Minimizing Electricity Fuel Cost of Thermal Generating Units by Using Improved Firefly Algorithm

Thang Trung Nguyen[1*], Bao Quoc Nguyen[2], Phuong Duy Nguyen[3] & Chiem Trong Hien[4]

[1]Power System Optimization Research Group, Faculty of Electrical and Electronics Engineering, Ton Duc Thang University, 19 Nguyen Huu Tho Street, Ho Chi Minh City 700000, Vietnam
[2]Faculty of Electrical and Electronics Engineering, Ton Duc Thang University, 19 Nguyen Huu Tho street, Ho Chi Minh City 700000, Vietnam
[3]Faculty of Electronics - Telecommunications, Saigon University, 273 An Duong Vuong street, Ho Chi Minh city 700000, Vietnam
[4]Faculty of Electrical and Electronic Technology, Ho Chi Minh City University of Food Industry, 140 Le Trong Tan street, Ho Chi Minh City 700000, Vietnam
*E-mail: nguyentrungthang@tdtu.edu.vn

Abstract. This paper presents the application of an improved firefly algorithm (IFA) for minimizing total electricity generation fuel cost while all loads are supplied by thermal generating units. The proposed IFA was developed by combining two proposed improvements of the firefly algorithm (FA), i.e. improvement of the distance between two considered solutions and improvement of the new-solution production technique. The effect of each proposed improvement on the conventional firefly algorithm (FA) and the performance of IFA were investigated in two study cases, i.e. single- and multi-fuel option based thermal generating units. In the first case, three different systems with three, six and twenty units were employed, while a ten-unit system with four different loads was tested in the second case. The comparison results between IFA and existing methods, including three other FA variants, revealed that the two proposed improvements of FA are very efficient and make IFA a very promising meta-heuristic algorithm for minimizing fuel cost of thermal generating units.

Keywords: *improved firefly algorithm; multi-fuel; single-fuel; thermal generating units; total fuel.*

1 Introduction

The world is currently experiencing rapid population growth, while many countries are confronted with high rates of urbanization. Thus, the question how to meet the increasing demands of essential products, energy and services – the main challenge of this century – needs to be considered. To solve this matter, a large power source is required to supply services and daily energy consumption. Hence, the electrical power market will become more competitive and more complicated than ever before. The solution is to distribute the power system

Received April 28th, 2018, Revised January 6th, 2019, Accepted for publication January 14th, 2019.
 ISSN: 2337-5779, DOI: 10.5614/j.eng.technol.sci.2019.51.1.9

load to generation units so that the lowest fuel cost function is accomplished while satisfying the system constraints, an approach that is known as optimal operation of thermal generating units (ELD) [1]. In the system operation conditions of the ELD problem, the fuel resources of the thermal units can be supplied according to two cases. The first case is single-fuel, where the fuel cost function of each generator can be represented approximately by a single quadratic function [2]. The second case is multi-fuel (coal, natural gas and oil), where the generator can be represented by a segmented-piecewise quadratic function [3-11]. Traditionally, a wide range of deterministic methods have been used to solve the ELD problem, namely the Lagrangian relaxation algorithm [12], the gradient method [13], the lambda iteration method [14] and the Hopfield model (HNN) [1,3,15-17]. These methods share the same advantages, such as requiring only a short execution time, having a small number of control parameters and providing a single optimal solution. However, there are some drawbacks when handling the problems related to complex multi-fuel constraints, large power systems and a non-differentiable objective.

During the previous decades several approaches have been adopted to deal with the ELD problem, such as Tabu Search (TS) [18], differential evolution (DE) [19], Non-dominated Sorting Genetic Algorithm II (NSGA-II) [20], biogeography-based optimization (BBO) [21-22], the Fuzzy Logic Controlled Genetic Algorithm (FCGA) [23], and the Cuckoo Search Algorithm (CSA) [24-25]. Among these, DE is one of the most popular methods and has been widely and successfully applied. DE can handle difficult problems with nonlinear constraints and complicated objective functions. In addition, it has a small number of control parameters that lie within a predetermined range. However, the task of finding the best values for these control parameters by tuning is time-consuming and needs a large number of evaluations for different results from different sets of control parameters [26]. In fact, DE has two main factors, the crossover factor and the mutation factor, where the first is from zero to 2 while the latter is from zero to 1. In addition, the new-solution generation method is based on the same formula, which has high probability of converging to a local optimum solution with low quality.

