time comparisons are also useful evidence for indicating the high performance of ICSA. Thus, it can conclude that ICSA is a strong method for Case 2.

Optimal solution obtained by ICSA for the case is shown in Table 19.

5.3. Obtained Results on Case 3 considering Four Systems with SFS and EoVLP. In this section, the real performance of the proposed ICSA approach has been investigated based on five

Table 1: Information of considered cases and adjustment parameters.

Case	Fuel cost function	Constraint	Subcase	No. of units	P_D (MW)	N_{ps}	G_{max}	Best P_a
1	SFS	PL	1.1	3	350, 400, 450, 500, 550, 600, 650, 700	10	15	0.9
			1.2	6	600, 650, 700, 750, 800, 850, 900, 950	10	25	0.9
			1.3	3	150	10	15	0.9
			1.4	6	700	10	25	0.9
2	SFS	-	2	110	15,000	10	1,000	0.9
3	SFS, EoVLP	-	3.1	3	850	5	20	0.5
			3.2	13	1,800	10	5,000	0.9
			3.3	13	2,520	10	5,000	0.6
			3.4	40	2,500	10	6,000	0.9
			3.5	80	4,100	20	6,000	0.9
4	SFS	PPZ, RPR	4.1	60	10,600	10	1,200	0.9
			4.2	90	15,900	10	1,500	0.9
5	SFS, and	RRL, PPZ, PL	5	15	2,630	20	400	0.5
6	MFS, EoVLP	-	6.1	80	21,600	10	4,000	0.1
			6.2	160	43,200	10	6,000	0.1
			6.3	320	86,400	10	9,000	0.1

Table 2: Comparison of minimum cost ($) obtained by different methods for Subcases 1.1 and 1.2.

Subcase 1.1			Subcase 1.2		
P_D (MW)	CCSA [57]	ICSA	P_D (MW)	CCSA [57]	ICSA
350	18564.5	18564.484	600	32094.7	32090.3906
400	20812.3	20812.2936	650	34482.6	34478.8747
450	23112.4	23112.3635	700	36912.2	36908.7027
500	25465.5	25465.4692	750	39384	39378.4477
550	27872.4	27872.4051	800	41896.9	41889.5633
600	30334	30333.9858	850	44450.3	44442.193
650	32851	32851.0461	900	47045.3	47036.1336
700	35424.4	35424.442	950	49682.1	49671.5358

Table 3: Comparison of minimum cost ($) obtained by different methods for Subcases 1.3 and 1.4.

Subcase	1.3			1.4		
Method	Cost ($)	N_{ps}	G_{max}	Cost ($)	N_{ps}	G_{max}
CCSA [56]	1600.46	20	5000	8356.06	20	5000
ABC [56]	1600.51	40	100	8372.27	40	100
FA [56]	1600.47	20	5000	8388.45	20	5000
ICSA	1599.984	5	20	8313.221	5	20

Table 4: Comparison of results obtained by different methods for Case 2.

Method	Minimum cost ($)	Mean cost ($)	Maximum cost ($)	Time (s)	N_{ps}	G_{max}	Recalculated cost ($)
BBO [36]	198241.2	197213.5	199102.6	208	-	400	∗
DE/BBO [36]	198231.1	198326.7	198828.6	184	-	400	∗
ORCCROA [36]	198016.3	198016.3	198016.9	60	-	400	199881.4
OIWO [40]	197989.1	197989.4	197989.9	31	-	250	197989.1
AGWO [52]	197988∗	197988	197988	-	-	-	215740.42
CCSA	198007.42	198150.1	198310.0	7.2	10	1000	198007.423
ICSA	197988.93	198011.9	198337.0	7.6	10	1000	197988.938

∗ Denote that recalculated cost cannot be obtained because solutions were not reported.

Table 5: Comparisons of obtained results for Subcase 3.1.

Method	Minimum cost ($)	Recalculated cost ($)	N_{ps}	G_{max}
MCSA [58]	8218.8585	8352.1956	10	100
CCSA	8234.27	8234.27	5	20
ICSA	8234.07	8234.07	5	20

Table 6: Result comparisons for Subcases 3.2 and 3.3.

Approach	Subcase 3.2		Subcase 3.3		Computer (Processor-Ram)
	Min. cost ($/h)	Time (s)	Min. cost ($/h)	Time (s)	
CEP [1]	18048.21	294.96	-	-	
FEP [1]	18018	168.11	-	-	0.35 Ghz-1.128 GB
IFEP [1]	17994.07	157.43	-	-	
DE [10]	17963.83	-	24169.92	-	-
GA-MU[15]	17963.985	8.28	-	-	0.7 GHz
QPSO[16]	17969.01	-	-	-	-
GA-PS-SQP [30]	17964.25	11.06	-	-	256 MB-1 GHz
PSO-SQP[32]	17969.93	33.97	24261.05	-	-
SOS [34]	17963.8292	0.85	24169.917	0.71	2.2 GHz- 4 GB
MSOS [34]	17963.8292	0.81	24169.917	0.65	2.2 GHz- 4 GB
CEA-SQT [38]	17963.85	34.33	24169.94	34.25	3 GHz-2-GB
TSBO [39]	17963.8292	-	-	-	-
IWA [40]	17963.83	5.3	-	-	3.06 GHz-1 GB
CBA [44]	17963.83	0.97	-	-	3.3 GHz -4 GB
Mβ-HCLSA [49]	17,960.97$*$	803.4	24,164.18$**$	746.71	2.4 Ghz- 2 GB
IABCA [50]	17,972.96	-	-	-	-
CCSA [59]	17963.83	3.7	24169.917	3.8	2 GHz- 2 GB
OSE-CSA [59]	17963.83	3.85	24169.917	2.8	2 GHz- 2 GB
Proposed method	17963.83	0.95	24169.917	0.95	2.4 GHz- 4 GB

$*$ Recalculated cost is $17,969.1; $**$ recalculated cost is $24,173.88.

subcases with the gradual increase of number of units. The smallest scale system considers 3 units but the largest scale system takes 80 units. In addition to the implementation of the proposed ICSA, we have also implemented CCSA for Subcase 3.1 and Subcase 3.5 for further comparison because CCSA has not been run for the two subcases so far.

Comparison of obtained results from Subcase 3.1 shown in Table 5 indicates that the proposed ICSA is superior to CCSA with lower fuel cost but it seems to be less effective than MCSA [58]. The minimum cost of MCSA reported in [58] is the smallest fuel cost but the recalculated cost is much higher than that of the proposed method. Furthermore, MCSA has been implemented by setting very high values to N_{ps} and G_{max}.

Reports for Subcases 3.2 and 3.3 shown in Table 6 are the comparisons of the proposed ICSA approach and other methods such as conventional Evolution programming (CEP) [1], Fast EP (FEP) [1], improved FEP (IFEP) [1], DE [12], multiplier Lagrange-based genetic algorithm with (GA-MU) [15], QPSO [16], GA-PS-SQP [30], PSO-SQP [32], Mβ-HCLSA [49], IABCA [50], CCSA [59], OSE-CSA [59], SOS [34], MSOS [34], CEA-SQT [38], TSBO [39], IWA [40], and CBA [44]. As observed from the table, ICSA approach obtains better solutions than most methods excluding DE [10], CCSA [59], OSE-CSA [59], SOS [34], MSOS [34], CEA-SQT [38], TSBO [39], IWA [40], and CBA [44], especially Mβ-HCLSA [49] with lower cost, $17,960.97. However recalculated cost from reported solution of Mβ-HCLSA is $17,969.1. Besides, ICSA is very fast as compared to most methods where two other versions of Cuckoo search algorithm, CCSA and OSE-CSA, are also included except two methods in [34]. The processor of computer that all the methods run on is also reported in the final column. Clearly, ICSA approach is very efficient for the case with the 13-unit system where effects of valve loading process are considered.

In Subcase 3.4, the number of units is much larger than that of three subcases above, up to 40 units [1]. The obtained result comparisons with others are indicated in Table 7. Clearly, the minimum cost comparisons reveal that the proposed method is one of the leading methods due to the lowest cost except the comparison with CCSA [23], OSE-CSA [59], SOS [34], MSOS [34], EMA [45], θ-MBA [47], and AGWOA [52]. It is noted that AGWOA [52] has reported the best minimum cost with $121404.30 but recalculated minimum cost, which was obtained by substituting reported optimal generation of all thermal generating units, is $121,413.31. The average and the maximum costs from the proposed method do not belong to the leading method group; however, the

Table 7: Result comparisons for Subcase 3.4.

