Iranian Journal of Science and Technology, Transactions of Electrical Engineering (2019) 43:1031–1051
https://doi.org/10.1007/s40998-019-00211-9

RESEARCH PAPER

An Efficient Adaptive Moth Flame Optimization Algorithm for Solving Large-Scale Optimal Power Flow Problem with POZ, Multifuel and Valve-Point Loading Effect

Hitarth Buch[1,2] · Indrajit N. Trivedi[3]

Received: 12 October 2018 / Accepted: 25 May 2019 / Published online: 4 June 2019
© Shiraz University 2019

Abstract
This paper offers an enhanced adaptive moth flame optimization (AMFO) algorithm to solve the optimal power flow (OPF) problems efficiently. The idea of moth flame optimization (MFO) is motivated by the movement of moth headed about the moon direction. AMFO is primarily centered on the notion of MFO with adjusting the direction of moths in an adaptive manner around the flame. AMFO is compared with standard MFO for 14 different benchmark test suites. Standard IEEE 118-bus test system is used to substantiate the effectiveness and robustness of AMFO algorithm. The authentication of the suggested algorithm is established on 12 case studies for various single-objective functions like fuel cost minimization, emission minimization, active power loss minimization, voltage stability enhancement and voltage profile improvement. The simulation findings of the suggested algorithm are compared with those found by other well-known optimization methods. The achieved results demonstrate the ability and strength of AMFO approach to solving OPF problems. The outcomes divulge that AMFO algorithm can obtain accurate and improved OPF solutions compared with the other methods. A comparison among the convergence qualities of AMFO and the different techniques demonstrates the predominance of AMFO to achieve the optimal power flow solution with rapid convergence.

Keywords Adaptive moth flame algorithm · Optimal power flow · Power system optimization

1 Introduction

Since the initiation of the concept—the optimal power flow by Carpentier (1962) in 1962, the OPF worth has been increasingly recognized, and in the present time, it has developed to be the most critical and essential instrument to determine the most cost-effective and secure state of planning and operation of power system. Diverse models have been established and adopted to form numerous types of OPF problems, objectives, set of state/control variables and limitations.

The OPF is an optimization problem which aims to fine-tune continuous and discrete control variables to enhance a predefined objective function while accomplishing operational equality and inequality limitations. Conventionally, the persistence of the OPF was to minimize the overall generating cost, i.e., economic dispatch. However, difficulties and restrictions like multifuels, valve-point effect, security constraint, prohibited zones must be incorporated to examine the more realistic OPF problem covering optimization of emission, voltage deviation and stability, active and reactive power loss, etc. The presence of these complexities makes the OPF problem an extremely constrained, mixed-integer, nonlinear and non-convex problem.

Primarily, numerous deterministic techniques were employed to solve the OPF problem. Such methods are found suitable for convex, smooth, continuous and differentiable objective functions. The optimal power flow (OPF) is a nonlinear, non-convex, intermittent problem owing to the presence of multifuels, valve-point effect, and prohibited

✉ Hitarth Buch
hitarth_buch_020@gtu.edu.in

Indrajit N. Trivedi
int@gecg28.ac.in

1 Gujarat Technological University, Ahmedabad 382424, Gujarat, India

2 Department of Electrical Engineering, Government Engineering College, Rajkot 360005, Gujarat, India

3 Department of Electrical Engineering, Government Engineering College, Gandhinagar 389001, Gujarat, India

zones, etc., and hence, gradient-based approaches fail to solve the OPF problem. Authors can find a thorough survey of deterministic methods in Pandya and Joshi (2005).

The evolutionary methods, i.e., metaheuristics, have witnessed tremendous development in the past decade. These techniques optimize a problem by attempting to improve a candidate solution iteratively about the given measure of quality. Metaheuristics make almost no assumptions regarding the problem being solved and are derivative free. The benefit of metaheuristics is that objective function can be discontinuous and differentiable as they do not employ a gradient search or Hessian matrix. Nevertheless, metaheuristics never ensure the optimal solution. Also, as suggested by no free lunch theorem (Wolpert and Macready 1997), if an algorithm works well on a particular category of problems, then it necessarily works degraded for another type of problem. It implies that the average rank of all the algorithms is same. This aspect has motivated the development of new and the enhancement of current approaches leading to a new form of metaheuristics.

As already stated, a specific metaheuristic may produce very efficient outcomes on a set of problems, but the same algorithm may show poor performance on another set of problems. This limitation has led to the application of different methods to solve an issue like the OPF problem. Various techniques such as genetic algorithm (GA) (Paranjothi and Anburaja 2002; Lai et al. 1997), particle swarm optimization (PSO) (Abido 2002a), tabu search algorithm (TS) (Abido 2002b), Simulated Annealing (SA) (Roa-Sepulveda and Pavez-Lazo 2001), differential evolution (DE) (Abou El Ela et al. 2010), imperialist competitive algorithm (ICO) (Ghanizadeh et al. 2011), harmony search algorithm (HAS) (Sinsuphan et al. 2013), black hole (BH) (Bouchekara 2014), teaching–learning-based algorithm (TLBO) (Bouchekara et al. 2014a), moth flame optimization algorithm (MFO) (Trivedi et al. 2016a; Buch et al. 2017), artificial bee colony algorithm (ABC), moth swarm algorithm (MSA) (Mohamed et al. 2017), league championship algorithm (LCA) (Bouchekara et al. 2014b), backtracking search algorithm (BSA) (Chaib et al. 2016), improved colliding bodies algorithm (ICBO) (Bouchekara et al. 2016), hybrid genetic-teaching–learning-based algorithm (H-TLBO) (Güçyetmez and Çam 2016), glowworm swarm optimization algorithm (GSA) (Surender Reddy et al. 2014), Krill Herd Algorithm (KHA) (Mukherjee and Mukherjee 2015), multi-verse optimizer (MVO) (Trivedi et al. 2016b), bat algorithm (BA) (Trivedi et al. 2016a) and many others are employed to solve the OPF problem. A comprehensive study of various metaheuristics applied to solve the OPF problem is presented in Niu et al. (2014), AlRashidi and El-Hawary (2009) and Frank et al. (2012).

In Buch and Trivedi (2018), eight different algorithms are compared to optimize the OPF problem. However, due to the complex kind of objectives included in the OPF problem, there is a continual need to apply a new and enhanced algorithm that can optimize the OPF problem reasonably. This fact has motivated us to develop and present an enhanced version of the moth flame optimization algorithm, i.e., adaptive moth flame optimization. The objective of this paper is to evaluate the performance of AMFO with MFO (Mirjalili 2015a) for optimizing the benchmark functions and also to implement AMFO for solving OPF on medium size test system, and compare its results with MFO (Mirjalili 2015a), GWO (Mirjalili et al. 2014), DA (Mirjalili 2016), SCA (Mirjalili 2015b), ALO (Mirjalili 2015c), MVO (Mirjalili et al. 2016), GOA (Saremi et al. 2017) and IMO (Javidy et al. 2015).

The key contributions of this work are summarized below:

1. Development of improved version of MFO, i.e., adaptive MFO and its implementation on standard benchmark functions.
2. A solution of the realistic OPF problem embedded with practical restraints like prohibited zones (POZ), valve-point effect (VP) and multifuels (MF) on a 118-bus test system.
3. Implementation of a complete set of tests to assess AMFO using different OPF problems on 118-bus test systems with different objective functions and limitations.
4. Utilization of nonparametric statistical assessments like Quade test (Quade 1979), Friedman (1937) and Friedman aligned test (Friedman 1940) for confirmation of results.

The structure of the rest of the paper is as follows: In Sect. 2, the OPF problem is framed. In Sect. 3, standard MFO is described in brief. Section 4 focuses on newly introduced adaptive MFO and its performance assessment on standard benchmark functions. In Section 5, the applications and results for solving the 118-bus OPF problem are discussed. Section 6 deals with the evaluation of AMFO based on the statistical test while the conclusion is drawn in the last section.

1.1 Devising the Optimal Power Flow (OPF) Problem

The optimal power flow is a problem which offers the best possible settings of the control variables for a specified set of the load by curtailing a predefined objective function such as the cost of power generation, voltage deviation, voltage stability index or transmission line losses. The majority of optimal power flow formulations may be characterized using the following standard equations:

$$\text{Minimize} \quad J(x, u) \tag{1}$$

$$\text{Subject to } g(x, u) = 0 \tag{2}$$

$$\text{and } h(x, u) \leq 0 \tag{3}$$

Here u represents the vector of independent variables or control variables. x represents the vector of dependent variables or state variables. $J(x, u)$ represents the system's optimization goal or objective function. $g(x, u)$ represents the set of equality constraints. $h(x, u)$ represents the set of inequality limitations.

1.2 Control Variables

These variables are adjusted to meet the load flow equations. The collection of control variables in the optimal power flow is as follows:

- P_G symbolizes active power generation at the PV buses except for the slack bus
- V_G symbolizes the voltage magnitude at PV buses
- T represents the tap setting of the transformer
- Q_c signifies the shunt VAR compensation

Hence, u can be expressed as:

$$u^T = \left[P_{G_2} \dots P_{G_{NG}}, V_{G_1} \dots V_{G_{NG}}, Q_{C_1} \dots Q_{C_{NC}}, T_1 \dots T_{NT}\right] \tag{4}$$

where NG, NT and NC represent the number of generators, regulating transformers and VAR compensators, respectively.

1.3 State Variables

State variables represent the electrical state of systems. State variables are given as follows:

- P_{G1} is active power output at slack bus
- V_L symbolizes the voltage magnitude at PQ buses, load buses
- Q_G is the reactive power output of all generator units
- S_l is the transmission line loading (or line flow)

Hence, x can be expressed as:

$$x^T = \left[P_{G_1}, V_{L_1} \dots V_{L_{NL}}, Q_{G_1} \dots Q_{G_{NG}}, S_{l_1} \dots S_{l_{nl}}\right] \tag{5}$$

where NL and nl are the numbers of load buses and the number of transmission lines, respectively.