BBO has better characteristics than DE since it uses two generations per iteration but only one evaluation time. Thus, BBO can overcome the shortcoming of easily converging to a local optimum but it has difficulty in coping with the selection of the control parameters. BBO has more control parameters, i.e. population size, iterations, maximum immigration rate, mutation coefficient, maximum emigration rate, retaining rate and habitat modification probability. CSA can overcome the limitations of these two methods. CSA can avoid falling into local optimum zones and finding lower

quality solutions by using two mechanisms: exploration via Levy flights and exploitation via mutation. The Levy flights mechanism can explore large search zones while the mutation operation focuses on narrow zones. Furthermore, CSA has a small number of adjustment parameters, i.e. population size, iterations and probability of mutation performance. The first two are popular parameters that all metaheuristic algorithms also have, while the third one is very a simple one for tuning within the range from 0 to 1. The firefly algorithm is a population based meta-heuristic algorithm similar to PSO, DE, CSA, etc. It was built by Yang in 2008 for solving optimization problems [27]. The configuration of FA consists of three procedures for updating the distance between two considered fireflies, updating the step size and updating the solutions.

In this paper, we propose two modifications of FA in order to tackle several of its disadvantages, such as premature convergence to a local optimum solution and impossibility of jumping out of a search zone with many local optimum solutions. In the first modification, we propose a new formula to update the radius between a considered firefly X_i (one solution) and another firefly X_j (another solution) with a lower fitness function than the considered solution. The proposed radius based on X_i and the best solution X_{Gbest} is more effective than that based on X_i and X_j in FA. In the second modification, we propose a new algorithm for producing new solutions of an old solution by suggesting two models for the updated step size. A larger or smaller updated step size will be used to find solutions in different zones and to avoid converging to a local optimum and getting trapped into a search zone with many local optimums. As a result, the new algorithm provides a very considerable improvement compared to FA. The application of each modification was evaluated by testing on four systems with nine cases, i.e. nine thermal generating units using single-fuel and multi-fuel ELD.

2 Problem Formulation

2.1 Objective Function

In single-fuel ELD, the fuel cost of each generating unit is expressed as a quadratic function of its power output. The objective of the problem is to minimize the total fuel cost of N available units, as shown in Eq. (1):

$$\text{Min } F = \sum_{s=1}^{N} F_s(P_s), \tag{1}$$

where P_s is the real power output of generator s and F_s is the fuel cost function of thermal unit s, which can be represented in Eqs. (2) and (3) corresponding to single-fuel and multi-fuel cases.

$$F_s(P_s) = a_s + b_s P_s + c_s P_s^2 \ \ (\$ / \text{h}), \tag{2}$$

$$F_s(P_s) = \begin{cases} a_{s1} + b_{s1} P_s + c_{s1} P_s^2, \text{ fuel 1}, P_{s,\min} \le P_s \le P_{s1,max} \\ ..... \\ a_{sm} + b_{sm} P_s + c_{sm} P_s^2, \text{ fuel m}, P_{sm,\min} \le P_s \le P_{sm,max} \\ ..... \\ a_{sMs} + b_{sMs} P_s + c_{sMs} P_s^2, \text{ fuel } Ms, P_{sMs,\min} \le P_s \le P_{s,\max} \end{cases}, \tag{3}$$

where a_s, b_s, and c_s are fuel cost coefficients of unit s with single-fuel option; a_{sm}, b_{sm}, c_{sm} denote fuel cost coefficients for fuel type m of unit s; $P_{sm,min}$ and $P_{sm,max}$ denote the lower and upper limits for fuel m of unit s, respectively; $P_{s,min}$ and $P_{s,max}$ represent the lowest value and the highest generations that thermal unit s can produce; M_s represents the number of fuel options of thermal unit s.

2.2 Set of Constraints

Active power balance: power from the generating units together with electricity load P_{LD} and power losses P_{TL} should satisfy the constraint of Eq. (4):

$$\sum_{s=1}^{N} P_s = P_{LD} + P_{TL}, \tag{4}$$

where P_{TL} is found by using Eq. (5) [1]:

$$P_{TL} = \sum_{i=1}^{N} \sum_{j=1}^{N} P_i B_{ij} P_j + \sum_{i=1}^{N} B_{0i} P_i + B_{00}, \tag{5}$$

where B_{ij}, B_{0i}, B_{00} are terms in the transmission power loss coefficient matrix.