Approach	Best cost ($/h)	Average cost ($/h)	Worst cost ($/h)	CPU time (s)	Computer (Processor-Ram)
IFEP [1]	122,624.35	123,382.00	125,740.63	1167.35	0.35 Ghz - 0.128 GB
QPSO[16]	121,448.21	122,225.07	-	-	-
CCSA [23]	121,412.5355	121,520.41	121,810.25	3.03	2.1 Ghz-2 GB
BBO [28]	121,426.9530	121, 508.03	121,688. 66	1.1749	2.3 Ghz-512-MB
GA-PS-SQP [30]	121,458.00	122,039.00	-	46.98	3.1 Ghz-256 MB
PSO–SQP [32]	122,094.67	122,245.25		733.97	-
SOS [34]	121412.5355	121,414.9546	121435.5918	-	2.2 Ghz - 4 GB
MSOS [34]	121412.5355	121,412.5355	121412.5355	18.13	2.2 Ghz - 4 GB
EPSO [37]	121412.5702	121,455.7003	121709.5582	50 47.24	2.5 Ghz
IWA [40]	121,412.54	-	-	-	3.06 Ghz - 1 GB
IDE [41]	121,442.2682	121,448.8196	121457.2746	7.96	3.0 Ghz
IAPR [42]	121,436.9729	121,648.4401	122492.7018	1.092	2.83 Ghz - 4 GB
CCDE [43]	121,412.6858	121,412.8507	121413.6302	-	-
CBA [44]	121,412.5468	121,418.9826	121436.15	1.55	3.3 Ghz - 4 GB
EMA [45]	121,412.5355	121,417.1328	121426.1548	-	-
GRASP-DE [46]	121,414.621	121,736.025	122245.696	-	-
θ-MBA [47]	121,412.5355	121,412.948	121412.786	-	2.4 Ghz -1GB
TBHSA [48]	121,425.15	121,528.65	-	-	2.66 Ghz- 4 GB
Mβ-HCLSA [49]	121,414.68	121,496.84	-	951.94	2.66 Ghz- 4 GB
AGWOA [52]	121404.30∗	121412.30	121446.70	-	2.4 Ghz- 2 GB
OSE-CSA [59]	121,412.5355	121,472.45	121596.18	3.02	2 Ghz - 2 GB
Proposed method	121,412.5355	121,601.0759	122502.2623	1.46	2.4 Ghz- 4 GB

∗Recalculated minimum cost is $121,413.31.

Table 8: Result comparisons for Subcase 3.5.

Method	Best cost ($/h)	Average cost ($/h)	Worst cost ($/h)	CPU time (s)	Computer (Processor-Ram)
Mβ-HCLSA [49]	242,854.52	242,992.69	-	1279.70	2.66 Ghz- 4 GB
AGWOA [52]	242,824.50	242919.40	242986.8	-	2.4 Ghz- 2 GB
CCSA	242,824.50	243365.14	244036.75	3.2	2.4 Ghz- 4 GB
ICSA	242,820.4	243018.65	243876.17	3.3	2.4 Ghz- 4 GB

execution time that ICSA approach takes is smaller than most ones except IAPR [42] and θ-MBA [47], which have been run on stronger computers. Note that MSOS [34] has been faster than the proposed method for Subcases 3.1 and 3.2 above but it is too slower than the proposed method for the case, namely, 18.13 seconds compared to 1.46 seconds. Compared to two other versions of Cuckoo search, CCSA [23] and OSE-CSA [59], the proposed method is also faster about three times although the processors are slightly different. For this case, θ-MBA [47] shows a very good performance; however, the method has not been tested on more complicated systems and larger scale and therefore more comparisons with the method must end. Clearly, the proposed method is still efficient for the case where large scale and effects of valve loading process are included.

Subcase 3.5 is the largest scale system with 80 units. In addition to the implementation of ICSA, CCSA has been also run for the Subcase 3.5 for further investigation of efficiency improvement of the proposed ICSA approach. The comparisons of minimum cost in Table 8 show that the proposed ICSA can find more optimal solution than Mβ-HCLSA [49], AGWOA [52], and CCSA. The proposed ICSA is also superior to CCSA in terms of more stable search ability and lower fluctuation since its average cost and maximum cost are less than those of CCSA. The outstanding figure cannot be reached as compared to AGWOA [52]; however, it is hard to conclude AGWOA [52] is superior to the proposed ICSA approach about more stable search ability and lower fluctuation. Actually, comparison of the values of population and iterations as well as execution time cannot be accomplished because the information was not reported in [52]. Thus, it can conclude that the proposed ICSA is effective for the subcase.

Optimal solutions obtained by ICSA for the case are shown in Tables 20–23.

5.4. Obtained Results on Case 4 with Two Systems considering SFS, and PPZ, and RPR Constraints. In this section, two

Table 9: Result comparisons for Subcases 4.1 and 4.2.

Subcase	Approach	Best cost ($/h)	Mean cost ($/h)	Worst cost ($/h)	Std. dev. ($/h)	CPU time (s)	Computer (Processor-Ram)
4.1	GA [9]	-	131992.310	-	-	563.81	0.7 Ghz
	IGA-MU [9]	-	130180.030	-	-	162.58	0.7 Ghz
	CCSA [23]	130170.3949	130171.5986	130174.0722	0.7531	2.028	2.10 GHz-2.0 GB
	OSE-CSA [59]	130169.9367	130170.8301	130175.178	0.85519	3.4412	2.0 GHz- 2.0 GB
	ICSA	130169.962	130173.9	130233.5	8.6509	0.9153	
4.2	GA [9]	-	198831.690	-	-	940.93	0.7 Ghz
	IGA-MUM [9]	-	195274.060	-	-	255.45	0.7 Ghz
	CCSA [23]	195258.7847	195264.3818	195271.7057	2.4857	3.036	2.10 Ghz-2.0 GB
	OSE-CSA [59]	195255.9077	195257.5392	195262.3724	1.4674	5.3352	2.0 Ghz- 2.0 GB
	ICSA	195255.762	195260.2	195269.4	2.8002	1.5892	2.40 Ghz- 4.0 GB

Table 10: Comparisons of result obtained by different methods for Case 5.

Method	Minimum Cost ($)	CPU time (s)	N_{ps}	G_{max}	Computer (Processor-Ram)
MCSA [58]	32 706.7358	4.8	50	200	4.0 GHz- 1.0 GB
PSO [61]	32858	59.45	100	200	Pentium III-256 MB
GA [61]	33113	81.8	100	200	Pentium III-256 MB
CCSA	32 708.9	4.45	20	400	2.40 Ghz- 4.0 GB
ICSA	32 705.998	4.6	20	400	2.40 Ghz- 4.0 GB

test systems with SFS and PPZ and RPR constraints are considered. The test system size is up to 60 and 90 units for Subcases 4.1 and 4.2, respectively. Comparison for the cases is only performed with two Genetic algorithms consisting of GA and IGA-MU in [9] and two other versions of Cuckoo search algorithms including CCSA [23] and OSE-CSA [59] and presented in Table 9. Other studies have tended to ignore such complicated constraints of PPZ and RPR. The costs reported in Table 9 indicate that ICSA approach can obtain more effective solution than CCSA and OSE-CSA because it has reached lower minimum cost than the two ones. Furthermore, the proposed method also takes shorter computation time for the two cases from about two times to about three times although the processor of the proposed method is slightly stronger. The mean costs of ICSA approach are much less than those from IGA-MU and GA and slightly higher than those from OSE-CSA but there is a trade-off between the proposed method and CCSA for the two subcases. In fact, the proposed method obtains higher mean cost for Subcase 4.1 but lower cost for Subcase 4.2. Although GA methods have been run on a weak computer with 0.7 GHz of the processor compared to that with 2.4 GHz in the study, their execution times are significantly higher, namely, 563.81 seconds (GA) and 162.58 seconds (IGA-MU) compared to 0.9153 seconds of ICSA approach for Subcase 4.1 and 940.93 seconds (GA) and 255.45 seconds (IGA-MU) compared to 1.5892 seconds (the proposed method). The analysis can point out that ICSA approach is more efficient than these compared methods in terms of optimal solutions and execution time.

Optimal solutions obtained by ICSA for Subcase 4.2 are shown in Table 24.

5.5. Obtained Results on Case 5 with a 15-Unit System considering SFS and RRL, PPZ, and PL Constraints. In this section, a 15-unit system considering RRL, PPZ, and PL constraints is considered to be solved for finding optimal solution. For efficiency investigation of the proposed ICSA, we have also implemented CCSA for comparison. As listed in Table 10, the proposed ICSA is the most effective method with the smallest fuel cost. The comparisons of control parameters as well as CPU time are also good evidence to confirm the strong search of the proposed ICSA approach since it has been run by smaller values of control parameter and faster execution time as compared to all methods excluding CCSA.

Optimal solution obtained by ICSA for the case is shown in Table 25.

5.6. Obtained Results on Case 6 with Three Systems considering MFS and EoVLP. In this section, three test systems with the challenge on objective function including multi-fossil fuel sources and effects of valve loading process are considered. The scale is up to 80 units, 160 units, and 320 units for Subcases 6.1, 6.2, and 6.3, respectively.

Comparison for Subcase 6.1 reported in Table 11 reveals that the proposed method is the best method in terms of the lowest best cost, the lowest mean cost, and the lowest standard deviation and the fastest execution time. The processor from this proposed method is about four times stronger than CGA-MU and IGA-MU but the speed is from ten times to 35 times faster than these methods. Compared to CCSA and OSE-CSA, the proposed method is about two times faster but the processor is slightly stronger.

Comparison for Subcase 6.2 is reported in Table 12. Clearly, the proposed ICSA approach obtains better values of the best, mean, and worst costs than most methods except MSOS [34] where the best cost difference is about $ 0.24. However, the proposed method is the second fastest one with 11.19 seconds where the first fastest one, CBA [44], has

Table 11: Comparisons of found results for Subcase 6.1.

Approach	Best cost ($/h)	Mean cost ($/h)	Worst cost ($/h)	Std. dev. ($/h)	CPU time (s)	Computer (Processor-Ram)
CGA-MU [15]	-	5008.1426	-	-	309.41	0.7 GHz
IGA-MU [15]	-	5003.8832	-	-	85.67	0.7 GHz
CCSA [23]	4992.6853	4993.7307	5003.4294	1.0931	18.25	2.10 GHz-2.0 GB
OSE-CSA [59]	4992.4215	4994.4987	4995.6717	0.4939	15.24	2.0 GHz- 2.0 GB
ICSA	4990.867	4991.384	5007.106	2.2483	8.4828	2.40 GHz- 4.0 GB

Table 12: Comparisons of found results for Subcase 6.2.