1.4 Constraints

The OPF constraints can be classified into equality and inequality constraints, which are described in the following subsections:

1.5 Equality Limits

The equality constraints reflect the behavior of the power system. The equality constraints are as follows:

- Real power limits

$$P_{Gi} - P_{Di} - V_i \sum_{j=1}^{NB} V_j \left[G_{ij} \cos(\theta_{ij}) + B_{ij} \sin(\theta_{ij})\right] = 0 \tag{6}$$

- Reactive power limits

$$Q_{Gi} - Q_{Di} - V_i \sum_{j=1}^{NB} V_j \left[G_{ij} \sin(\theta_{ij}) - B_{ij} \cos(\theta_{ij})\right] = 0 \tag{7}$$

where θ_{ij} is the bus voltage angle difference between bus i and j, i.e., $\theta_{ij} = \theta_i - \theta_j$, NB is the number of buses, P_G is the active power generation, Q_G is the reactive power generation, P_D and Q_D are active and reactive power demand, respectively, and G_{ij} and B_{ij} are elements of the admittance matrix representing conductance and susceptance between bus i and j, respectively.

1.6 Inequality Constraints

The inequality constraints represent the restrictions on physical gadgets present in the power system. These constraints also present the limits created to guarantee system security. These inequality constraints are as follows.

1.7 Generator Limits

For all generators comprising a slack bus, voltage, active and reactive outputs should be restricted by their upper and lower boundaries as follows:

$$\begin{aligned} V_{G_i}^{\min} &\leq V_{G_i} \leq V_{G_i}^{\max} \quad i = 1, \dots, NG \\ P_{G_i}^{\min} &\leq P_{G_i} \leq P_{G_i}^{\max} \quad i = 1, \dots, NG \\ Q_{G_i}^{\min} &\leq Q_{G_i} \leq Q_{G_i}^{\max} \quad i = 1, \dots, NG \end{aligned} \tag{8}$$

$V_{G_i}^{\min}$ and $V_{G_i}^{\max}$ are the minimum and maximum bus voltage limits, $P_{G_i}^{\min}$ and $P_{G_i}^{\max}$ are the minimum and maximum active power generation limits, while $Q_{G_i}^{\min}$ and $Q_{G_i}^{\max}$ are the minimum and maximum reactive power generation limits.

1.8 Transformer Constraints

Transformer tap settings should be confined within their specified lower and upper limits as follows:

$$T_i^{\min} \le T_i \le T_i^{\max} \quad i = 1, 2, \ldots, NT \tag{9}$$

$T_i^{\min}$ and $T_i^{\max}$ are the minimum and maximum transformer tap setting limits.

1.9 Shunt VAR Compensator Constraints

Shunt VAR Compensator settings are to be confined within their specified lower and upper bounds as follows:

$$Q_{C_i}^{\min} \le Q_{C_i} \le Q_{C_i}^{\max} \quad i = 1, \ldots, NC \tag{10}$$

$Q_{c_i}^{\min}$ and $Q_{c_i}^{\max}$ are the minimum and maximum shunt reactive power compensation limits.

1.10 Security Constraints

Transmission line loadings and voltage magnitude at load buses are part of this category. The voltage of each load bus V_{L_i} must be confined within its lower and upper operating limits, i.e., $V_{L_i}^{\min}$ and $V_{L_i}^{\max}$. Line flow S_{l_i} through every transmission line is restricted by its capacity limits $S_{l_i}^{\max}$. These constraints can be mathematically formulated as follows:

$$\begin{aligned} V_{L_i}^{\min} \le V_{L_i} \le V_{L_i}^{\max} \quad & i = 1, \ldots, NL \\ S_{l_i} \le S_{l_i}^{\max} \quad & i = 1, \ldots, nl \end{aligned} \tag{11}$$

1.11 Adaptive Moth Flame Optimization Algorithm

1.11.1 Inspiration

Moths are tiny insects quite like the family of butterflies. The sheer motivating fact about moths is their distinct steering approaches during the night. They maintain a fixed angle concerning the moon to travel long expanses over a straight line. Due to the more considerable distance among the moth and the moon, such mechanism guarantees flying in a straight line. Artificial lights trick moths, and hence, moths fly spirally around the human-made lights. Moths try to retain a similar tilt for artificial light to fly in a straight line. As artificial lights are incredibly close to the moth, keeping a similar angle triggers a dangerous spiral route for moths. In result, the moth ultimately converges on the artificial light. Figure 1a, b presents movement of moths about the moon and an artificial light, respectively.

1.12 MFO Algorithm

MFO algorithm articulates the spiral movement of moth toward the flame (light). In the present algorithm, moths and position of moths represent candidate solutions and problem's variables, respectively. The set of moths M is represented in the following matrix:

$$M = \begin{pmatrix} m_{11} & \ldots & m_{1d} \\ \vdots & \ddots & \vdots \\ m_{n1} & \cdots & m_{nd} \end{pmatrix} \tag{12}$$

where n is the number of moths and d is the number of variables, respectively.

The following array presents sorted fitness values.

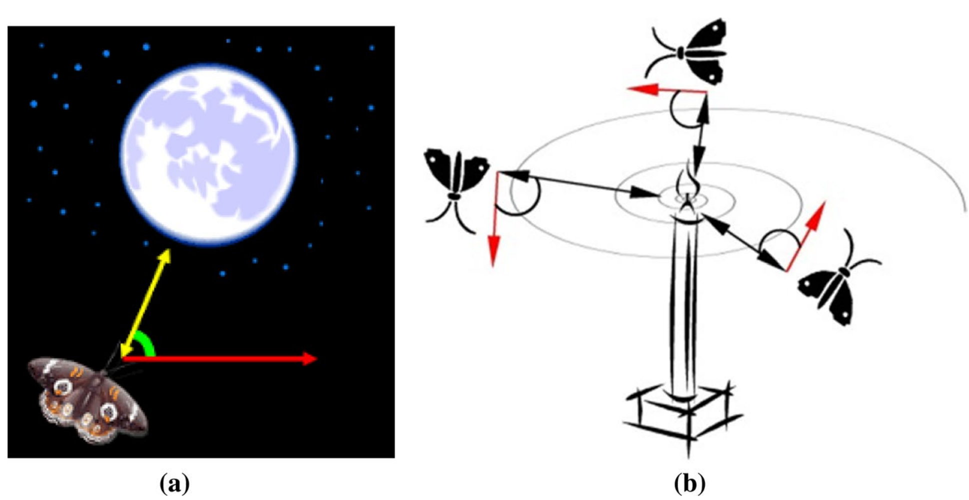

Fig. 1 Movement of moth with respect to moon and artificial light (Mirjalili 2015a)

$$OM = \begin{bmatrix} OM_1 \\ OM_2 \\ \vdots \\ OM_n \end{bmatrix} \tag{13}$$

All moths are passed through the fitness function, and their return value is the fitness value of objective function. OM array is identical to the fitness function output. Flames represent an underlying matrix of MFO. Flames matrix can be presented as follows:

$$F = \begin{pmatrix} f_{11} & \cdots & f_{1d} \\ \vdots & \ddots & \vdots \\ f_{n1} & \cdots & f_{nd} \end{pmatrix} \tag{14}$$

The dimension of the flame's matrix and moth's matrix is equal. Flames are sorted as per the following array based on the fitness values.

$$OF = \begin{bmatrix} OF_1 \\ OF_2 \\ \vdots \\ OF_n \end{bmatrix} \tag{15}$$

Another critical component in the suggested approach is flames, which present the best position of moths. Hence, every moth searches about the flame and revises itself in the case of obtaining a superior solution. Such mechanism assures that moth never loses its best position.

The position of each moth with respect to flame is updated using the following expression:

$$M_i = S(M_i, F_i) \tag{16}$$

where M_i indicates the ith moth, F_j presents the jth flame, and S is the spiral function. The spiral function is defined as:

$$S(M_i, F_j) = D_i e^{bt} \cos(2\pi t) + F_j \tag{17}$$

Here D_i shows the distance between ith moth and jth flame, b is a constant identifying the shape of spiral, and t is a random number between $[-1, 1]$.

$$D_i = \left|F_j - M_i\right| \tag{18}$$

Equation (17) represents spiral search path of the flying moths while updating their position with respect to flames. Parameter t presents closeness of moth with flame. When $t = -1$, the moth is closest to the flame, while when $t = 1$, the moth is furthest from the flame. Flames are considered as best solutions to enhance the search around the better solutions. Moths update their positions with respect to flames according to Eqs. (16) and (17).

The position updating in Eq. (17) necessitates the moths to move toward a flame, yet it causes the MFO algorithm to be stuck in local optima rapidly. To avoid this, each moth revises its position utilizing only one of the flames in Eq (17). Another worry here is that the position updating of moths with respect to the number of different flames in the search space may damage the exploitation of the best capable results. To solve this concern, an adaptive mechanism is devised for the number of flames. With the increasing the number of iterations, the number of flames decreases as per Eq. (19).

$$\text{Flame no.} = N - l * \left(\frac{N-1}{T}\right) \tag{19}$$

where l is the current iteration, N is a maximum number of flames, and T is the maximum number of iterations. The progressive reduction in the number of flames throughout iterations balances among exploration and exploitation within the search space.

1.13 Adaptive MFO

In stochastic metaheuristics methods, introducing randomization plays a vital role. Authors in Bhesdadiya et al. (2017), Li et al. (2016) and Soliman et al. (2016) presented different variants of MFO using Levy flight and Cauchy operators. With the standard scenario, the MFO updates its agents toward the candidate solution based on Eq. (17). The efficacy of the standard MFO (Mirjalili 2015a) is undeniable, meaning that when assumed adequate computation period, it is sure to converge to the optimum answers ultimately. However, the search process may be sluggish. To improve the convergence rate while upholding the noticeable physiognomies of the MFO, an enhanced searching procedure which is, likewise to the adaptive cuckoo search algorithm (Ong 2014; Kumar et al. 2015) is presented here. Figure 2 shows a flowchart of the adaptive MFO approach.