Limitations of the thermal generating units: the power output of each thermal generating unit must follow the rule in Eq. (6):

$$P_{s,\min} \le P_s \le P_{s,\max}. \tag{6}$$

3 Proposed Improved Firefly Algorithm

3.1 Firefly Algorithm

Each firefly i is represented by a position X_i corresponding to solution X_i at the current iteration. When the fitness function of solution i is higher than that of another solution j, the distance between firefly i and j is obtained by using Eq. (7):

$$r_{ij} = \sqrt{(X_i - X_j)^2}. \tag{7}$$

Then the updated new distance solutions are carried out using Eqs. (8) and (9):

$$\beta = \beta_0 e^{-\gamma r_{ij}^2}, \tag{8}$$

$$\Delta X_{ijnew} = X_i + \beta.\Delta X_{ij} + rand_i, \tag{9}$$

where $rand_i$ is a random solution i, β_0 is the attractiveness at zero distance (normally set to 1). X_j is a solution with a lower fitness function than X_i; and ΔX_{ij} is the updated step size calculated by employing Eq. (10).

$$\Delta X_{ij} = (X_{ij} - X_i) \tag{10}$$

The whole description of FA is shown in detail in the flowchart in Figure 1.

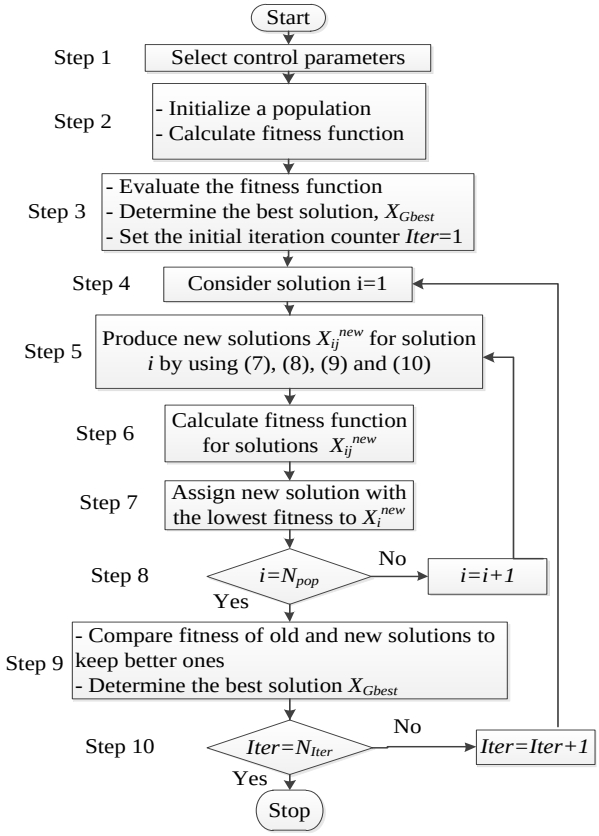

Figure 1 Flowchart of implementing FA for a general optimization problem.

3.2 Proposed Improved Firefly Algorithm

In the paper, we propose two improvements regarding the considered radius and the updated step size. Instead of using the distance between the considered solution i and another better solution to determine the radius, the best solution X_{Gbest} is recommended to be used for calculating the radius:

$$r_{iBest} = \sqrt{(X_i - X_{Gbest})^2}. \tag{11}$$

where X_{Gbest} is the best solution in the population.

In the second improvement, a novel technique is proposed for producing new solutions with higher quality than those of FA. It is clear that the manner of producing the updated step size by using Eq. (9) is similar to the mutation operation of the differential evolution algorithm (DEA) in which β acts as mutation factor, ranging from 0 to 2. Some previous studies [26] have pointed out disadvantages of DEA, such as low convergence to a global optimum or easily getting trapped in a local optimum. Consequently, the proposed improvement aims to tackle the limitations of FA by using Eqs. (12)-(14):

$$\Delta X_{1ij} = (X_j - X_i + X_{r1} - X_{r2}), \tag{12}$$

$$\Delta X_{2ij} = \Delta X_{1ij} + (X_{Gbest} - X_{Worst}), \tag{13}$$

$$\Delta X_{ij} = \begin{cases} \Delta X_{1ij} & if \;\; RN_i > PT \\ \Delta X_{2ij} & otherwise \end{cases}. \tag{14}$$

The definitions in Eq. (14) are as follows: X_{r1} and X_{r2} are two random solutions among the current population that are different from X_i and X_j; X_{Gbest} and X_{Worst} are the best and the worst solutions among the current population; RN_i is a random number ranging from 0 to 1, generated for solution i; PT is predetermined tolerance, which was set to 0.5 for all cases in this paper to ensure that the probability is 50% for each model. The implementation of the proposed IFA for a general optimization problem is similar to the flowchart shown in Figure 1 of FA. The difference between the two considered algorithms is the way in which new solutions are produced.