Approach	Best cost ($/h)	Average cost ($/h)	Worst cost ($/h)	Std. dev. ($/h)	CPU time (s)	Computer (Processor-Ram)
CGA-MU [15]	-	10,143.7263	-	-	621.30	0.7 GHz
IGA-MU [15]	-	10,042.4742	-	-	174.62	0.7 GHz
CCSA [23]	9,990.6548	9,996.6390	10,014.0183	4.9268	75.42	2.1 GHz-2 GB
SOS [34]	10,121.20529	10,179.03436	10,241.07846	24.960	-	2.2 GHz-4 GB
MSOS [34]	9,981.3114	9,981.9798	9,983.6394	0.477	25.72	2.2 GHz-4 GB
TLBO [35]	10,006.0117	10,006.2821	10,005.994	0.069	48.216	2 GHz-1 GB
CRBA [36]	9,989.9444	99,92.050362	9,996.8317	-	67.50	-
CBA [44]	10,002.8596	10,006.3251	10,045.2265	9.5811	5.71	3.3 GHz -4 GB
OSE-CSA [59]	9,989.9444	9,992.0503	9,996.8317	1.4138	67.50	2 GHz- 2 GB
Proposed method	9,981.556	9,982.76	10,018.16	5.0904	11.19	2.4 GHz-4 GB

Table 13: Comparisons of found results for Subcase 6.3.

Approach	Best cost ($/h)	Average cost ($/h)	Worst cost ($/h)	Std. dev. ($/h)	CPU time (s)	Computer (Processor-Ram)
SOS [34]	20,081.0151	20,346.5908	20,470.2934	105.53	-	2.2 GHz-4.0 GB
MSOS [34]	19,962.6238	19,963.7147	19,965.2493	0.64	96.41	2.2 GHz-4.0 GB
CCSA [54]	19,964.17	19,976.39	19,982.76	16.64	59.82	3.06 GHz-1.0 GB
ICSA	19,962.8485	19,965.3410	19,996.5234	4.5673	17.1384	2.4 GHz-4.0 GB

spent 5.71 seconds. Clearly, MSOS is better than the proposed ICSA approach with respect to slightly less best cost but is worse than the proposed method in terms of execution time, namely, 25.72 seconds compared to 11.19 seconds while CBA [44] is faster than the proposed method but obtains significantly worse costs. The analysis can conclude that the proposed ICSA approach is very powerful for the subcase with 160 units.

Table 13 presents the comparison of three other methods including CCSA [54], SOS [34], and MSOS [34] accompanied with the proposed method for Subcase 6.3. The obtained result comparisons imply that ICSA approach can obtain better values of the best and standard deviation costs than CCSA and SOS but obtains slightly higher cost than MSOS by approximately $ 0.22. Besides, the execution time from ICSA approach is much shorter than others; especially, it is higher than five times faster than MSOS. The four methods have been run on approximately strong computers. Briefly, the proposed ICSA approach can find and converge to more favorable solution than other methods with shorter CPU time except the comparison with MSOS, which had better solution but spent higher than five times execution times. Consequently, the proposed method is a very promising optimization algorithm for Subcase 6.3, a system up to 320 units and with multi-fossil fuel sources and effects of valve loading process.

Optimal solution obtained by ICSA for Subcase 6.3 is shown in Table 26.

5.7. The Improvement of ICSA Approach Performance

5.7.1. The Outstanding Improvement over CCSA. In this section, the performance improvement of ICSA over CCSA has been investigated by analyzing obtained results and set control parameters. Table 14 has been formed by adding reduction cost, improvement level of the best cost, execution time, and control parameters consisting of N_{ps} and G_{max}. Among the compared factors, reduction cost is the deviation of the cost of CCSA and that of ICSA whereas the improvement level is the ratio of the reduction cost to the cost of CCSA. The reduction costs indicate the proposed method could find either equal quality of solutions or higher quality of solutions than CCSA for all study cases. The reduction cost is from $0 to $42.839 corresponding to the improvement level

Table 14: Summary of results obtained by CCSA and ICSA for all study cases.

Study case	Reduction cost ($)	Improvement level (%)	Execution time (s)		N_{ps}		G_{max}	
			CCSA	ICSA	CCSA	ICSA	CCSA	ICSA
Subcase 1.1	0	0.0000	-	0.08	-	10	-	15
Subcase 1.2	10.5642	0.0213	-	0.11	-	10	-	25
Subcase 1.3	0.476	0.0297	-	0.09	20	10	5,000	15
Subcase 1.4	42.839	0.8968	-	0.12	20	10	5,000	25
Case 2	18.485	0.0093	7.2	7.6	10	10	1,000	1,000
Subcase 3.1	0.2	0.0024	0.1	0.1	5	5	20	20
Subcase 3.2	0	0.0000	3.7	0.95	10	10	20,000	5,000
Subcase 3.3	0	0.0000	3.8	0.95	10	10	20,000	5,000
Subcase 3.4	0	0.0000	3.03	3.03	10	10	10,000	6,000
Subcase 3.5	4.1	0.0017	3.2	3.3	20	20	6,000	6,000
Subcase 4.1	0.4329	0.0003	2.1	0.92	10	10	1,500	1,200
Subcase 4.2	3.0227	0.0015	3.06	1.59	10	10	1,900	1,500
Case 5	2.902	0.0089	4.45	4.6	20	20	400	400
Subcase 6.1	1.82	0.0364	18.25	8.48	20	10	4,000	4,000
Subcase 6.2	9.0988	0.0911	75.42	11.19	20	10	6,000	6,000
Subcase 6.3	1.3215	0.0066	59.82	17.14	320	10	1,200	9,000

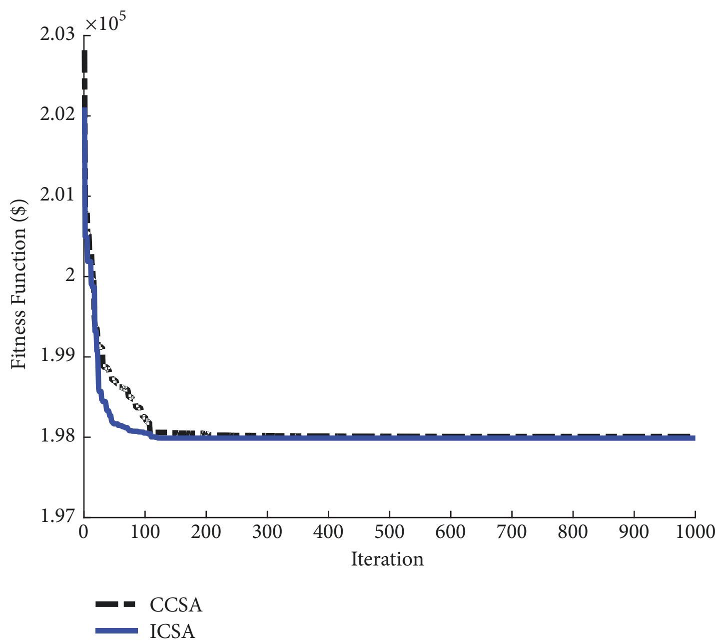

Figure 4: The best run obtained by CCSA and ICSA for Case 2.

that is from 0 to 0.8968%. The saving cost is not too much for one hour but the operation in one day, one month, or one year is very high. However, it should be noted that CCSA has been run by setting much higher population size and iterations for many cases excluding study cases implemented in the study such as Case 2, Subcase 3.1, Subcase 3.5, and Case 5. For instance, ICSA has used $N_{ps} = 10$ and $G_{max} = 15$ for Subcase 1.3, and $N_{ps} = 10$ and $G_{max} = 25$ for Subcase 1.4 whereas CCSA has been run by setting $N_{ps} = 20$ and $G_{max} = 5000$ for the two subcases. Similarly, CCSA has been run for Subcases 3.2, 3.3, and 3.4 with much higher number of iterations. For the last subcase, CCSA has been run by setting $N_{ps} = 320$ and $G_{max} = 1{,}200$ but those of ICSA have been 10 and 9,000. Due to the higher value of control parameters, CCSA has tended to spend more time in finding such high quality solutions for almost all study cases. Execution time of ICSA is less than 18 seconds while that of CCSA is up to higher 75 seconds. It is clear that the proposed ICSA could find better optimal solutions than CCSA for such considered systems. For further investigation of performance comparison, the best runs over 50 runs and fuel cost values of 50 runs obtained by CCSA and the proposed ICSA for Case 2, Subcase 3.1, Subcase 3.5, and Case 5 have been plotted in from Figures 4–11. The best run curves show the faster search of the proposed ICSA method

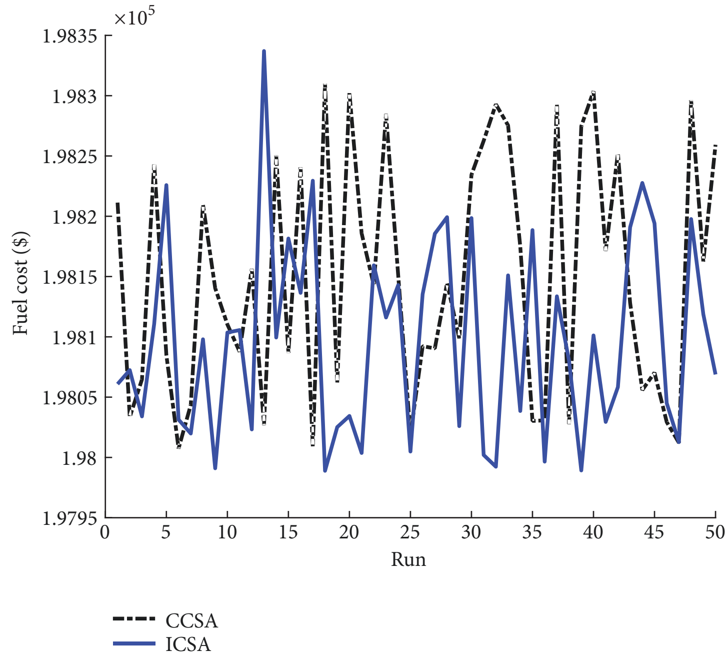

Figure 5: The best fuel cost of 50 runs obtained by CCSA and ICSA for Case 2.