The standard MFO algorithm updates the moth position based on the distance of moth with respect to the flame. Here, we attempt to include the step size based on the best and worst moth position as well as current moth position. The step size determines how far a new moth position is located from the current position. As presented in Eq. (20), step size varies inversely with generation, i.e., with an increase in the iteration, step size reduces. As shown in Eq. (21), the calculated step size is added to the current moth position to obtain a new moth position.

$$X_i^{t+1} = \left(\frac{1}{t}\right)^{\left|\frac{(\text{best} f(t) - f_i(t))}{\text{best} f(t) - \text{worst} f(t)}\right|} \tag{20}$$

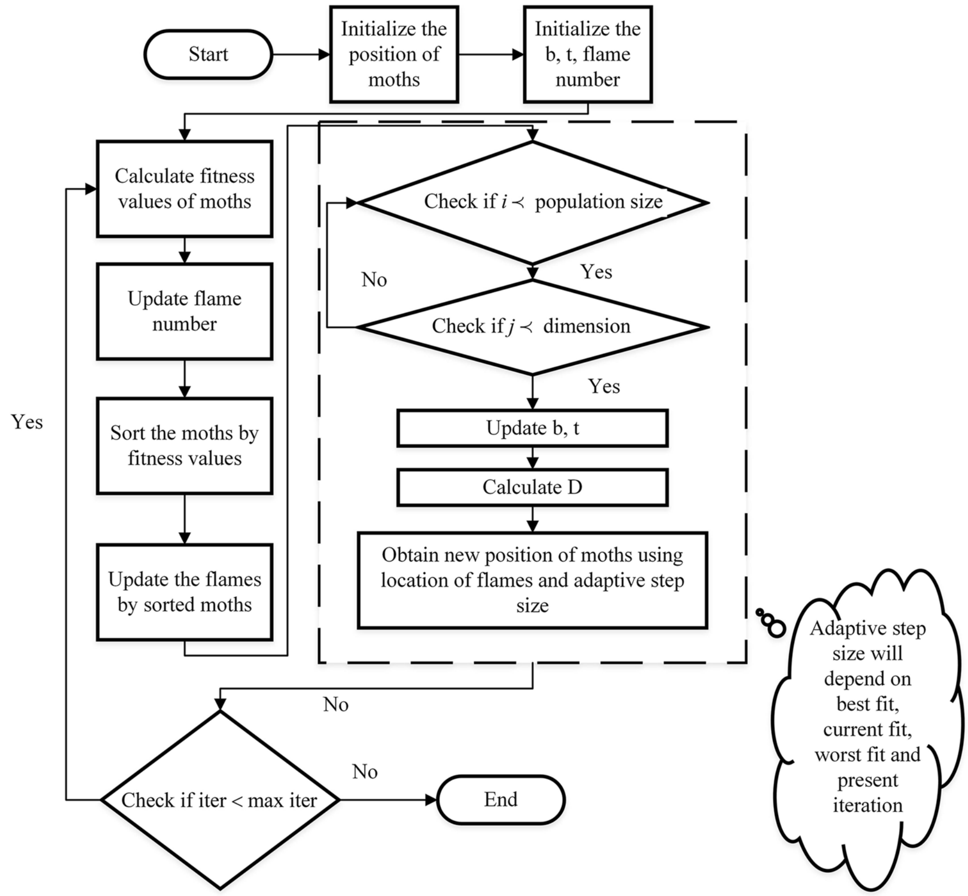

Fig. 2 Flowchart of AMFO

$$\text{Moth\_pos}(t+1) = \text{Moth\_pos}(t) + p * X_i^{t+1} \tag{21}$$

In Eq. (21), is a random number between [0, 1] introducing arbitrary component in position update equation. Next subsection presents performance assessment of AMFO on various standard single-objective benchmark functions.

1.14 Results on Benchmark Test Functions

To assess the proposed adaptive moth flame optimization (MFO) algorithm, we carried out the performance study on 14 well-known benchmark functions (Yao et al. 1999). The benchmark functions used are of two types: unimodal and multimodal with flexible dimension. Unimodal functions are those which have only one local minimum, while multimodal ones have multiple local minima. Table 1 presents details of benchmark functions, their mathematical depiction, the range of search and theoretical ideal values.

Function F1 is continuous, convex and unimodal having n global minima except the global one. The second function, i.e., Schwefel's function 2.22, is a continuous, convex, unimodal, non-differentiable and separable function. Function F3 is Schwefel's function 1.2, an extension of the axis parallel hyper-ellipsoids. This function is also continuous, convex and unimodal. Function F4 is also continuous, convex, unimodal, non-differentiable and separable function like F2. The fifth function, i.e., Rosenbrock function, is non-convex function. The global minimum is inside a long, narrow parabolic-shaped valley. Step function is the demonstrative of the problem of flat surfaces. Flat surfaces do not guide algorithms about search in favorable directions. Unless the algorithm has variable step size, the algorithm is likely to get stuck at one of the plateaus. Thus, the step function makes the search process more difficult by injecting small plateaus in continuous function. The quartic function, i.e., F7, is continuous, non-convex, multimodal, differentiable and separable function. Schwefel's function is somewhat easier than Rastrigin's function and is characterized by a second-best minimum which is far away from the global optimum. In optimization studies, Rastrigin function (F9) is a fairly difficult problem to optimize because of its large search space and large number of local minima. It is a typical example of nonlinear multimodal non-convex problem. The Ackley function (F10) is extensively applied for analysis of optimization algorithms. It has a nearly flat outer region and a large hole at the center. The function presents a threat for optimization algorithms to be trapped in one of its various local minima. The generalized Griewank function (F11) has

Table 1 Benchmark functions used

BF	Mathematical depiction	Dim	ROS	TI
Sphere (F1)	$f(x) = \sum_{i=1}^{n} x_i^2 * R(x)$	10	[− 100, 100]	0
Schwefel's function 2.22 (F2)	$f(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert * R(x)$	10	[−10, 10]	0
Schwefel's function 1.2 (F3)	$f(x) = \sum_{i=1}^{n} \left(\sum_{j=1}^{n} x_j \right)^2 * R(x)$	10	[− 100, 100]	0
Schwefel's function 2.21 (F4)	$f(x) = \max_i \left\{ \lvert x_i \rvert, 1 \le i \le n \right\}$	10	[− 100, 100]	0
Generalized Rosenbrock's function (F5)	$f(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2] * R(x)$	10	[− 30, 30]	0
Step function (F6)	$f(x) = \sum_{i=1}^{n} ([x_i + 0.5])^2 * R(x)$	10	[−100, 100]	0
Quartic function, i.e., noise (F7)	$f(x) = \sum_{i=1}^{n} i x_i^4 + \mathrm{random}[0,1] * R(x)$	10	[− 1.28, 1.28]	0
Generalized Schwefel's problem 2.26 (F8)	$F(x) = \sum_{i=1}^{n} -x_i \sin\left(\sqrt{\lvert x_i \rvert} \right) * R(x)$	10	[− 500, 500]	0
Generalized Rastrigin's function (F9)	$F(x) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10] * R(x)$	10	[− 5.12, 5.12]	0
Ackley's function (F10)	$F(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2} \right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi x_i) \right) + 20 + e * R(x)$	10	[− 32, 32]	0
Generalized Griewank function (F11)	$F(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}} \right) + 1 * R(x)$	10	[− 600, 600]	0
Generalized penalized function 1 (F12)	$F(x) = \frac{\pi}{n}\left\{ \begin{array}{c} 10\sin(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 \\ [1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2 \end{array} \right\}$ $y_i = 1 + \frac{x_i + 1}{4}$ $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a < x_i < a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	10	[− 50, 50]	0
Generalized penalized function 2 (F13)	$F(x) = 0.1\left\{ \begin{array}{l} \sin^2(3\pi x_1) + \sum_{i=1}^{n} (x_i - 1)^2 \\ [1 + \sin^2(3\pi x_i + 1)] \\ + (x_n - 1)^2[1 + \sin^2(2\pi x_n)] \end{array} \right\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4) * R(x)$	10	[− 50, 50]	0
De Jong (Shekel's Foxholes) (F14)	$F(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{ij})^6} \right)^{-1}$	2	[− 65.536, 65.536]	1

BF benchmark functions, *Dim* dimension, *ROS* range of search, *TI* theoretical ideals

many widespread regularly distributed local minima. The generalized penalized function is a multimodal non-convex benchmark test function. The fourteenth function, i.e., De Jong (Shekel's Foxholes) (F14), is multimodal benchmark function with very sharp drops on a primarily even surface. Table 1 presents each benchmark function, their associated variables and applicable limits.

Our primary focus is to augment the basic moth flame search algorithm for improved searchability. Table 2 compares the results of AMFO with MFO. The comparison criteria are the average value, best value, standard deviation and average simulation time. The best results are bold faced. Figure 3 presents the convergence curve of AMFO and MFO for each test function. For almost all test functions, AMFO presents smooth convergence characteristics.

Figure 4 presents a comparison between AMFO and MFO for optimizing standard benchmark functions. Average simulation time presents a calculated central value of a set of numbers, while the standard deviation is a measure stating by how much the elements of a group vary from the mean value of a group. Standard deviation is a measure of the distribution of data, while the average value presents the "center of mass" of data. Higher standard deviation represents a higher deviation from the mean value. The algorithm having lower average value may not have lower standard deviation also. In this study, both algorithms are

Table 2 Results obtained by AMFO and MFO on standard benchmark test functions

F	Moth flame optimization (MFO)				Adaptive moth flame optimization (AMFO)			
	Average	Best	SD	Avg. time	Average	Best	SD	Avg. time
F1	1.37E−32	1.28E−28	3.75752E−28	0.611	**5.75E−34**	**1.84E−30**	**2.90466E−30**	**0.593**
F2	5.18E−20	**2.24158E−19**	**1.38605E−19**	**0.624**	**6.89E−20**	2.49226E−18	5.90337E−18	0.703
F3	4.79E−09	2.39782E−06	4.30939E−06	**0.562**	**1.16E−09**	**4.27541E−07**	**6.41762E−07**	0.641
F4	5.36E−04	0.054712809	0.115761161	0.780	**1.95E−05**	**0.037250486**	**0.062669362**	**0.722**
F5	**8.23E−03**	6.03190889	6.246514319	0.781	5.95E−02	5.1247847	**4.626332327**	**0.702**
F6	1.85E−32	**2.49647E−30**	**4.78636E−30**	**0.657**	**1.23E−32**	2.18631E−30	2.72004E−30	0.686
F7	1.78E−03	**0.005663175**	**0.002540046**	0.655	**2.41E−03**	0.00664573	0.006171464	**0.576**
F8	−3.89E+03	−3261.00077	434.0208788	0.689	**−4.07E+03**	**−3329.93773**	**349.5922389**	**0.668**
F9	6.96	18.062025	8.877682154	0.766	**2.98**	**14.2279**	**7.050150222**	**0.705**
F10	4.44E−15	**4.97381E−15**	**1.06581E−15**	0.737	4.44E−15	5.15144E−15	1.42108E−15	**0.672**
F11	**4.67E−02**	**0.13395505**	**0.047527982**	0.675	1.97E−02	0.1498496	0.122346947	**0.576**
F12	**4.83E−32**	2.22712E−29	7.93594E−29	0.688	4.76E−32	**2.64868E−30**	**7.13047E−30**	**0.532**
F13	1.35E−32	0.00164805	0.0043948	0.816	**1.60E−32**	**0.0010987**	**0.0032961**	**0.765**
F14	9.98E−01	1.59166	1.515994971	0.480	9.98E−01	**1.39443**	**0.907697672**	**0.467**

independently compared for average value and standard deviation values, respectively, to assess the overall performance of both the algorithms. It is observable that AMFO provides better results for majority criteria. Thus, to sum up, global searchability of the MFO algorithm is enhanced using an adaptive approach. AMFO provides more efficient outcomes mainly for unimodal and multimodal benchmark functions.