4 Implementation of IFA for ELD Problem

4.1 Dealing with Load Demand- Supply Balance Constraint

In order to deal with the load demand-supply balance constraint, one thermal generating unit must be considered as dependent variable while the rest are decision variables, which are included in the position of each firefly in the initialization step and are updated in each iteration by using the search strategy

of IFA. Consequently, the position of firefly *i* will go from thermal generating unit 1 to unit N-1 as shown in Eq. (15):

$$X_i=[P_{1,i}, P_{2,i}, \ldots., P_{N-1,i}];\ i=1, \ldots, N_{pop}, \tag{15}$$

where X_i must always meet the constraint of Eqs. (16)-(18):

$$X_{\min} \le X_i \le X_{\max}, \tag{16}$$

$$X_{min}=[P_{1,min}, P_{2,min}, \ldots., P_{N-1,min}], \tag{17}$$

$$X_{max}=[P_{1,max}, P_{2,max}, \ldots., P_{N-1,max}]. \tag{18}$$

As a result, the load demand-supply balance constraint can be dealt with successfully by using the dependent variable $P_{N,i}$ obtained by Eq. (19) [25].

$$P_{N,i} = P_{LD} + P_{TL} - \sum_{s=1}^{N-1} P_{s,i}. \tag{19}$$

4.2 Penalizing Violations by $P_{N,I}$

Eq. (20) indicates that there is a possibility that $P_{N,i}$ violates its limitations, i.e. being lower than the lowest generation or higher than the highest generation. Therefore, the violation must be controlled and considered in the quality evaluation of the solutions. This is done by calculating the penalty term as indicated in Eq. (20):

$$Penalty_i = \begin{cases} P_{N,\mathrm{i}} - P_{N,\max} & if\ \ P_{N,\mathrm{i}} > P_{N,\max} \\ P_{N,\min} - P_{N,\mathrm{i}} & if\ \ P_{N,\mathrm{i}} < P_{N,\min} \\ 0 & if\ \ P_{N,\min} \le P_{N,\mathrm{i}} \le P_{N,\max} \end{cases}, \tag{20}$$

4.3 Fitness Function

The fitness function of all solutions should be determined to arrange the effectiveness of all the solutions. The fitness function, which considers the objective function and the penalty term, is shown in Eq. (21):

$$FT_i = \sum_{s=1}^{N} F_s(P_s) + PF \times (Penalty_i)^2, \tag{21}$$

where FT_i is the fitness function of solution *i* and PF is the penalty factor used to amplify the violation of the dependent variable.

5 Numerical Results

The proposed IFA, FA and two other improved versions corresponding to the first improvement (called IFA1) and the second improvement (called IFA2)

were tested in four cases, where the first three cases considered thermal generating units using only the single-fuel option while the last one took thermal generating units using the multi-fuel option into consideration. The details of the four test systems were as follows:

Case 1*:* Three thermal generating unit test systems with a load of 850 MW [18].
Case 2*:* Six thermal generating unit test systems with varying loads, i.e. 800 MW, 1200 MW and 1800 MW corresponding to cases 2.1, 2.2 and 2.3 [23].
Case 3*:* Twenty thermal generating unit test systems with a load of 2500 MW [15].
Case 4*:* Ten thermal generating units with varying loads, i.e. 2400 MW, 2500 MW, 2600 MW and 2700 MW corresponding to cases 4.1, 4.2, 4.3 and 4.4 [6].

In addition, the population size and the highest iteration number selected for implementation of IFA, FA, IFA1 and IFA2 were identical, as shown in Table 1. In all four cases, each method was run in fifty independent trials using Matlab and a computer with 4GB of RAM and a 2.4 Ghz processor.

Table 1 Selection of population size and highest iteration number.

Type of fuel	Case	N_{pop}	N_{Iter}
	1	10	15
Single-fuel	2	10	40
	3	20	500
Multi-fuel	4	15	200

5.1 Impact of Proposed Modifications on Obtained Results

In this section, the impact of each modificaiton on the performance of the proposed method is discussed as well as the advantages of the proposed method over FA. Thus, four FA variants were run in cases 1, 2 and 3. The results, consisting of minimum cost, average cost, maximum cost and standard deviation cost, are reported in Tables 2 and 3.

The comparison of best cost reflects the best optimal solution and the comparison of standard deviation cost reflects the stabilization of the search ability. The two comparison criteria are both essential to indicate the performance of each method. In case 1 with the 3-TGU system, the proposed method obtained lower best cost than FA, IFA1 and IFA2 by $0.034, $0.014 and $0.006 respectively. Similarly, the standard deviation cost of IFA was lower than that of FA, IFA1 and IFA2 by $55.3, $1.16, $0.085 respectively.