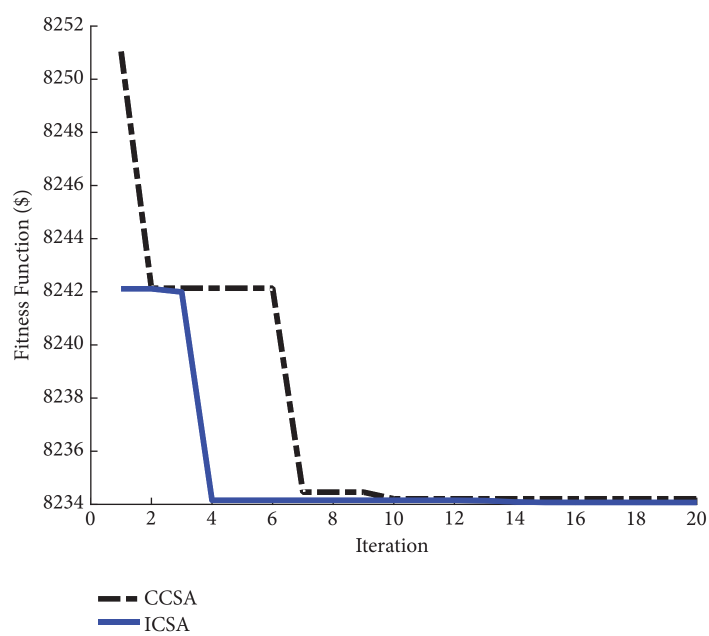

Figure 6: The best run obtained by CCSA and ICSA for Subcase 3.1.

whereas 50 values of fuel cost indicate that the proposed ICSA can find many solutions with better quality. Clearly, the proposed ICSA is outstanding in terms of stabilization of solution search and faster convergence. As a result, it can conclude that the proposed ICSA approach is more effective than CCSA in solving OLD problem with considered systems.

5.7.2. The Improvement of Results over Other Methods. In this article, we have tested ICSA approach on 6 cases with 16 systems with different fuel cost forms, different constraints, and different scale systems from 3 units to 320 units. We have compared the yielded results from ICSA approach and other existing ones for evaluating the efficiency of ICSA approach. In subsections above, we have shown yielded results from ICSA approach and compared these results to those of other ones. However, the demonstration had not been very good for observing and comparing to lead to a conclusion. Thus, in the subsection we have summarized the result comparisons of the

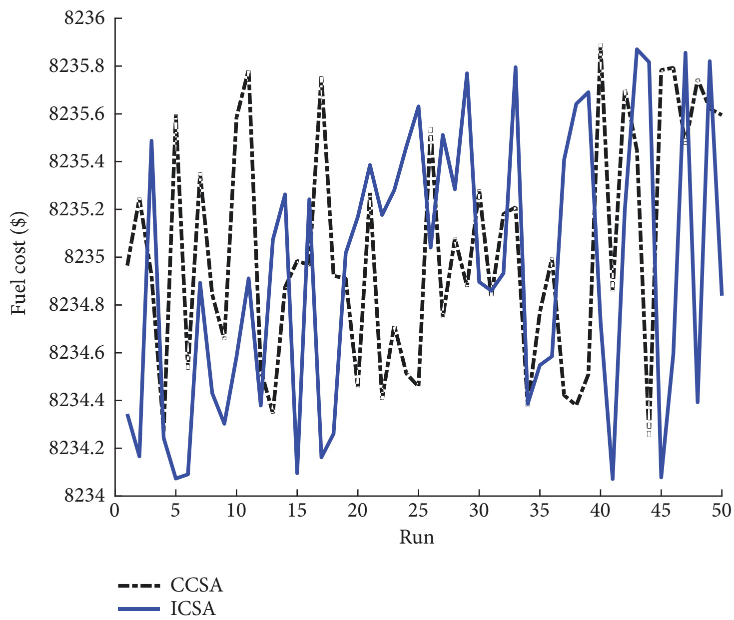

Figure 7: The best fuel cost of 50 runs obtained by CCSA and ICSA for Subcase 3.1.

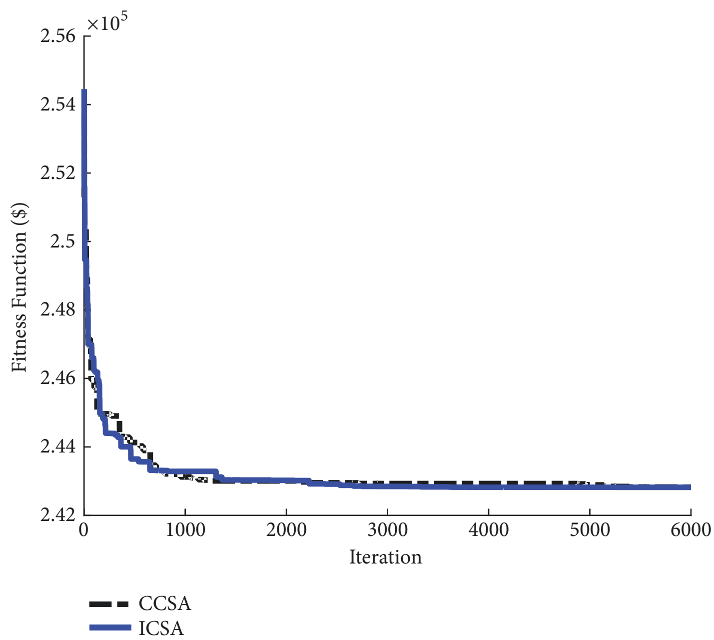

Figure 8: The best run obtained by CCSA and ICSA for Subcase 3.5.

proposed and other ones. Table 15 has reported the reduction cost (in $) of ICSA approach compared to other ones. In addition, we have converted the reduction cost into improvement level (in %) for better comparison. The improvement has been shown from the lowest level to the highest level in terms of reduction cost and improvement percentage. In addition, we have also given the slowest and the fastest execution time of other compared methods together with that of the proposed method. The table implies that ICSA approach can find better optimal solutions with less fuel cost up to $0.52 for Subcase 1.3, $75.229 for Subcase 1.4, $17751.49 for Case 2, $118.1256 for Subcase 3.1, $84.38 for Subcase 3.2, $91.13 for Subcase 3.3, $1211.81 for Subcase 3.4, $34.12 for Subcase 3.5, $0.4329 for Subcase 4.1, $3.0227 for Subcase 4.2, $407.002 for Case 5, $1.8183 for Subcase 6.1, $139.65 for Subcase 6.2, and $118.17 for Subcase 6.3. These reduction costs are equivalent to improvement level (IL) of 0.0329%, 0.8968%, 8.2282%, 1.4143%, 0.47%, 0.38%, 0.99%, 0.01405%,

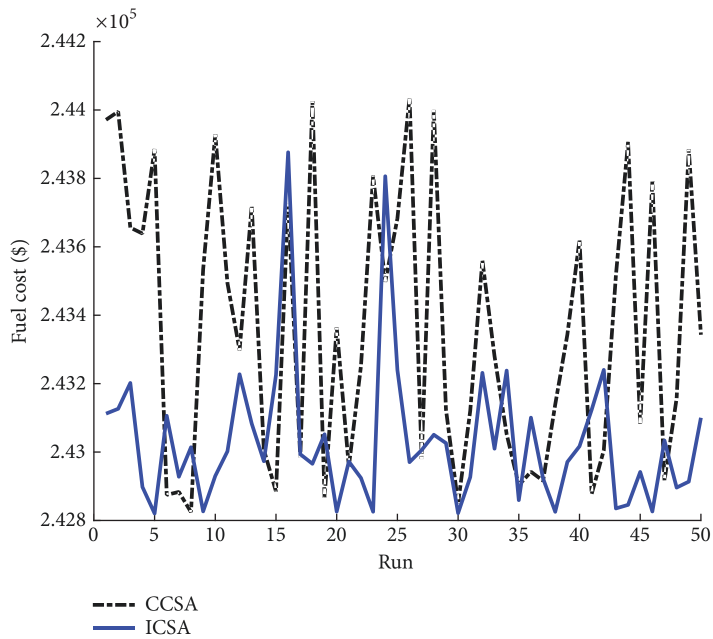

Figure 9: The best fuel cost of 50 runs obtained by CCSA and ICSA for Subcase 3.5.

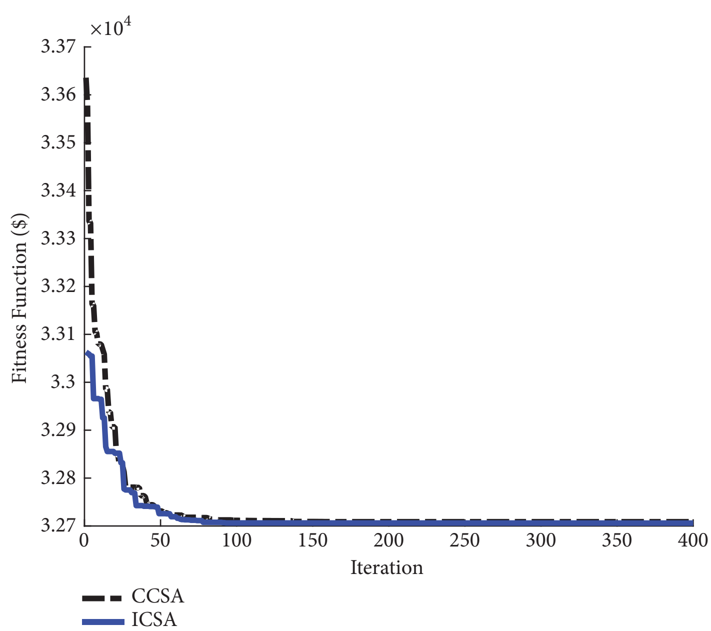

Figure 10: The best run obtained by CCSA and ICSA for Case 5.