1.15 Solving OPF Using the AMFO Algorithm

To test the performance of AMFO on complex and larger dimensional system AMFO along with other contemporary algorithms such as moth flame optimization algorithm (MFO) (Mirjalili 2015a), grey wolf optimization algorithm (GWO) (Mirjalili et al. 2014), Dragonfly algorithm (DA) (Mirjalili 2016), sine–cosine algorithm (SCA) (Mirjalili 2015b), ant lion optimizer (ALO) (Mirjalili 2015c), multi-verse optimizer (MVO) (Mirjalili et al. 2016), Grasshopper optimization algorithm (GOA) (Saremi et al. 2017), ion motion optimization algorithm (IMO) (Javidy et al. 2015) are implemented to solve the OPF problem on standard IEEE 118-bus test system. Total of 13 different objective functions test cases are considered as presented in Table 3. In this work, the population size is selected to be 25, and each algorithm is analyzed for thirty independent runs with 500 iterations per run.

1.16 IEEE 118-Bus Test System

As shown in Table 4, the system includes fifty-four thermal units, 118 buses, 177 branches and nine transformers (Fig. 5). Table 4 also presents upper and lower bounds of voltage and transformer tap settings.

Table 5 represents cost and emission coefficients of IEEE-118 bus test system. Tables 6 and 7 present cost coefficients considering multifuel and prohibited operating zone.

1.17 Cast Studies

As previously mentioned, 13 test cases are considered in this article. These cases are listed in Table 3.

2 Results and Discussion

All algorithms have been applied to the investigated cases, and the optimal results are given in Tables 8, 9 and 10. In these tables, the best results are bold faced while the worst results are bold underlined. A detailed study of each objective function is described in the following subsections.

2.1 Minimization of Quadratic Fuel Cost

The objective function, in this issue, is to optimize the fuel cost as formulated by Eq. (22)

$$f = \left(\sum_{i=1}^{NG} a_i P_{Gi}^2 + b_i P_{Gi} + c_i \right) \tag{22}$$

where a_i, b_i and c_i are the cost coefficients of ith generator. For this case, the AMFO produces the best fuel cost solution as compared to other algorithms, whereas fuel cost obtained by SCA is the highest among the rest of the algorithms.

AMFO has smooth and speedy convergence rate as compared to other algorithms as shown in Fig. 6a. Tables 8 and 9 present a comparison of algorithms in terms of the best and average objective function value. For both the cases, AMFO obtains the least value proving its superiority. MFO has least simulation time.

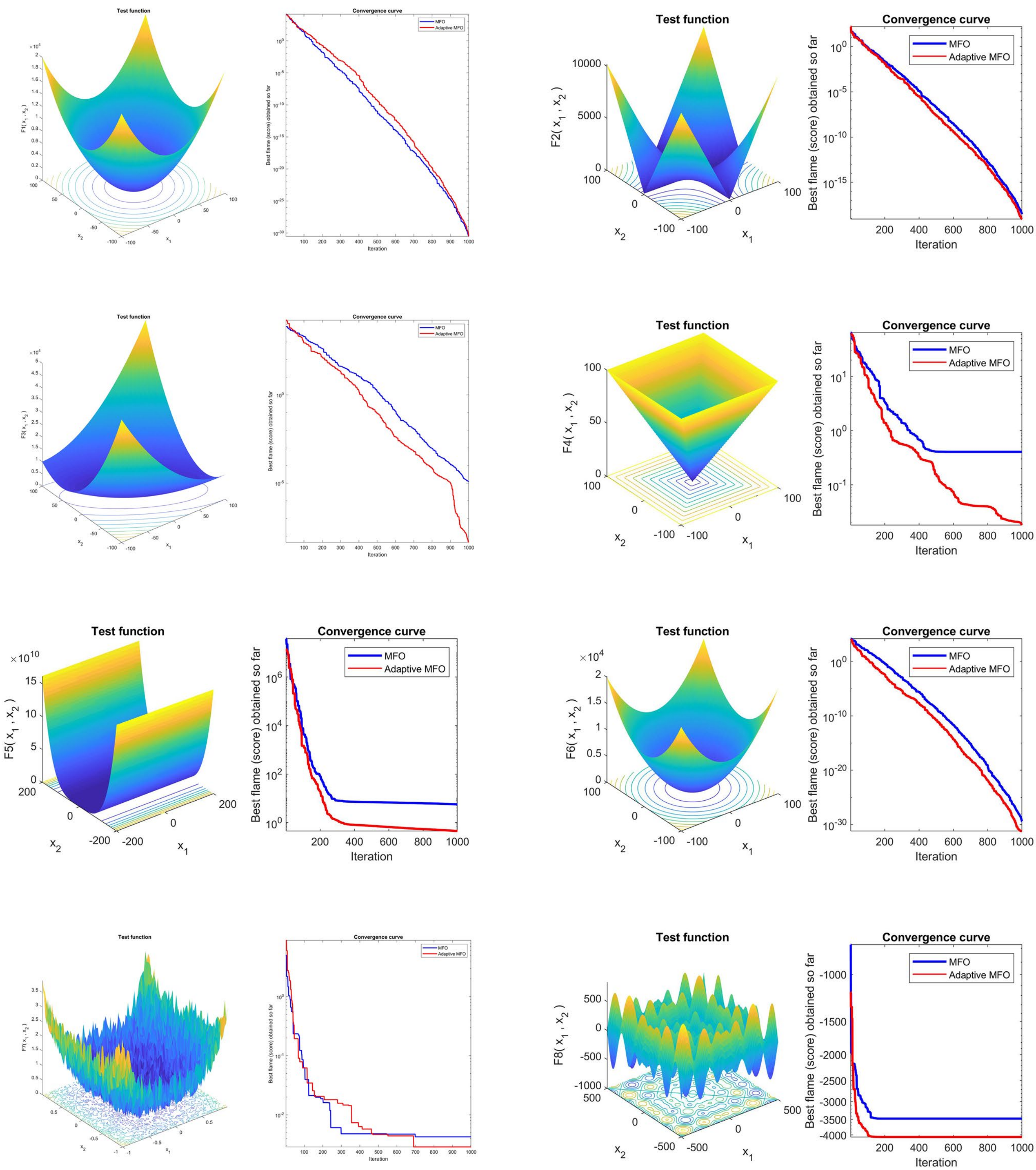

Fig. 3 Convergence curve for benchmark test functions (F1–F14)

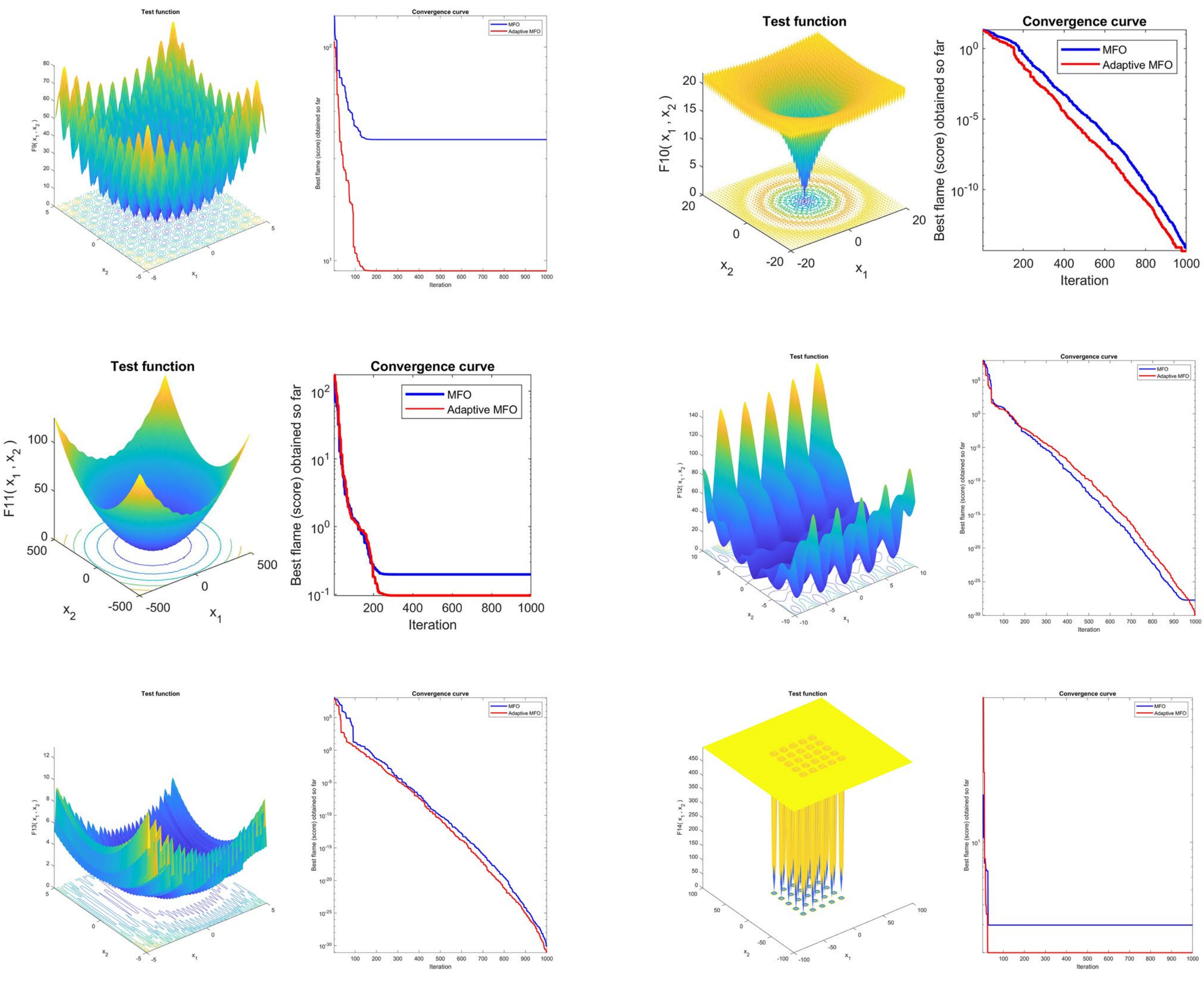

Fig. 3 (continued)