The comparison shows that the proposed method performed the best and FA the worst, while IFA2 was better than IFA1. The same outcome was obtained in the subcases of case 2 and case 3. Clearly, the first modification has only a slight

impact on the results of the proposed method, while the second modification has a significant impact. The best cost after fifty runs obtained by the four methods for cases 1 and 3 (shown in Figures 2 and 3) shows the superiority of IFA over FA by small fluctuations, high stablization and approximate convergence to the best solution. For the multi-fuel case, the result comparisons are shown in Table 4 and the fifty runs of case 4.1 are plotted in Figure 4. The minimum cost confirms the better performance of the proposed method over FA, while the standard deviation and the figure give evidence of a stable search in the proposed method. Consequently, it can be concluded that the proposed method is much more effective and robust than FA.

Table 2 Results ($/h) obtained by FA methods in case 1.

Method	Best cost	Mean cost	Worst cost	Std. dev.
FA	8344.627	8350.38	8378.291	55.30577
IFA1	8344.607	8344.71	8349.779	1.16356
IFA2	8344.599	8344.6	8344.72	0.08551
IFA	8344.593	8344.59	8344.593	0.00006

Table 3 Best Cost ($/h) obtained by FA methods in case 2 and case 3.

Method	Case 2.1	Case 2.2	Case 2.3	Case 3
FA	8243.2632	11482.6	16583.26	62514.98
IFA1	8230.7388	11480	16581.9	62460.49
IFA2	8227.5393	11477.3	16579.6	62458.88
IFA	8227.0986	11477.1	16579.33	62456.64

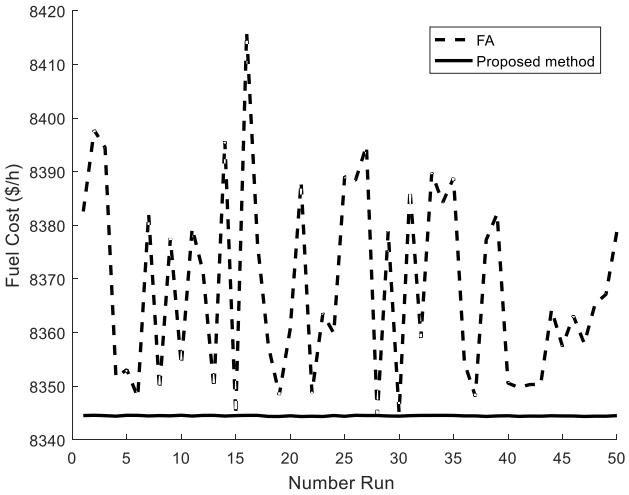

Figure 2 The best cost after fifty runs obtained by FA and the proposed method in case 1.

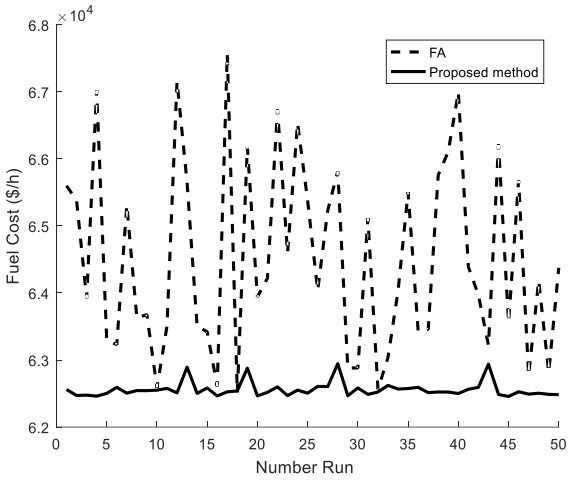

Figure 3 The best cost after fifty runs obtained by FA and the proposed method in case 3.

Table 4 Results ($/h) obtained by FA methods for case 4.

Method	Case 4.1		Case 4.2		Case 4.3		Case 4.4	
	Min.	**Std.**	**Min.**	**Std.**	**Min.**	**Std.**	**Min.**	**Std.**
FA	485.661	6.78	528.11	6.54	577.003	8.89	627.887	3.25
IFA1	482.821	5.82	528.091	5.27	575.41	5.17	626.73	3.07
IFA2	481.933	2.1	526.77	3.14	575.03	4.38	624.05	2.43
IFA	481.723	0.24	526.24	0.33	574.381	1.63	623.81	0.83

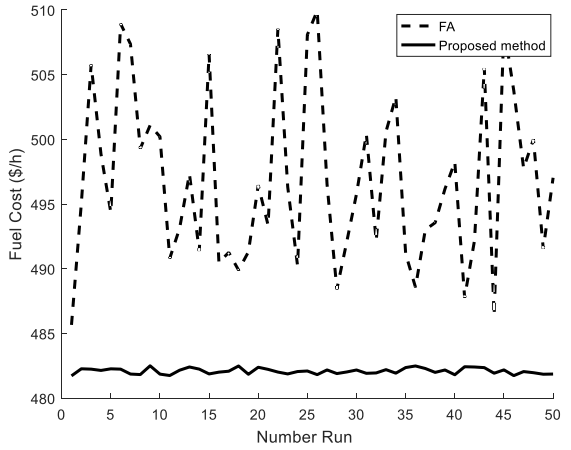

Figure 4 The best cost after fifty runs obtained by FA and the proposed method in case 4.1.