0.0003%, 0.002%, 1.2291%, 0.04%, 1.38%, and 0.59%. These quantitative comparisons reveal that larger scale systems can lead to better reduction cost but the improvement level is not high because total cost of compared methods tends to be large for large scale systems. Furthermore, very large scale systems with nondifferentiable objective have been normally solved by strong methods. In fact, systems in Case 3 have the same characteristic with single fuel and effects of valve loading process but Subcase 3.4 is a larger scale system with 40 units while Subcases 3.1, 3.2, and 3.3 are constructed by 3 units, 13 units, and 13 units. So, the improvement percentage of Subcase 3.4 can be up to 0.99% whilst that of Subcases 3.2 and 3.3 is 0.47% and 0.38%, respectively. Subcase 3.5 is with the largest system, 80 units, but the reduction cost is not much, only $34.12 because compared methods with ICSA are either state-of-the-art ones or improved ones. Also, Subcase 4.1 and Subcase 4.2 have considered single fuel and PPZ and spinning reserve constraints but Subcase 4.2 is larger scale

Table 15: Performance improvement summary of the proposed method.

Study cases	Reduction cost ($)		Improvement level (%)		Execution time (s)		
	From	To	From	To	Slowest method	Fastest method	Proposed method
Subcase 1.3	0.486	0.526	0.0304	0.0329	-	-	0.09
Subcase 1.4	59.049	75.229	0.7053	0.8968	-	-	0.12
Case 2	0.17	17751.49	0.0001	8.2282	-	-	7.6
Subcase 3.1	0.2	118.1256	0.0024	1.4143	-	-	0.1
Subcase 3.2	0	84.38	0	0.47	294.96	0.85	0.95
Subcase 3.3	0	91.13	0	0.38	34.25	0.65	0.95
Subcase 3.4	0	1211.81	0	0.99	1167.35	1.55	1.46
Subcase 3.5	4.1	34.12	0.001688	0.01405	1279.7	-	3.3
Subcase 4.1	0	0.4329	0	0.0003	563.81	2.028	0.92
Subcase 4.2	0.1457	3.0227	0	0.002	940.93	3.036	1.59
Case 5	0.7378	407.002	0.0023	1.2291	81.8	4.8	4.6
Subcase 6.1	1.5545	1.8183	0.031	0.04	309.41	15.24	8.48
Subcase 6.2	0	139.65	0	1.38	75.42	5.71	11.19
Subcase 6.3	0	118.17	0	0.59	96.41	59.82	17.14

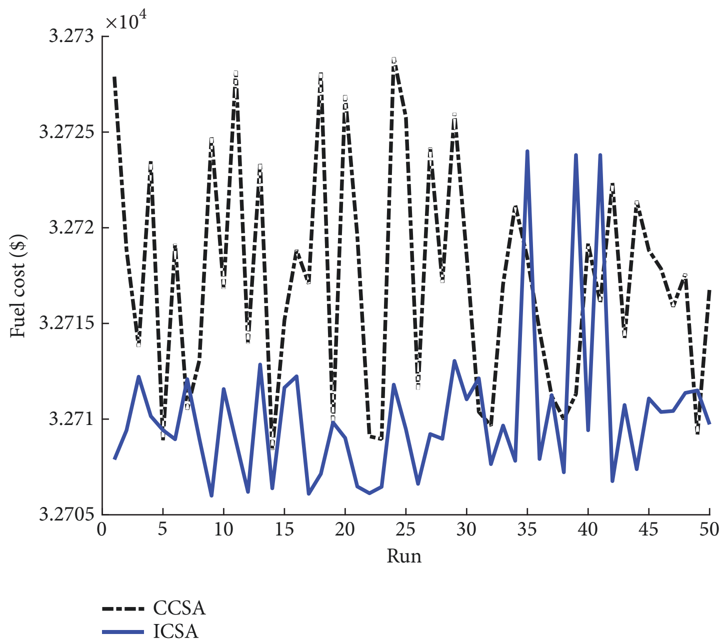

Figure 11: The best fuel cost of 50 runs obtained by CCSA and ICSA for Case 5.

system with 90 units and Subcase 4.1 is only with 60 units. Thus, the improvement of Subcase 4.1 is lower with 0.0003% but that of Subcase 4.2 is 0.002%. Similarly, systems in Cases 6 have the same feature with multi-fossil fuel sources and effects of valve loading process but they are, respectively, constructed by 80, 160, and 320 units. As a result, the improvements of Subcase 6.2, 1.38%, and Subcase 6.3, 0.59%, are much higher than Subcase 6.1, 0.04%. However, Subcase 6.2 with smaller number of units but getting higher improvement is easily understood because there were nine compared methods but only three compared methods are considered for Subcase 6.3. In general, the improvement is not high; it is about under one dollar, several dollars, tens of dollars, and over one thousand dollars per hour; however, the saving cost will be significant if the operation is considered to be one month with 720 hours or one year with 8760 hours.

Execution time comparison can be evaluated by observing the fastest and the slowest compared methods in Table 15. These execution times of the proposed method are approximately equal to that of the fastest methods for Subcases 3.2, 3.3, and 3.4, and much shorter than other fastest compared methods for other cases, especially for Subcases 6.1 and 6.3. The fastest method for Subcase 6.2 is CBA [44] showing 5.7 seconds while that of the proposed method is 11.19 seconds. However, it cannot conclude that CBA is more effective than the proposed method because the proposed method could

Table 16: Optimal solutions for Subcase 1.1.

P_D (MW)	P_1 (MW)	P_2 (MW)	P_3 (MW)	Cost (\$/h)
350	70.3452	156.2332	129.1983	18564.484
400	82.0693	174.9981	150.5007	20812.2936
450	93.9197	193.8184	171.8748	23112.3635
500	105.8786	212.7296	193.3062	25465.4692
550	117.9104	231.7314	214.835	27872.4051
600	130.0261	250.8404	236.4375	30333.9858
650	142.231	270.0616	258.1069	32851.0461
700	154.5059	289.3434	279.919	35424.442

Table 17: Optimal solutions for Subcase 1.2.

P_D (MW)	P_1 (MW)	P_2 (MW)	P_3 (MW)	P_4 (MW)	P_5 (MW)	P_6 (MW)	Cost (\$/h)
600	25.1066	10	95.5958	100.9756	203.2808	179.1733	25.1066
650	28.6187	10	104.5932	105.3786	218.2103	199.8954	28.6187
700	29.007	10.4656	116.8803	121.6634	224.5097	216.7929	29.007
750	29.9964	12.141	131.5535	130.3179	241.5602	226.5418	29.9964
800	33.7513	14.5234	142.2418	135.7029	255.3572	243.5787	33.7513
850	34.9093	18.0112	152.6422	141.9012	271.4989	259.4877	34.9093
900	39.3589	19.8696	162.8286	155.2237	282.8438	271.6499	39.3589
950	41.1506	25.2743	175.1899	162.2651	294.9102	286.5136	41.1506

Table 18: Optimal solutions for Subcases 1.3 and 1.4.

Variables	Subcase 1.3	Subcase 1.4
P_1 (MW)	33.4743	303.5073
P_2 (MW)	64.0977	113.7678
P_3 (MW)	55.094	143.4992
P_4 (MW)	-	50
P_5 (MW)	-	50
P_6 (MW)	-	50
Cost (\$/h)	1599.984	8313.221

find better optimal solution with less fuel cost by \$21.30. As considering execution time of the slowest compared methods, it can point out that ICSA is a very fast optimization tool since the execution time of these methods is 294.96 and 34.25 seconds for Subcases 3.2 and 3.3, 1167.35 seconds for Subcase 3.4, 563.81 seconds for Subcase 4.1, 940.93 seconds for Subcase 4.2, 1279.7 seconds for Case 5, 309.41 seconds for Subcase 6.1, 75.42 seconds for Subcase 6.2, and 96.41 seconds for Subcase 6.3 while the execution time of the proposed method for these cases is, respectively, 0.95, 1.46, 0.91, 1.5892, 4.6, 8.4828, 11.19, and 17.1384 seconds. It is clearly shown that ICSA is very fast as compared to these methods.

In summary, the proposed method has found approximately high quality solutions with several standard state-of-the-art meta-heuristic algorithms and improved versions of them together with other old methods. In addition, the proposed method could improve result better than approximately all methods with faster execution time. Compared to other methods with the fastest convergence speed and high quality solutions, the proposed method has been as fast as for some cases and much faster for other cases. The comparison with the slowest methods could show that the proposed method was extremely powerful since it was up to nearly one thousand times faster. Consequently, the proposed ICSA approach can be one of the strongest optimization tools for OLD problem.