Fig. 4 Comparison between AMFO and MFO on standard benchmark functions

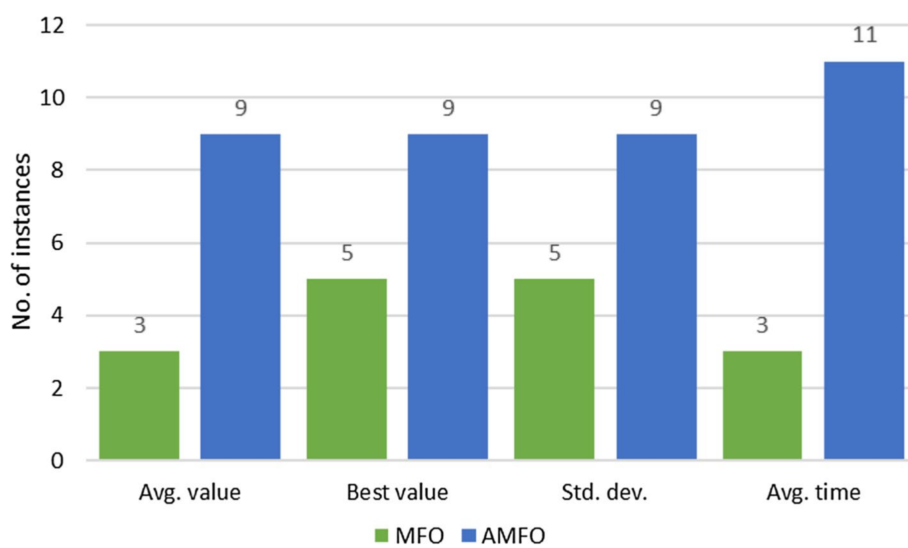

Table 3 Summary of case studies for IEEE 118-bus test systems

Test system	Case #	Objective
IEEE 118-bus test system	Case #1	Quadratic fuel cost minimization
	Case #2	Cost minimization with valve-point loading effect
	Case #3	Cost minimization with multifuel
	Case #4	Cost minimization with prohibited operating zone
	Case #5	Cost minimization by combining valve-point loading effect and multifuel
	Case #6	Cost minimization by combining valve-point loading effect and prohibited operating zone
	Case #7	Cost minimization by combining prohibited operating zone and multifuel
	Case #8	Cost minimization by combining valve-point loading effect, prohibited operating zone and multifuel
	Case #9	Emission minimization
	Case #10	Voltage stability enhancement
	Case #11	Voltage deviation minimization
	Case #12	Active power loss minimization
	Case #13	Reactive power loss minimization

Table 4 Characteristics of the IEEE 118-bus test system

System characteristics	Value	Details
Thermal units	54	
Active power demand	4242 MW	
Reactive power demand	1438 MVAR	
Buses	118	Group, IIT Power (2017)
Branches	186	Group, IIT Power (2017)
Generators	54	Buses: 1, 4, 6, 8, 10, 12, 15, 18, 19, 24, 25, 26, 27, 31, 32, 34, 36, 40, 42, 46, 49, 54, 55, 56, 59, 61, 62, 65, 66, 69, 70, 72, 73, 74, 76, 77, 80, 82, 85, 87, 89, 90, 91, 92, 99, 100, 103, 104, 105, 107, 111, 112, 113, 116
Shunts	14	Buses: 5, 34, 37, 44, 45, 46, 48, 74, 79, 82, 83, 105, 107, 110
Transformers	9	Branches: 8, 32, 36, 51, 93, 95, 102, 107, 127
Transformer tap setting	0.9 p.u. to 1.1. p.u.	
Voltage bounds	0.95 p.u. to 1.1 p.u.	
Control variables	131	–

2.2 Minimization of Quadratic Fuel Cost with Valve-Point Loadings

In this case, valve-point loading is modeled as an absolute sinusoidal function added to the cost characteristics.

$$f = \left(\sum_{i=1}^{NG} a_i P_{Gi}^2 + b_i P_{Gi} + c_i \right) + \left| d_i \sin(e_i(P_{Gi}^{\min} - P_{Gi})) \right| \quad \forall i = 1 \text{ to } NG \tag{23}$$

where a_i, b_i, c_i, d_i and e_i are the cost coefficients of ith generator. Due to valve-point loading effect, the optimum value of fuel cost increases as shown in Table 8. In this case, AMFO provides the best solution, while SCA provides the worst solution. Concerning simulation speed, IMO outperform the rest of the algorithms. Figure 6b presents the convergence trend of all algorithms.

2.3 Minimization of Quadratic Fuel Cost with Multifuel

From a practical point of view, thermal generating plants may have multifuel sources like coal, natural gas and oil. Hence, the following piecewise quadratic function expresses fuel cost function:

$$f = \sum_{i=1}^{n} (a_{ik} P_i^2 + b_{ik} P_i + c_{ik}) + \text{Penalty if } P_{ik}^{\min} \le P_i \le P_{ik}^{\max} \tag{24}$$

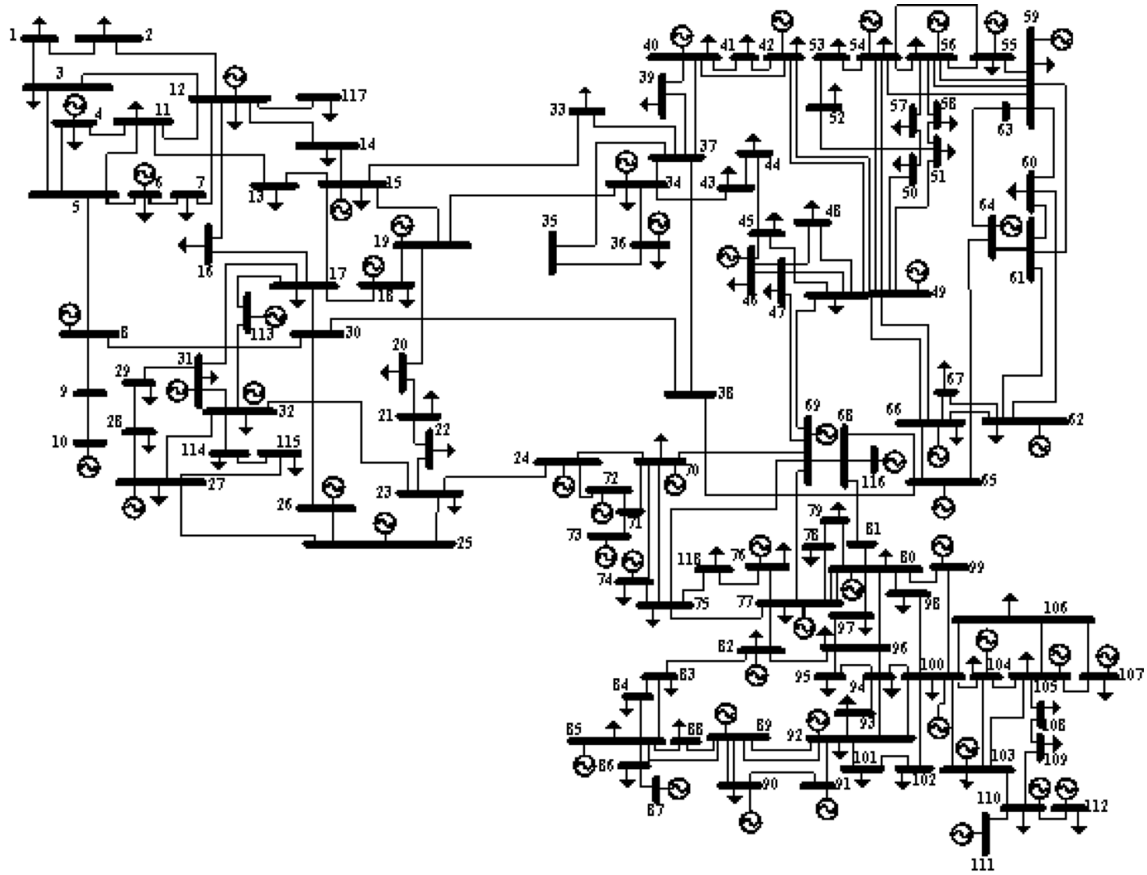

Fig. 5 Single-line diagram of IEEE 118-bus test system

where a_{ik}, b_{ik} and c_{ik} represent the cost coefficients of the ith generator for fuel type k. The optimal cost obtained by AMFO among all algorithms for this case is 64,970.59 \$/h, while SCA provides the worst solution. Minimum average simulation time and the standard deviation are offered by AMFO and GWO, respectively, while maximum average simulation time and the standard deviation are provided by GOA and DA, respectively. Figure 7a presents convergence characteristics for optimum fuel cost value with multifuel. It is evident that AMFO approaches final solution smoothly, while SCA highlights maximum tendency of stagnation throughout iterations.