5.2 Comparison and Discussion

In order to further investigate the performance of IFA, comparisons were carried out of results obtained by IFA and other optimization tools, such as EALHN [1], HNN [3], HRCGA [4], RCGA [4], DE [5], HNUM [6], AHNN [7], ELANN [8], IEP [9], AIS [10], HICDEDP [11], Lambda [15], HM [15], TS [18], IGA [20], BBO [21-22], CGA [23] FCGA [23], CSA [24-25], and ORCSA [26]. In addition to the comparison of best cost, another comparison criterion was considered, i.e. the number of fitness evaluations N_{FES}, which is shown in Eq. (24):

$$N_{FES} = \omega \times N_{pop} \times N_{Iter}, \tag{24}$$

where ω is the number of generations in each iteration. For some optimization algorithms with two new-solution generations, CSA and ORCSA, ω is 2 while for the other two, with one new-solution generation, PSO and DE, ω is 1. For the proposed IFA, only one new solution is generated in each iteration, thus ω is equal to 1. The value of N_{FES} was added to each table for comparison, which indicates that methods with a lower N_{FES} are more efficient if its best cost is also lower or equal.

Table 5 shows the best cost, ACTFER and N_{FES} from IFA and other methods for case 1. The best cost comparison indicates that IFA has the same solution quality as BBO [21] and CSA [24] but better solution quality than TS [18] and IGA [20]. The reported time indicates faster search ability of IFA compared to CSA. No values were reported by the other methods. However, IGA and BBO used a very high number of N_{FES}, 10,000,000 for IGA and 30,000 for BBO, whereas the value was very low for IFA (150). Consequently, IFA is a very efficient method for case 1.

Table 5 Result comparisons for case 1.

Method	Cost ($)	ACTFER (s)	N_{pop}	N_{Iter}	N_{FES}
TS [18]	8344.598	-	-	-	-
IGA [20]	8344.598	-	500	20,000	10,000,000
BBO [21]	8344.592	-	100	300	30,000
CSA [24]	8344.59	0.09	-	-	-
IFA	8344.592	0.06	10	15	150

The comparisons for cases 2.1, 2.2 and 2.3 (Table 6) show that the costs from IFA were equal to those from CSA but much lower than those from FCGA [23] and CGA [23]: by $3.93 and $5.79 for case 2.1, $2.94 and $16.65 for case 2.2, and $6.52 and $9.72 for case 2.3 respectively. Clearly, IFA obtained better solutions than CGA and FCGA for the three cases. Furthermore, IFA used only

400 fitness evaluations while CGA and FCGA used 10,000 fitness evaluations. CSA does not report its population and iterations, thus we cannot calculate its N_{FES}. Consequently, IFA is very promising for cases 2.1, 2.2 and 2.3. Comparisons with other methods, i.e. Lambda [15], HM [15], BBO [22], CSA [25] and ORCSA [25] for case 3, are given in Table 7. The best cost comparison shows that the method yielded much better cost than FA and the same or approximate solution quality as the other methods. However, Lambda and HM had a high error rate (about 10^{-3}), while the other methods and IFA had low error rates.

Table 6 Result comparisons for cases 2.1, 2.2 and 2.3.

Method	Case 2.1	Case 2.2	Case 2.3	
	Cost ($)	Cost ($)	Cost ($)	N_{FES}
FCGA [23]	8231.030	11480.030	16585.850	10,000
CGA [23]	8232.890	11493.740	16589.050	10,000
CSA [24]	8227.100	14477.090	16579.330	-
IFA	8227.0986	11477.09	16579.33	400

Table 7 Result comparisons for case 3.

Method	Cost ($)	N_{pop}	N_{Iter}	N_{FES}
Lambda [15]	624656.639	-	-	-
HM [15]	62456.6341	-	-	-
CSA [25]	62456.633	10	500	10,000
ORCSA [25]	62456.633	10	500	10,000
BBO [22]	62456.7926	50	400	20,000
IFA	62456.638	20	500	10,000

Moreover, IFA used the same N_{FES} as CSA, ORCSA but half that of BBO. Clearly, IFA is also an effective method for case 3. For the multi-fuel cases, the best cost and fitness evaluations are shown in Table 8.