6. Conclusions

This paper has proposed a good ICSA method for solving OLD problem in which many test systems with different objective functions and complicated constraints from simple to complex have been used as studied cases. The proposed ICSA method has been developed by performing several modifications on the second solution update of CCSA, which contained several drawbacks to global convergence and fast manner. The OLD problem has covered from single fuel to multi-fossil fuels, from quadratic objective function to nonconvex objective function in addition to PPZ, RPR, and RRL constraints. Many existing optimization algorithms have

Table 19: Optimal solutions for Case 2.

i	P_i (MW)	i	P_i (MW)	i	P_i (MW)	i	P_i (MW)	i	P_i (MW)
1	2.4	23	68.9	45	659.9972	67	70	89	82.1667
2	2.4002	24	349.991	46	617.3078	68	70	90	88.1748
3	2.4009	25	399.9924	47	5.4001	69	70	91	56.4408
4	2.4003	26	400	48	5.4015	70	359.9917	92	100
5	2.4003	27	499.9948	49	8.4001	71	399.9986	93	440
6	4.0003	28	499.9999	50	8.401	72	400	94	499.9999
7	4.0001	29	199.9718	51	8.4024	73	104.4979	95	600
8	4	30	100	52	12.0011	74	191.601	96	473.4828
9	4.0001	31	10.0018	53	12.0001	75	90	97	3.6004
10	64.0006	32	19.9875	54	12.0005	76	49.9996	98	3.6
11	62.9079	33	80	55	12	77	160.0246	99	4.4007
12	36.8567	34	250	56	25.2	78	296.6528	100	4.4006
13	57.1595	35	359.9919	57	25.2083	79	175.6282	101	10.001
14	25.001	36	399.9946	58	35.0001	80	96.7143	102	10.0013
15	25.0004	37	39.9782	59	35.0388	81	10.0008	103	20.0013
16	25	38	69.9974	60	45.0023	82	12	104	20.0018
17	154.9936	39	99.994	61	45.0005	83	20.2773	105	40.0035
18	154.9968	40	119.9999	62	45.0002	84	199.8833	106	40
19	154.9954	41	157.6538	63	184.9144	85	324.744	107	50.0008
20	154.9989	42	219.9983	64	184.9969	86	439.9977	108	30.0005
21	68.9	43	439.9987	65	184.98	87	14.3477	109	40.0008
22	68.9004	44	559.9991	66	184.9961	88	22.5542	110	20.0006

Table 20: Optimal solution for Subcase 3.1.

i	P_i (MW)
1	300.2468
2	400
3	149.7532
Cost ($/h)	8234.083

Table 21: Optimal solutions for Subcases 3.2 and 3.3.

Subcase 3.2		Subcase 3.3	
i	P_i (MW)	i	P_i (MW)
1	628.3185	1	628.3185
2	149.5997	2	299.1993
3	222.7491	3	299.1993
4	109.8666	4	159.7331
5	109.8666	5	159.7331
6	109.8666	6	159.7331
7	109.8666	7	159.7331
8	60.0000	8	159.7331
9	109.8666	9	159.7331
10	40.0000	10	77.3999
11	40.0000	11	77.3999
12	55.0000	12	92.3999
13	55.0000	13	87.6845

been concerned in aim to compare the performance and give the final conclusion on the proposed method. There have been six main cases with sixteen subcases. The evaluations have been made at the end of each study case. Clearly, the proposed ICSA approach has yielded more effective optimal solutions with faster execution time than almost all methods. Consequently, it can be concluded that the proposed method is much more superior to CCSA and is a very promising method for solving OLD problem.

Appendix

See Tables 16–26.

Nomenclature

$\delta_i, \lambda_i, \alpha_i, \beta_i, \gamma_i$: Fuel cost function coefficients of the ith thermal generation unit

B_{00}, B_{0j}, B_{ji}: Power loss matrix coefficients

FF_x, FF_{best}: The values of fitness of solution x and the so-far most effective solution among the current set of solutions

m_i: Number of fuels burnt in the ith thermal generation unit

Table 22: Optimal solution for Subcase 3.4.

i	P_i (MW)	i	P_i (MW)	i	P_i (MW)	i	P_i (MW)
1	110.79981	11	94.00001	21	523.27939	31	190.0000
2	110.79978	12	94.00001	22	523.27938	32	190.0000
3	97.39992	13	214.75979	23	523.27942	33	190.0000
4	179.73308	14	394.27940	24	523.27941	34	164.7998
5	87.79992	15	394.27940	25	523.27937	35	194.3978
6	140.00001	16	394.27940	26	523.27942	36	200.0000
7	259.59969	17	489.27940	27	10.0000	37	110.0000
8	284.59969	18	489.27940	28	10.0000	38	110.0000
9	284.59969	19	511.27941	29	10.0000	39	110.0000
10	130.0000	20	511.27938	30	87.7999	40	511.27939

Table 23: Optimal solution for Subcase 3.5.

i	P_i (MW)	i	P_i (MW)	i	P_i (MW)	i	P_i (MW)
1	112.1507	21	523.2804	41	110.8005	61	523.2795
2	110.7998	22	523.2796	42	110.7998	62	523.2811
3	97.39771	23	523.28	43	97.39996	63	523.2903
4	179.7288	24	523.2891	44	179.7331	64	523.2794
5	87.80075	25	523.2793	45	87.79894	65	523.2794
6	140	26	523.27	46	140	66	523.2794
7	259.5996	27	10	47	259.6004	67	10
8	284.5997	28	10	48	284.5997	68	10
9	284.5981	29	10	49	284.598	69	10
10	130	30	87.79985	50	130	70	87.79985
11	94.00001	31	190	51	94.00001	71	190
12	94.00001	32	190	52	94.00001	72	190
13	214.7584	33	190	53	214.7593	73	190
14	394.2826	34	164.7999	54	394.2643	74	164.8281
15	394.2789	35	187.4316	55	394.2841	75	200
16	394.2791	36	200	56	394.2793	76	200
17	489.2798	37	110	57	489.292	77	110
18	489.2794	38	110	58	489.2805	78	110
19	511.2794	39	110	59	511.2752	79	110
20	511.2794	40	511.2696	60	511.2784	80	511.2681

Table 24: Optimal solution for Subcase 4.2.

i	P_i (MW)	i	P_i (MW)	i	P_i (MW)	i	P_i (MW)	i	P_i (MW)	i	P_i (MW)
1	454.9445	16	454.9003	31	454.9473	46	455	61	455.0000	76	454.9652
2	454.9987	17	454.945	32	455.0000	47	454.9701	62	455.0000	77	454.9984
3	130.0000	18	129.9896	33	129.9783	48	130	63	129.9938	78	129.9939
4	130.0000	19	130	34	130.0000	49	129.9796	64	130.0000	79	129.9774
5	335.495	20	312.884	35	322.8285	50	321.6314	65	287.0957	80	321.9409
6	459.4493	21	459.9967	36	459.9993	51	459.8946	66	459.9971	81	460.0000
7	464.9997	22	464.977	37	464.9712	52	465	67	464.8247	82	464.8564
8	60.0058	23	60.0018	38	60.0394	53	60.0000	68	60.0173	83	60.0066
9	25.0027	24	25.0001	39	25.0000	54	25.0188	69	25.0000	84	25.0042
10	20.0000	25	20.0027	40	20.0193	55	20.0036	70	20.0378	85	20.2219
11	20.0001	26	20.0038	41	20.0122	56	20.0357	71	20.0002	86	20.0000
12	58.4585	27	55.1457	42	59.3489	57	58.9256	72	59.4547	87	57.7331
13	25.001	28	25	43	25.0082	58	25.0006	73	25.0094	88	25.0032
14	15.0000	29	15.0333	44	15.0016	59	15.0000	74	15.0000	89	15.0078
15	15.0007	30	15	45	15.0000	60	15.0033	75	15.0038	90	15.0031

Table 25: Optimal solution for Case 5.

i	P_i (MW)	i	P_i (MW)	i	P_i (MW)
1	454.95546	6	460	11	80
2	380	7	430	12	80
3	130	8	85.94226	13	25.0831
4	130	9	43.29983	14	15.6197
5	170	10	160	15	15.7882

Table 26: Optimal solution for Subcase 6.3.