2.4 Minimization of Quadratic Fuel Cost with the Prohibited Operating Zone (POZ)

In practical systems, the entire unit operating range is not forever accessible for operation. Some of the online units may have prohibited operating zones due to physical operating limitations. Units can have prohibited zones due to intensified vibrations in a shaft bearing in an operating region, faults in the machines themselves or the associated auxiliaries, such as boilers, feed pumps. The use of units in these regions leads to volatilities, leaving them incapable of holding any load for any considerable time. Hence, the avoidance of operation in these zones will improve the economic condition and performance.

The POZ is accounted for by the insertion of penalty to decrease the fitness of the fuel cost function. As presented in Table 8, AMFO provides the best solution as compared to the rest of the algorithms with smooth convergence characteristics, whereas fuel cost obtained by SCA is worst among all algorithms with stagnant convergence characteristics.

2.5 Minimization of Fuel Cost with Combined Valve-Point Effect and Multifuel

This case is a combination of case 2 and case 3. In this case, best-optimized fuel cost value obtained is 65,241.5878 \$/h which is higher than 64,636.9307 \$/h due to the introduction of valve-point loading effect. This case presents a combination of non-sinusoidal objective function superimposed on a piecewise quadratic fuel cost function. Figure 8a shows a convergence trend for this case. It is observable that AMFO converges to its final solution smoothly contrary to SCA which follows stepwise convergence behavior.

Table 5 Cost and emission coefficients of IEEE 118-bus test system

G	Bus	a	b	c	d	e	α	β	γ	σ	xci
1	1	31.67	26.2438	0.0697	0	0	6.131	5.555	5.151	0.00001	6.667
2	4	31.67	26.2438	0.0697	0	0	6.131	5.555	5.151	0.00001	6.667
3	6	31.67	26.2438	0.0697	0	0	6.131	5.555	5.151	0.00001	6.667
4	8	6.78	12.8875	0.0109	100	0.084	2.543	6.047	5.638	0.00005	3.333
5	10	6.78	12.8875	0.0109	100	0.084	6.131	5.555	5.151	0.00001	6.667
6	12	31.67	26.2438	0.0697	0	0	5.326	3.55	3.38	0.00002	2
7	15	10.15	17.82	0.0128	0	0	6.131	5.555	5.151	0.00001	6.667
8	18	31.67	26.2438	0.0697	0	0	6.131	5.555	5.151	0.00001	6.667
9	19	31.67	26.2438	0.0697	0	0	2.543	6.047	5.638	0.00005	3.333
10	24	6.78	12.8875	0.0109	100	0.084	4.091	5.554	6.49	0.00002	2.857
11	25	32.96	10.76	0.003	120	0.0077	6.131	5.555	5.151	0.00001	6.667
12	26	31.67	26.2438	0.0697	0	0	6.131	5.555	5.151	0.00001	6.667
13	27	31.67	26.2438	0.0697	0	0	5.326	3.55	3.38	0.00002	2
14	31	10.15	17.82	0.0128	0	0	6.131	5.555	5.151	0.00001	6.667
15	32	31.67	26.2438	0.0697	0	0	5.326	3.55	3.38	0.00002	2
16	34	10.15	17.82	0.0128	0	0	6.131	5.555	5.151	0.00001	6.667
17	36	31.67	26.2438	0.0697	0	0	6.131	5.555	5.151	0.00001	6.667
18	40	31.67	26.2438	0.0697	0	0	5.326	3.55	3.38	0.00002	2
19	42	10.15	17.82	0.0128	0	0	2.543	6.047	5.638	0.00005	3.333
20	46	28	12.3299	0.0024	0	0	2.543	6.047	5.638	0.00005	3.333
21	49	28	12.3299	0.0024	0	0	5.326	3.55	3.38	0.00002	2
22	54	10.15	17.82	0.0128	0	0	5.326	3.55	3.38	0.00,002	2
23	55	10.15	17.82	0.0128	0	0	4.258	5.094	4.586	0.00001	8
24	56	39	13.29	0.0044	0	0	4.258	5.094	4.586	0.00001	8
25	59	39	13.29	0.0044	0	0	5.326	3.55	3.38	0.00002	2
26	61	10.15	17.82	0.0128	0	0	4.091	5.554	6.49	0.00002	2.857
27	62	64.16	8.3391	0.0106	150	0.063	2.543	6.047	5.638	0.00005	3.333
28	65	64.16	8.3391	0.0106	150	0.063	5.326	3.55	3.38	0.00002	2
29	66	6.78	12.8875	0.0109	0	0	6.131	5.555	5.151	0.00001	6.667
30	69	74.33	15.4708	0.0459	0	0	6.131	5.555	5.151	0.00001	6.667
31	70	31.67	26.2438	0.0697	0	0	6.131	5.555	5.151	0.00001	6.667
32	72	31.67	26.2438	0.0697	0	0	5.326	3.55	3.38	0.00002	2
33	73	17.95	37.6968	0.0283	0	0	5.326	3.55	3.38	0.00002	2
34	74	10.15	17.82	0.0128	0	0	2.543	6.047	5.638	0.00005	3.333
35	76	10.15	17.82	0.0128	0	0	5.326	3.55	3.38	0.00002	2
36	77	6.78	12.8875	0.0109	105	0.081	6.131	5.555	5.151	0.00001	6.667
37	80	10.15	17.82	0.0128	0	0	4.258	5.094	4.586	0.00001	8
38	82	31.67	26.2438	0.0697	0	0	6.131	5.555	5.151	0.00001	6.667
39	85	32.96	10.76	0.003	0	0	4.258	5.094	4.586	0.00001	8
40	87	6.78	12.8875	0.0109	0	0	2.543	6.047	5.638	0.00005	3.333
41	89	17.95	37.6968	0.0283	0	0	2.543	6.047	5.638	0.00005	3.333
42	90	58.81	22.9423	0.0098	0	0	2.543	6.047	5.638	0.00005	3.333
43	91	6.78	12.8875	0.0109	200	0.042	6.131	5.555	5.151	0.00001	6.667
44	92	6.78	12.8875	0.0109	200	0.042	5.326	3.55	3.38	0.00002	2
45	99	6.78	12.8875	0.0109	200	0.042	5.326	3.55	3.38	0.00002	2
46	100	17.95	37.6968	0.0283	0	0	6.131	5.555	5.151	0.00001	6.667
47	103	10.15	17.82	0.0128	0	0	4.258	5.094	4.586	0.00001	8
48	104	10.15	17.82	0.0128	0	0	5.326	3.55	3.38	0.00002	2
49	105	17.95	37.6968	0.0283	0	0	5.326	3.55	3.38	0.00002	2
50	107	58.81	22.9423	0.0098	0	0	5.326	3.55	3.38	0.00002	2
51	111	10.15	17.82	0.0128	0	0	4.258	5.094	4.586	0.00001	8

Table 5 (continued)

G	Bus	a	b	c	d	e	α	β	γ	σ	xci
52	112	10.15	17.82	0.0128	0	0	25	100	0	0	0
53	113	10.15	17.82	0.0128	0	0	25	100	0	0	0
54	116	58.81	22.9423	0.0098	0	0	25	50	0	0	0

Table 6 Cost coefficients for multifuel

G	P_{min}	P_1	P_{max}	$c1$	$b1$	$a1$	$e1$	$f1$	$c2$	$b2$	$a2$	$e2$	$a2$
4	150	200	300	0.010875	12.8875	6.78	0	0	0.04875	12.8875	6.78	0	0
27	100	200	420	0.01059044	8.339147142	64.16	0	0	0.059044049	9.339147142	64.16	0	0
39	100	210	300	0.03	10.76	32.96	0	0	0.003	10.76	32.96	0	0

Table 7 Power generation boundaries for IEEE 118-bus system

Generator	Bus	Prohibited zones					
		Zone 1		Zone 2		Zone 3	
		Min	Max	Min	Max	Min	Max
1	7	35	50	65	85	–	–
2	10	120	145	180	190	220	235
3	30	40	50	60	70	–	–
4	34	40	50	70	90	–	–
5	35	40	50	70	90	–	–
6	47	40	60	–	–	–	–

Table 8 Best value obtained by different algorithms for case 1 to case 13

Case #	AMFO	MFO	GWO	DA	SCA	ALO	MVO	GOA	IMO
1	**64,337.1857**	64,986.4682	65,321.4822	65,149.8083	**69,682.4881**	65,643.0665	66,581.3632	65,716.5534	66,970.2663
2	**64,748.0741**	65,220.893	65,948.8033	67,196.1249	**70,339.4785**	66,367.0934	67,526.9341	67,938.2316	67,777.9746
3	**64,636.9307**	65,559.5052	66,075.4233	66,580.1423	**69,703.3689**	66,718.8294	67,228.4096	67,131.247	68,170.9496
4	**64,055.17**	64,878.1972	65,502.6685	65,321.2214	**69,256.2838**	65,944.0186	66,530.2263	65,539.8698	66,640.8124
5	**65,241.5878**	66,133.8623	66,553.8878	68,149.9077	**70,989.2011**	67,304.6818	67,717.7889	68,047.8763	68,008.5765
6	**64,537.665**	65,471.2423	66,093.6657	65,982.9886	**70,361.4918**	66,444.4729	66,876.9991	67,640.5598	67,108.2717
7	**63,549.7006**	64,515.7297	65,192.151	64,916.6604	**69,019.6882**	66,783.5684	67,465.0439	67,060.1753	68,365.842
8	**64,056.4343**	65,258.7414	65,525.8987	67,043.2587	**70,060.0148**	67,483.2107	67,684.9752	68,337.5191	68,394.2614
9	**6.7791**	6.8163	7.0316	7.1561	**8.7803**	7.2231	7.0727	7.1572	7.387
10	**0.050,204**	**0.050204**	**0.050204**	0.050205	**0.050211**	0.050205	**0.050204**	**0.050204**	0.050205
11	0.26495	**0.25664**	0.29993	0.49122	**1.0707**	0.2898	0.35458	0.40276	0.36258
12	**23.635**	32.1991	49.3499	28.4817	**121.8916**	33.9076	50.9927	38.627	30.1149
13	**−1752.6162**	−1730.547	−1419.5535	−1688.1883	**−901.4022**	−1558.9009	−1412.4968	−1441.0359	−1683.593

2.6 Minimization of the Quadratic Fuel Cost Function with Prohibited Operating Zone and Valve-Point Loading Effect

In this case, the objective function is a mixture of cases 2 and 4 resulting in non-convex and nonlinear objective function. For such a complex objective function, AMFO provides the best solution among all algorithms, while SCA provides the worst solution. Figure 8b illustrates that AMFO reaches the best solution smoothly, while the convergence curve of SCA presents the poor search of solution space.