Table 8 Comparison of best cost (in $/h) for case 4.

Method	Case 4.1	Case 4.2	Case 4.3	Case 4.4	N_{FES}
EALHN [1]	481.723	526.239	574.381	623.809	-
HNN [3]	487.780	526.130	574.260	626.120	-
HRCGA [4]	481.7226	526.2388	574.3808	623.8092	8,000
RCGA [4]	481.7233	526.2393	574.3966	623.8094	8,000
DE [5]	481.723	526.239	574.381	623.809	12,000
HNUM [6]	488.500	526.700	574.030	625.180	-
AHNN [7]	481.720	526.230	574.370	626.240	-
ELANN [8]	481.740	526.270	574.410	623.880	-
IEP [9]	481.779	526.304	574.473	623.851	-
AIS [10]	481.723	526.24	574.381	623.809	3,000
HICDEDP [11]	481.723	526.239	574.381	623.809	4,000
FA	485.661	528.11	577.003	627.887	3,000
IFA	481.723	526.240	574.381	623.810	3,000

The best cost comparison indicates that IFA had the same optimal solution quality as most methods, excluding a number of methods that had higher cost, i.e. HNN [3], HNUM [6], ELANN [8], and IEP [9]. The proposed method especially had much better cost than FA. Furthermore, IFA was one of methods with the lowest N_{FES} value (3,000), while RCGA and HRCGA in [4] needed 8,000, DE [5] needed 12,000 and HICDEDP [11] needed 4,000. Clearly, IFA is one of the most efficient methods, being able to find the lowest fuel cost and using the smallest number of fitness evaluations.

6 Conclusions

In this paper, two improvements of the conventional firefly algorithm were proposed. The first improvement was to determine the effective distance between two considered solutions and the second improvement was aimed at finding an efficient algorithm for generating new solutions. Each improvement had a significant impact on the performance of the proposed IFA since IFA1 (with application of the first improvement) and IFA2 (with application of the second improvement) performed better than conventional FA. The proposed IFA with both improvements also performed much better than FA.

Further investigation of the proposed IFA was done by comparing its performance with that of several other methods in nine cases, considering the single-fuel and multi-fuel options. Result comparisons indicated that IFA can obtain high approximate solution quality or better solutions than the other methods while using a lower or equal number of fitness evaluations compared to these methods. Consequently, the proposed IFA is very promising for solving the problem of optimal operation of thermal generating units.

References

[1] Vo, D.N. & Ongsakul, W., *Economic Dispatch with Multiple Fuel Types by Enhanced Augmented Lagrange Hopfield Network*, Applied Energy, **91**(1), pp. 281-289, 2012.

[2] Tran, C.D., Nguyen, T.T., Hoang, H.M. & Nguyen, B.Q., *One Rank Cuckoo Search Algorithm for Bi-Objective Load Dispatch Problem*, International Journal of Grid and Distributed Computing, **9**(4), pp. 13-26, 2016.

[3] Park, J.H., Kim, Y.S., Eom, I.K. & Lee, K.Y., *Economic Load Dispatch for Piecewise Quadratic Cost Function Using Hopfield Neural Network*, IEEE Transactions on Power Systems, **8**(3), pp. 1030-1038, 1993.

[4] Baskar, S., Subbaraj, P. & Rao, M.V.C., *Hybrid Real Coded Genetic Algorithm Solution to Economic Dispatch Problem*, Computers & Electrical Engineering, **29**(3), pp. 407-419, 2003.

[5] Noman, N. & Iba, H., *Differential Evolution for Economic Load Dispatch Problems*, Electric Power Systems Research, **78**(8), pp. 1322-1331, 2008.

[6] Lin, C.E. & Viviani, G.L., *Hierarchical Economic Dispatch for Piecewise Quadratic Cost Functions*, IEEE Transactions on Power Apparatus and Systems, **6**, pp. 1170-1175, 1984.

[7] Lee, K.Y., Sode-Yome, A. & Park, J.H., *Adaptive Hopfield Neural Networks for Economic Load Dispatch*, IEEE Transactions on Power Systems, **13**(2), pp. 519-526, 1998.

[8] Lee, S.C. & Kim, Y.H., *An Enhanced Lagrangian Neural Network for the ELD Problems with Piecewise Quadratic Cost Functions and Nonlinear Constraints*, Elect Power, Syst Res, **60**(3), pp. 167-77, 2002.