i	P_i (MW)	i	P_i (MW)	i	P_i (MW)	i	P_i (MW)	i	P_i (MW)	i	P_i (MW)
1	219.1364	56	239.93	111	219.1364	166	239.9298	221	219.1364	276	239.9295
2	212.4025	57	287.7776	112	212.4025	167	287.7776	222	212.4025	277	287.7776
3	280.6577	58	238.8802	113	280.6577	168	238.8802	223	280.6577	278	238.8802
4	239.2836	59	426.0203	114	239.2836	169	426.0203	224	239.2836	279	426.0203
5	279.9217	60	275.8712	115	279.9217	170	275.8712	225	279.9217	280	275.8712
6	239.93	61	219.1364	116	239.93	171	219.1364	226	239.93	281	219.1364
7	287.7776	62	212.4025	117	287.7776	172	212.4025	227	287.7776	282	212.4025
8	238.8802	63	280.6577	118	238.8802	173	280.6579	228	238.8802	283	280.6577
9	426.0203	64	239.2836	119	426.0203	174	239.2836	229	426.0203	284	239.2836
10	275.8712	65	279.9217	120	275.8712	175	279.9217	230	275.8712	285	279.9217
11	217.5651	66	239.93	121	219.1364	76	239.93	231	219.1364	286	239.93
12	212.4022	67	287.7776	122	212.4025	77	287.7776	232	212.4025	287	287.7776
13	282.6687	68	238.8802	123	280.6577	78	238.8802	233	280.6577	288	238.8802
14	240.0905	69	426.0203	124	239.2836	79	426.0203	234	239.2836	289	426.0203
15	280.165	70	275.8712	125	279.9217	80	275.8712	235	279.9217	290	275.8712
16	240.4661	71	219.1364	126	239.93	181	219.1364	236	239.93	291	219.1364
17	287.5061	72	212.4025	127	287.7776	182	212.4025	237	287.7776	292	212.4025
18	239.6852	73	280.6577	128	238.8802	183	280.6577	238	238.8802	293	280.6577
19	426.1049	74	239.2836	129	426.0203	184	239.2836	239	426.0203	294	239.2836
20	277.0318	75	279.9217	130	275.8712	185	279.9217	240	275.8712	295	279.9217
21	219.1364	76	239.9301	131	219.1364	186	239.9298	241	219.1364	296	239.9305
22	212.4025	77	287.7776	132	212.4025	187	287.7776	242	212.4025	297	287.7776
23	280.6577	78	238.8802	133	280.6577	188	238.8802	243	280.6577	298	238.8802
24	239.2836	79	426.0203	134	239.2836	189	426.0203	244	239.2836	299	426.0203
25	279.9217	80	275.8712	135	279.9217	190	275.8712	245	279.9217	300	275.8712
26	239.93	81	219.1363	136	239.9309	191	219.1364	246	239.93	301	219.1364
27	287.7776	82	212.4025	137	287.7776	192	212.4025	247	287.7776	302	212.4025
28	238.8802	83	280.6577	138	238.8802	193	280.6577	248	238.8802	303	280.6577
29	426.0203	84	239.2836	139	426.0203	194	239.2836	249	426.0203	304	239.2836
30	275.8712	85	279.9217	140	275.8712	195	279.9217	250	275.8712	305	279.9217
31	219.1365	86	239.93	141	219.1364	196	239.93	251	219.1364	306	239.93
32	212.4025	87	287.7776	142	212.4025	197	287.7776	252	212.4025	307	287.7776
33	280.6577	88	238.8802	143	280.6577	198	238.8802	253	280.6577	308	238.8802
34	239.2836	89	426.0203	144	239.2836	199	426.0203	254	239.2836	309	426.0203
35	279.9217	90	275.8712	145	279.9217	200	275.8712	255	279.9217	310	275.8714
36	239.93	91	219.1344	146	239.93	201	219.1364	256	239.93	311	219.1369
37	287.7776	92	212.4025	147	287.7776	202	212.4025	257	287.7776	312	212.4025
38	238.8802	93	280.6577	148	238.8802	203	280.6577	258	238.8802	313	280.6577
39	426.0203	94	239.2836	149	426.0193	204	239.2836	259	426.0203	314	239.2836
40	275.8712	95	279.9217	150	275.8712	205	279.9217	260	275.8712	315	279.9217
41	219.1364	96	239.93	151	219.1364	206	239.93	261	219.1364	316	239.93
42	212.4025	97	287.7776	152	212.4025	207	287.7776	262	212.4025	317	287.7776
43	280.6577	98	238.8802	153	280.6577	208	238.8802	263	280.6577	318	238.8802

Table 26: Continued.

i	P_i (MW)	i	P_i (MW)	i	P_i (MW)	i	P_i (MW)	i	P_i (MW)	i	P_i (MW)
44	239.2836	99	426.0203	154	239.2832	209	426.0203	264	239.2836	319	426.0203
45	279.9217	100	275.8712	155	279.9217	210	275.8712	265	279.9217	320	275.8710
46	239.93	101	219.1384	156	239.93	211	219.1359	266	239.93		
47	287.7776	102	212.4025	157	287.7776	212	212.4025	267	287.7776		
48	238.8798	103	280.6577	158	238.8806	213	280.6577	268	238.8802		
49	426.0203	104	239.2840	159	426.0213	214	239.2836	269	426.0203		
50	275.8712	105	279.9217	160	275.8712	215	279.9217	270	275.8712		
51	219.1364	106	239.93	161	219.1364	216	239.93	271	219.1364		
52	212.4025	107	287.7776	162	212.4025	217	287.7776	272	212.4025		
53	280.6577	108	238.8802	163	280.6577	218	238.8802	273	280.6577		
54	239.2836	109	426.0203	164	239.2836	219	426.0203	274	239.2836		
55	279.9217	110	275.8712	165	279.9217	220	275.8712	275	279.9217		

N: Number of all available thermal generation units

n_i: Number of prohibited power zones of the ith thermal generation unit

P_{ik}^u, P_{ik}^l: Upper and lower limits of the ith thermal generation unit corresponding to the kth PPZ

P_a: Probability of replacing control variables in each old solution

P_D: Real power demand of all loads in system

$P_{i,max}$, $P_{i,min}$: The highest and lowest real power outputs of the ith thermal generation unit

$P_{ij,max}$, $P_{ij,min}$: The highest and lowest real power outputs of the ith thermal generation unit corresponding to the jth fuel type

$S_{i,max}$: Maximum real power reserve contribution of the thermal generation unit i

Sol_{rand1}, Sol_{rand2}, Sol_{rand3}, Sol_{rand4}: Randomly mixed solutions from the set of current solutions

Sol_x, Sol_{Gbest}: The old solution x and the most effective solution

S_R: Real power reserve requirement of system

δ_{ij}, λ_{ij}, α_{ij}, β_{ij}, γ_{ij}: Fuel cost function coefficients of the ith thermal generation unit corresponding to the jth fuel type

ε_1, ε_2, ε_3, ε_4, ε_5, ε_6: Random numbers between 0 and 1

Ω: Number of generation units considering PPZ constraint.

Data Availability

No data were used to support this study.

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] N. Sinha, R. Chakrabarti, and P. K. Chattopadhyay, “Evolutionary programming techniques for economic load dispatch,” *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 1, pp. 83–94, 2003.

[2] P. Hansen, “A separable approximation dynamic programming algorithm for economic dispatch with transmission losses,” *Yugoslav Journal of Operations Research*, vol. 12, no. 2, 2002.

[3] N. T. Thang, “Solving economic dispatch problem with piecewise quadratic cost functions using lagrange multiplier theory,” in *Proceedings of the 3rd International Conference on Computer Technology and Development, (ICCTD ’11)*, pp. 359–364, ASME Press, 2011.

[4] S. K. Mishra and S. K. Mishra, “A comparative study of solution of economic load dispatch problem in power systems in the environmental perspective,” *Procedia Computer Science*, vol. 48, pp. 96–100, 2015.

[5] A. A. Al-Subhi and H. K. Alfares, “Economic load dispatch using linear programming: a comparative study,” *International Journal of Applied Industrial Engineering*, vol. 3, no. 1, pp. 16–36, 2016.

[6] C. Zhou, G. Huang, and J. Chen, "Planning of electric power systems considering virtual power plants with dispatchable loads included: an inexact two-stage stochastic linear programming model," *Mathematical Problems in Engineering*, vol. 2018, Article ID 7049329, 12 pages, 2018.

[7] T. T. Nguyen, N. V. Quynh, and L. Van Dai, "Improved firefly algorithm: a novel method for optimal operation of thermal generating units," *Complexity*, vol. 2018, Article ID 7267593, 23 pages, 2018.

[8] S. S. Haroon, S. Hassan, S. Amin et al., "Multiple fuel machines power economic dispatch using stud differential evolution," *Energies*, vol. 11, no. 6, pp. 1–20, 2018.

[9] C.-T. Su and C.-L. Chiang, "Nonconvex power economic dispatch by improved genetic algorithm with multiplier updating method," *Electric Power Components and Systems*, vol. 32, no. 3, pp. 257–273, 2004.

[10] N. Noman and H. Iba, "Differential evolution for economic load dispatch problems," *Electric Power Systems Research*, vol. 78, no. 8, pp. 1322–1331, 2008.

[11] P. Somasundaram and K. Kuppusamy, "Application of evolutionary programming to security constrained economic dispatch," *International Journal of Electrical Power & Energy Systems*, vol. 27, no. 5-6, pp. 343–351, 2005.

[12] T. Jayabarathi, K. Jayaprakash, and D. Jeyakumar, "Evolutionary programming technique for different kinds of economic dispatch problems," *Electric Power Systems Research*, vol. 73, no. 2, pp. 169–176, 2005.

[13] P. Subbaraj, R. Rengaraj, and S. Salivahanan, "Enhancement of combined heat and power economic dispatch using self adaptive real-coded genetic algorithm," *Applied Energy*, vol. 86, no. 6, pp. 915–921, 2009.

[14] C.-L. Chiang and C.-T. Su, "Adaptive-improved genetic algorithm for the economic dispatch of units with multiple fuel options," *Cybernetics and Systems*, vol. 36, no. 7, pp. 687–704, 2005.

[15] C. Chiang, "Improved genetic algorithm for power economic dispatch of units with valve-point effects and multiple fuels," *IEEE Transactions on Power Systems*, vol. 20, no. 4, pp. 1690–1699, 2005.

[16] K. Meng, H. G. Wang, Z. Y. Dong, and K. P. Wong, "Quantum-inspired particle swarm optimization for valve-point economic load dispatch," *IEEE Transactions on Power Systems*, vol. 25, no. 1, pp. 215–222, 2010.

[17] P. Erdogmus, A. Ozturk, and S. Duman, "Environmental/economic dispatch using genetic algorithm and simulated annealing," in *Proceedings of the International Conference on Electric Power and Energy Conversion Systems, EPECS '09*, pp. 1–4, IEEE, 2009.

[18] L. D. S. Coelho and V. C. Mariani, "Improved differential evolution algorithms for handling economic dispatch optimization with generator constraints," *Energy Conversion and Management*, vol. 48, no. 5, pp. 1631–1639, 2007.

[19] H. Dakuo, W. Fuli, and M. Zhizhong, "A hybrid genetic algorithm approach based on differential evolution for economic dispatch with valve-point effects," *Electrical Power and Energy Systems*, vol. 30, pp. 31–38, 2008.