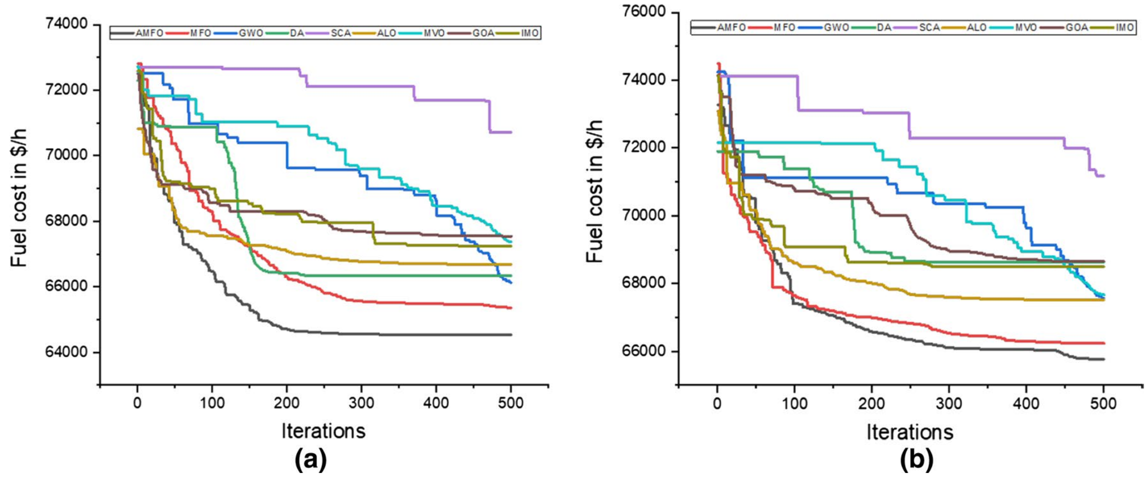

Fig. 6 Convergence curve for case 1 and case 2

Table 9 Average value obtained by different algorithms for case 1 to case 13

Case #	AMFO	MFO	GWO	DA	SCA	ALO	MVO	GOA	IMO
1	**64,985.1501**	65,825.5534	65,957.2012	67,221.283	**70,478.7446**	66,665.292	67,539.5427	66,268.3522	67,822.5193
2	**65,473.971**	66,225.0723	66,424.6769	68,654.8012	**71,239.4581**	67,348.3127	68,687.1136	69,059.0896	68,619.4705
3	**65,493.8999**	66,411.8286	66,582.5908	68,864.896	**71,503.1791**	67,840.1545	68,441.9779	68,011.4268	69,178.3949
4	**64,699.6836**	65,669.2674	65,925.8766	67,354.6531	**70,359.8849**	66,640.6049	67,877.5858	66,241.954	67,750.8756
5	**65,968.6196**	66,947.5534	67,160.4524	70,006.5745	**72,308.8061**	68,513.5415	68,859.0632	69,626.2731	69,640.6333
6	**65,104.7535**	66,311.5205	66,605.2792	68,093.2991	**71,195.2108**	67,505.6941	68,348.5475	68,756.7117	68,705.9474
7	**64,314.1969**	65,489.1629	65,597.7857	67,587.4676	**70,200.4736**	67,852.9186	68,475.5022	67,992.9606	69,145.5722
8	**64,859.6856**	65,983.6118	65,993.7649	68,636.5943	**70,914.7937**	68,564.1333	68,752.0305	69,722.6272	69,926.2274
9	**6.8176**	7.0825	7.1626	8.1141	**10.9927**	7.5724	7.1789	7.3705	7.8051
10	**0.050205**	0.050206	**0.050205**	0.050206	**0.050449**	0.050206	**0.050205**	**0.050272**	0.050,206
11	0.38385	0.39939	0.38219	0.78555	**1.283**	**0.37521**	0.42641	0.56847	0.41287
12	**25.0752**	42.9503	57.4422	38.4745	**135.7891**	41.747	63.4967	48.3841	33.9501
13	**− 1734.3727**	− 1603.0571	− 1345.8385	− 1504.9807	**− 785.4632**	− 1485.8685	− 1314.041	− 1363.9144	− 1639.4945

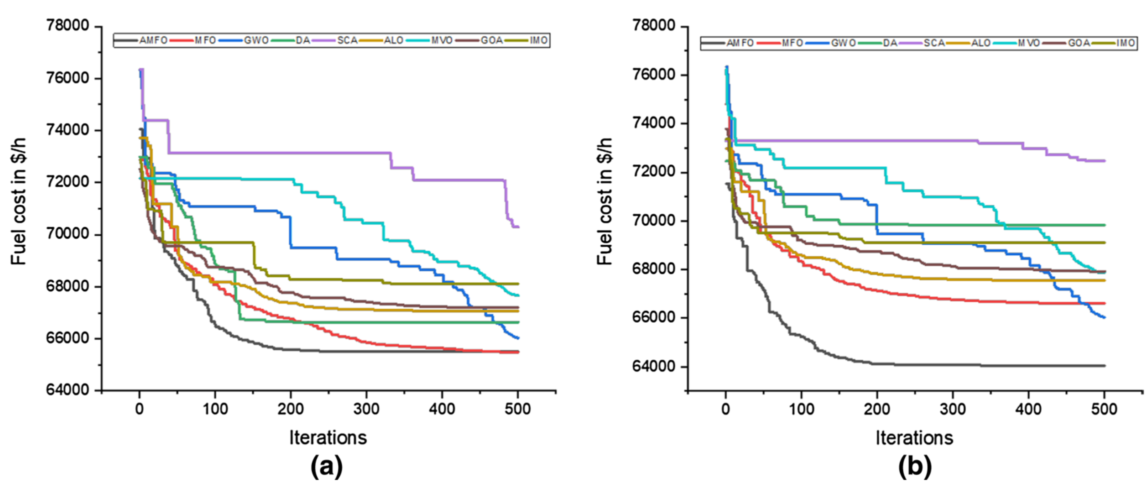

Fig. 7 Convergence curve for case 3 and case 4

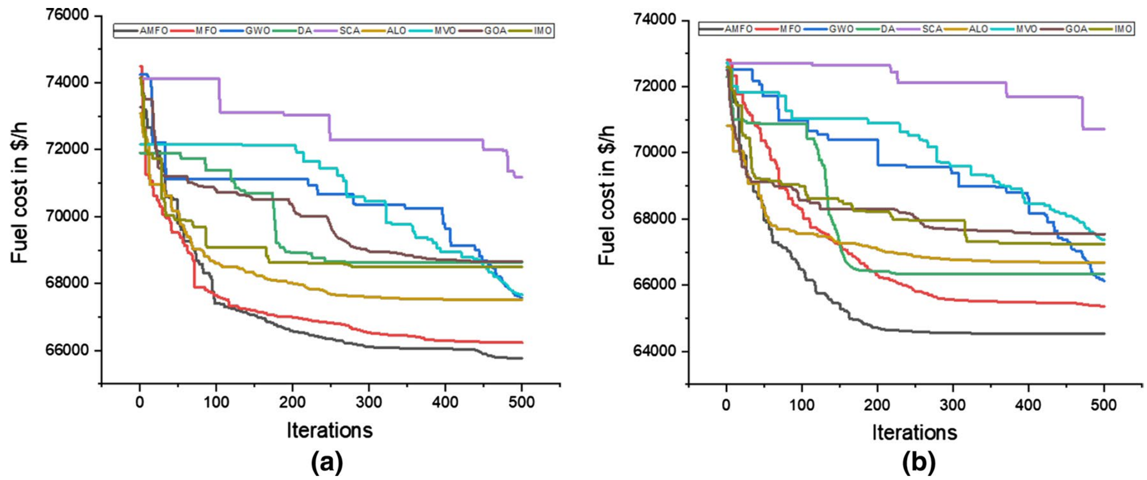

Fig. 8 Convergence curve for case 5 and case 6

Table 10 Average simulation time for case 1 to case 13

Case	AMFO	MFO	GWO	DA	SCA	ALO	MVO	GOA	IMO
1	146.8057	**104.8021**	105.8021	129.4161	139.5401	275.2698	107.1552	**376.937**	106.5594
2	140.7911	121.3224	158.1563	123.225	112.0984	173.1021	126.9266	**391.8667**	**105.7943**
3	147.2229	113.3943	105.2406	108.2078	**106.1677**	234.7927	111.4214	**363.5854**	108.3458
4	149.5484	190.8526	136.2135	126.8979	115.7224	242.9885	121.4875	**339.9089**	**110.074**
5	139.0406	135.2734	150.1833	112.6396	**106.3495**	234.1438	133.1349	**452.8599**	133.6896
6	142.9391	181.4594	163.601	111.9354	**107.0182**	168.3755	113.0099	**327.6453**	109.4656
7	153.4615	111.6823	**106.3401**	112.7724	107.3823	191.625	116.5266	**416.1521**	112.8818
8	145.8052	141.2312	115.437	123.338	**107.0438**	237.8906	138.5255	**417.4318**	136.6958
9	192.8	105.7417	**100.7813**	108.9255	103.1333	162.0828	105.2901	**322.8797**	104.7083
10	150.8573	157.3122	157.3626	171.7631	160.5243	172.7146	142.2323	**337.4839**	**130.149**
11	101.6661	**99.9286**	101.2708	104.4245	102.3318	162.1219	104.6865	**329.874**	139.4552
12	153.8375	103.4849	**101.5307**	118.8594	109.5333	168.7188	104.3443	**322.5151**	107.7422
13	144.3984	106.1172	**101.9385**	109.6984	103.4984	169.8031	104.3349	**322.6995**	106.8745

2.7 Minimization of Fuel Cost Considering Prohibited Operating Zone and Multifuel

Combining case 3 and 4 results in this case, i.e., optimal power flow with a disjoint and piecewise quadratic fuel cost function. Tables 8 and 9 present the best values and average values. For both values, AMFO and SCA provide the best and worst results, respectively. As presented in Table 10, GWO optimizes the objective function in minimum simulation time. Figure 8a presents the convergence curve of all the algorithms. AMFO and MFO present smooth convergence, while SCA presents poor convergence characteristics.