[9] Park, Y.M., Wong, J.R. & Park, J.B., *A New Approach to Economic Load Dispatch Based on Improved Evolutionary Programming*, Eng Intell Syst Elect Eng Commun, **6**(2), pp. 103-10, 1998.

[10] Panigrahi, B.K., Yadav, S.R., Agrawal, S. & Tiwari, M.K., *A Clonal Algorithm to Solve Economic Load Dispatch*, Electric Power Systems Research, **77**(10), pp. 1381-1389, 2007.

[11] Balamurugan, R. & Subramanian, S., *Hybrid Integer Coded Differential Evolution–Dynamic Programming Approach for Economic Load Dispatch with Multiple Fuel Options*, Energy Convers Manage, **49**(4), pp. 608-614, 2008.

[12] Bard, J.F., *Short-Term Scheduling of Thermal-Electric Generators Using Lagrangian Relaxation*, Operations Research, **36**(5), pp. 756-766, 1988.

[13] Dodu, J.C., Martin, P., Merlin, A. & Pouget, J., *An Optimal Formulation and Solution of Short-Range Operating Problems for a Power System with Flow Constraints*, Proceedings of the IEEE, **60**(1), pp. 54-63, 1972. DOI: 10.1109/PROC.1972.8557

[14] Chen, C.L. & Wang, S.C., *Branch-And-Bound Scheduling for Thermal Generating Units*, IEEE Transactions on Energy Conversion, **8**(2), pp. 184-189, 1993.

[15] Su, C.T. & Lin, C.T., *New Approach with A Hopfield Modeling Framework to Economic Dispatch*, IEEE Transactions on Power Systems, **15**(2), pp. 541-545, 2000.

[16] King, T.D., El-Hawary, M.E. & El-Hawary, F., *Optimal Environmental Dispatching of Electric Power Systems Via an Improved Hopfield Neural Network Model*, IEEE Transactions on Power Systems, **10**(3), pp.1559-1565, 1995.

[17] Vo, D.N. & Ongsakul, W., *Economic Dispatch with Multiple Fuel Types by Enhanced Augmented Lagrange Hopfield Network*, Applied Energy, **91**(1), pp. 281-289, 2012.

[18] Roa, C., Salazar, E., Gracia, E., Knight, U.G. & Coonick, A., *Environmental Economic Dispatch Via Hopfield Neural Network and Taboo Search*, In 31st Universities Power Engineering Conference Proceedings, Irakilo, Greece., pp. 18-20, 1996.

[19] Mandal, K.K. & Chakraborty, N., *Effect of Control Parameters on Differential Evolution Based Combined Economic Emission Dispatch with Valve-Point Loading and Transmission Loss*, International Journal of Emerging Electric Power Systems, **9**(4), 2008.

[20] Rughooputh, H.C. & King, R.A., *Environmental/Economic Dispatch of Thermal Units Using an Elitist Multiobjective Evolutionary Algorithm, In Industrial Technology*, IEEE International Conference, pp. 48-53, 2003.

[21] Roy, P.K., Ghoshal, S.P. & Thakur, S.S., *Combined Economic and Emission Dispatch Problems Using Biogeography-Based Optimization*, Electrical Engineering, **92**(4-5), pp. 173-184, 2010.

[22] Bhattacharya, A. & Chattopadhyay, P.K., *Biogeography-Based Optimization for Different Economic Load Dispatch Problems*, IEEE Transactions on Power Systems, **25**(2), pp. 1064-1077, 2010.

[23] Song, Y.H., Wang, G.S., Wang, P.Y. & Johns, A.T., *Environmental/Economic Dispatch Using Fuzzy Logic Controlled Genetic Algorithms*, IEE Proceedings-Generation, Transmission and Distribution., pp. 377-382, 1997.

[24] Thao, N.T.P. & Thang, N.T., *Environmental Economic Load Dispatch with Quadratic Fuel Cost Function Using Cuckoo Search Algorithm*, International Journal of u- and e-Service, Science and Technology, **7**(2), pp. 199-210, 2014.

[25] Nguyen, T.T. & Vo, D.N., *The Application of One Rank Cuckoo Search Algorithm for Solving Economic Load Dispatch Problems*, Applied Soft Computing, **37**, pp. 763-773, 2015.

[26] Le, K.C., Dinh, B.H., & Nguyen, T.T., *Environmental Economic Hydrothermal System Dispatch by Using a Novel Differential Evolution*, Journal of Engineering and Technological Sciences, **50**(1), pp.1-20, 2018.

[27] Yang, X.S., *Nature-Inspired Metaheuristic Algorithms*, 1st, Frome, UK: Luniver Press, Bristol, United Kingdom, 2008.