[20] C.-F. Chang, J.-J. Wong, J.-P. Chiou, and C.-T. Su, "Robust searching hybrid differential evolution method for optimal reactive power planning in large-scale distribution systems," *Electric Power Systems Research*, vol. 77, no. 5-6, pp. 430–437, 2007.

[21] M. R. Farooqi, P. Jain, and K. R. Niazi, "Using Hopfield neural network for economic dispatch of power systems," in *Proceedings of the National Power Engineering Conference, PECon '03*, pp. 5–10, Bangi, Malaysia, 2003.

[22] T. Yalcinoz, H. Altun, and U. Hasan, "Constrained economic dispatch with prohibited operating zones: a Hopfield neural network approach," in *Proceedings of the 10th Mediterranean Electrotechnical Conference. Information Technology and Electrotechnology for the Mediterranean Countries. Proceedings. MeleCon '00 (Cat. No. 00CH37099)*, pp. 570–573, Lemesos, Cyprus, 2000.

[23] D. N. Vo, P. Schegner, and W. Ongsakul, "Cuckoo search algorithm for non-convex economic dispatch," *IET Generation, Transmission & Distribution*, vol. 7, no. 6, pp. 645–654, 2013.

[24] J.-B. Park, K.-S. Lee, J.-R. Shin, and K. Y. Lee, "A particle swarm optimization for economic dispatch with non-smooth cost functions," *IEEE Transactions on Power Systems*, vol. 20, no. 1, pp. 34–42, 2005.

[25] T. Niknam, H. D. Mojarrad, and M. Nayeripour, "A new fuzzy adaptive particle swarm optimization for non-smooth economic dispatch," *Energy*, vol. 35, no. 4, pp. 1764–1778, 2010.

[26] L. D. S. Coelho and V. C. Mariani, "An improved harmony search algorithm for power economic load dispatch," *Energy Conversion and Management*, vol. 50, no. 10, pp. 2522–2526, 2009.

[27] M. Mahdavi, M. Fesanghary, and E. Damangir, "An improved harmony search algorithm for solving optimization problems," *Applied Mathematics and Computation*, vol. 188, no. 2, pp. 1567–1579, 2007.

[28] A. Bhattacharya and P. K. Chattopadhyay, "Solving complex economic load dispatch problem using biogeography-based optimization," *Expert Systems with Applications*, vol. 37, no. 5, pp. 3605–3615, 2010.

[29] D. Simon, "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008.

[30] J. S. Alsumait, J. K. Sykulski, and A. K. Al-Othman, "A hybrid GA-PS-SQP method to solve power system valve-point economic dispatch problems," *Applied Energy*, vol. 87, no. 5, pp. 1773–1781, 2010.

[31] M. S. Turgut and G. K. Demir, "Quadratic approximation–based hybrid Artificial Cooperative Search algorithm for economic emission load dispatch problems," *International Transactions on Electrical Energy Systems*, vol. 27, no. 4, pp. 1–14, 2017.

[32] T. A. A. Victoire and A. E. Jeyakumar, "Hybrid PSO-SQP for economic dispatch with valve-point effect," *Electric Power Systems Research*, vol. 71, no. 1, pp. 51–59, 2004.

[33] D.-K. He, F.-L. Wang, and Z.-Z. Mao, "Hybrid genetic algorithm for economic dispatch with valve-point effect," *Electric Power Systems Research*, vol. 78, no. 4, pp. 626–633, 2008.

[34] D. C. Secui, "A modified symbiotic organisms search algorithm for large scale economic dispatch problem with valve-point loading effects," *Energy*, vol. 113, pp. 366–384, 2016.

[35] K. Bhattacharjee, A. Bhattacharya, and S. H. N. Dey, "Teaching-learning-based optimization for different economic dispatch problems," *Scientia Iranica*, vol. 21, no. 3, pp. 870–884, 2014.

[36] K. Bhattacharjee, A. Bhattacharya, and S. H. N. Dey, "Oppositional Real Coded Chemical Reaction Optimization for different economic dispatch problems," *International Journal of Electrical Power & Energy Systems*, vol. 55, pp. 378–391, 2014.

[37] V. Hosseinnezhad, M. Rafiee, M. Ahmadian, and M. T. Ameli, "Species-based Quantum Particle Swarm Optimization for economic load dispatch," *International Journal of Electrical Power & Energy Systems*, vol. 63, pp. 311–322, 2014.

[38] M. S. P. Subathra, S. E. Selvan, T. A. A. Victoire, A. H. Christinal, and U. Amato, "A hybrid with cross-entropy method and sequential quadratic programming to solve economic load dispatch problem," *IEEE Systems Journal*, vol. 9, no. 3, pp. 1031–1044, 2015.

[39] J. Zhan, Q. H. Wu, C. Guo, and X. Zhou, "Economic dispatch with non-smooth objectives, part I: local minimum analysis," *IEEE Transactions on Power Systems*, vol. 30, no. 2, pp. 710–721, 2015.

[40] A. K. Barisal and R. C. Prusty, "Large scale economic dispatch of power systems using oppositional invasive weed optimization," *Applied Soft Computing*, vol. 29, pp. 122–137, 2015.

[41] M. Basu, "Improved differential evolution for economic dispatch," *International Journal of Electrical Power & Energy Systems*, vol. 63, pp. 855–861, 2014.

[42] V. S. Aragón, S. C. Esquivel, and C. C. Coello, "An immune algorithm with power redistribution for solving economic dispatch problems," *Information Sciences*, vol. 295, pp. 609–632, 2015.

[43] M. Ghasemi, M. Taghizadeh, S. Ghavidel, and A. Abbasian, "Colonial competitive differential evolution: an experimental study for optimal economic load dispatch," *Applied Soft Computing*, vol. 40, pp. 342–363, 2016.

[44] B. R. Adarsh, T. Raghunathan, T. Jayabarathi, and X.-S. Yang, "Economic dispatch using chaotic bat algorithm," *Energy*, vol. 96, pp. 666–675, 2016.

[45] N. Ghorbani and E. Babaei, "Exchange market algorithm for economic load dispatch," *International Journal of Electrical Power & Energy Systems*, vol. 75, pp. 19–27, 2016.

[46] J. X. V. Neto, G. Reynoso-Meza, T. H. Ruppel, V. C. Mariani, and L. D. S. Coelho, "Solving non-smooth economic dispatch by a new combination of continuous GRASP algorithm and differential evolution," *International Journal of Electrical Power & Energy Systems*, vol. 84, pp. 13–24, 2017.

[47] A. Kavousi-Fard and A. Khosravi, "An intelligent θ-Modified Bat Algorithm to solve the non-convex economic dispatch problem considering practical constraints," *Electrical Power and Energy Systems*, vol. 82, pp. 189–196, 2016.

[48] M. A. Al-Betar, M. A. Awadallah, A. T. Khader, and A. L. Bolaji, "Tournament-based harmony search algorithm for non-convex economic load dispatch problem," *Applied Soft Computing*, vol. 47, pp. 449–459, 2016.

[49] M. A. Al-Betar, M. A. Awadallah, I. A. Doush, E. Alsukhni, and H. ALkhraisat, "A non-convex economic dispatch problem with valve loading effect using a new modified β-hill climbing local search algorithm," *Arabian Journal for Science and Engineering*, pp. 1–8, 2018.

[50] M. A. Awadallah, M. A. Al-Betar, A. L. Bolaji, E. M. Alsukhni, and H. Al-Zoubi, "Natural selection methods for artificial bee colony with new versions of onlooker bee," *Soft Computing*, pp. 1–40, 2018.

[51] S. H. Kaboli and A. K. Alqallaf, "Solving non-convex economic load dispatch problem via artificial cooperative search algorithm," *Expert Systems with Applications*, vol. 128, pp. 14–27, 2019.

[52] D. Singh and J. Dhillon, "Ameliorated grey wolf optimization for economic load dispatch problem," *Energy*, vol. 169, pp. 398–419, 2019.

[53] X. S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proceedings of the World Congress on Nature and Biologically Inspired Computing (NaBIC '09)*, pp. 210–214, India, 2009.

[54] S. Sahoo, K. M. Dash, R. C. Prusty, and A. K. Barisal, "Comparative analysis of optimal load dispatch through evolutionary algorithms," *Ain Shams Engineering Journal*, vol. 6, pp. 107–120, 2015.

[55] M. Basu and A. Chowdhury, "Cuckoo search algorithm for economic dispatch," *Energy*, vol. 60, pp. 99–108, 2013.

[56] A. B. Serapião, "Cuckoo search for solving economic dispatch load problem," *Intelligent Control and Automation*, vol. 04, no. 04, pp. 385–390, 2013.

[57] A. H. Bindu and M. D. Reddy, "Economic load dispatch using cuckoo search algorithm," *International Journal of Engineering Research and Applications*, vol. 3, pp. 498–502, 2013.

[58] E. Afzalan and M. Joorabian, "An improved cuckoo search algorithm for power economic load dispatch," *International Transactions on Electrical Energy Systems*, vol. 25, no. 6, pp. 958–975, 2015.

[59] T. T. Nguyen and D. N. Vo, "The application of one rank cuckoo search algorithm for solving economic load dispatch problems," *Applied Soft Computing*, vol. 37, pp. 763–773, 2015.

[60] S. M. Islam, S. Das, S. Ghosh, S. Roy, and P. N. Suganthan, "An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 2, pp. 482–499, 2012.

[61] Z. L. Gaing, "Particle swarm optimization to solving the economic dispatch considering the generator constraints," *IEEE Transactions on Power Systems*, vol. 18, no. 3, pp. 1187–1195, 2003.

[62] X.-S. Yang and S. Deb, "Engineering optimisation by Cuckoo search," *International Journal of Mathematical Modelling and Numerical Optimisation*, vol. 330, pp. 43–51, 2010.