2.8 Minimization of Fuel Cost Combining Valve-Point Loading Effect, Prohibited Operating Zone and Multifuel

Combining cases 2, 3 and 4 results in this case, i.e., optimal power flow with a nonlinear, discontinuous, disjoint and piecewise quadratic fuel cost function. Observing Table 8 suggests that MFO provides the optimum fuel cost closely followed by AMFO as compared to the rest of the algorithms, while SCA provides the worst solution. GWO and AMFO prove to be best contenders regarding standard deviation and average simulation speed, while GOA and DA stand last regarding simulation speed and standard deviation, respectively. Figure 9b highlights the convergence trend of all algorithms for this case. It reflects that MFO and AMFO present smooth convergence trend, while SCA and MVO highlight stepwise convergence.

2.9 Minimization of Emission

In the present case, the objective is to lessen the emission level of pollutants. The objective function can be written as:

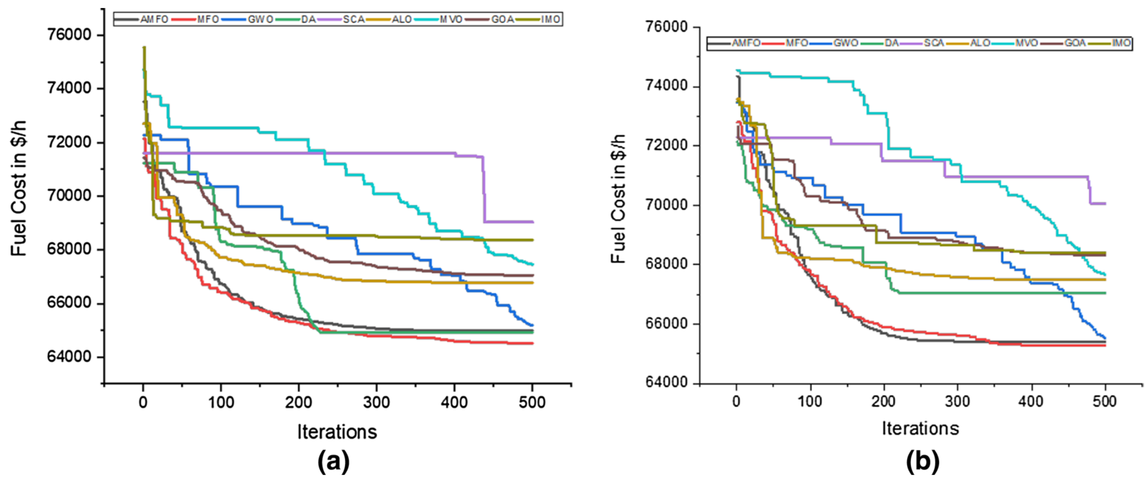

Fig. 9 Convergence curve for case 7 and case 8

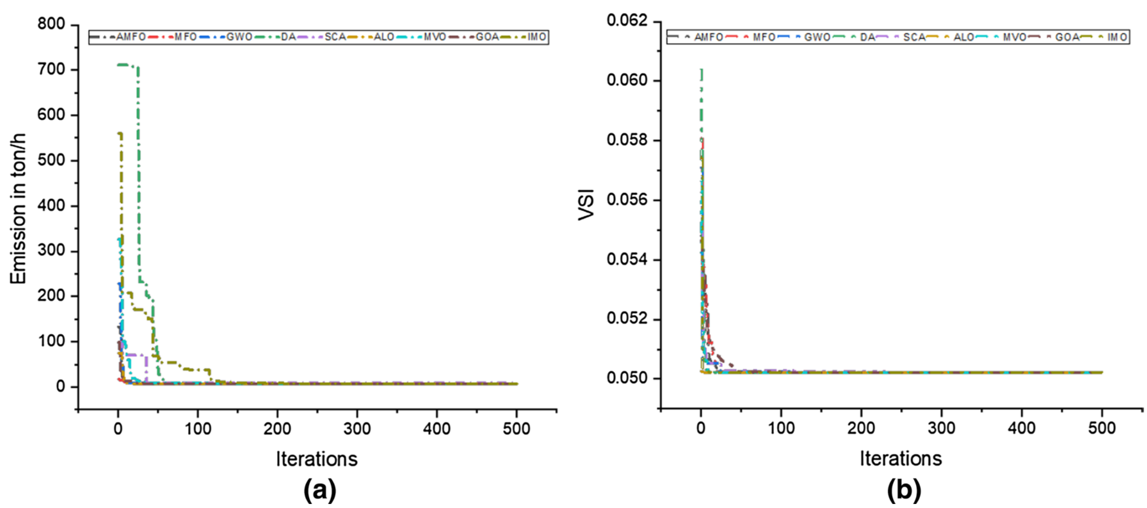

Fig. 10 Convergence curve for case 9 and case 10

$$f = \sum_{i=1}^{NG} \gamma_i P_{Gi}^2 + \beta_i P_{Gi} + \alpha_i + \zeta_i e^{(\lambda_i P_{Gi})} \text{ (ton/h)} \quad (25)$$

where γ_{i}, β_{i}, α_{i}, ς_{i} and λ_{i} are the emission coefficients of ith unit. In this case, AMFO provides the best solution of emission, i.e., 6.7791 ton/h. SCA provides the highest emission value of 8.7803 ton/h. GWO and GOA are the top and worst performers in terms of average simulation time. GOA presented the worst performance in terms of simulation speed. Figure 10a shows the sketch of convergence behavior of all the algorithms.

2.10 Voltage Stability Enhancement

The voltage stability is an essential index for verification of power system ability to preserve the voltage continually at each power system bus within a suitable level under nominal operating conditions. A disturbance, any change in system configuration, and a rise in load demand are the main reasons for the voltage instability state in the power system, which may lead to a progressive reduction in voltage. Therefore, the minimization of voltage stability indicator, called L-index (Kessel and Glavitsch 1986), is a significant objective function for power system planning and operation. The degree of voltage collapse of jth bus can be expressed, based on local indicators L_j as follows:

$$L_j = \left| 1 - \sum_{i=1}^{NG} F_{ji} \frac{V_i}{V_j} \right| \quad \forall j = 1, 2, \ldots NL \quad (26)$$

From Table 8, AMFO, MFO, GWO, MVO and GOA achieve the best solution, while SCA provides the worst solution.

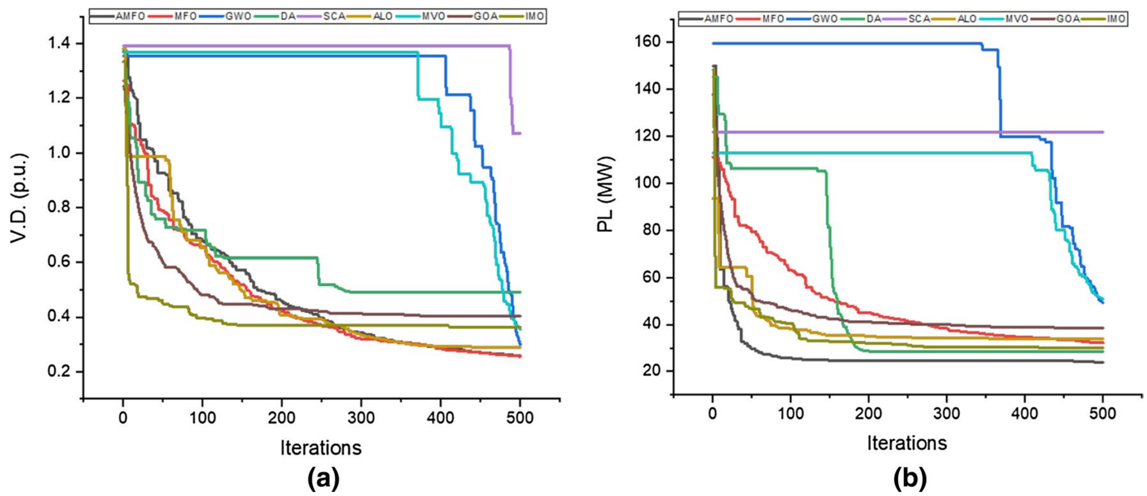

Fig. 11 Convergence curve for case 11 and case 12

IMO optimizes in minimum average simulation time, while AMFO and GWO attain best average function values. Figure 10b presents the convergence trend of all the algorithms. Most of the algorithms reach final solution quite smoothly and speedily.

2.11 Voltage Deviation Minimization

It is necessary to continuously retain load bus voltages inside stipulated deviation boundaries typically within $\pm 5\%$ of the nominal value. In this case, control variables are adjusted to curtail voltage deviation. Voltage deviation can be described as:

$$f = \sum_{i=1}^{NG} |V_i - 1.0| \tag{27}$$

From Table 8, MFO achieves the best solution closely followed by AMFO, while SCA provides the worst solution of voltage deviation minimization. ALO and MFO provide minimum average value and simulation time, respectively. Figure 11a presents convergence trend of all the algorithms suggesting that MVO, GWO and SCA have flat convergence profile for a maximum number of iterations, whereas AMFO, IMO and MFO reach the optimized value quite smoothly.

2.12 Active Power Loss Minimization

In this case, the goal is to reduce the power losses, which can be indicated as follows:

$$f = \sum_{i=1}^{NB} P_i = \sum_{i=1}^{NB} P_{Gi} - \sum_{i=1}^{NB} P_{Di} \tag{28}$$

Analyzing results obtained, it is found that AMFO performs the best regarding the best solution and average solution, while SCA provides the worst solutions. GWO and GOA are the fastest and slowest algorithms, respectively. Figure 11b presents that SCA fails to find global optimum due to flat convergence profile. AMFO, MFO, IMO and GOA present the continuous variation in objective function values throughout the iterations.

2.13 Minimization of Reactive Power Losses

Transportation of real power from the source to sink depends upon the availability of reactive power support. Voltage stability margin also hinges on reactive power support or accessibility. Based on the idea, the reactive power losses are reduced employing the following equation:

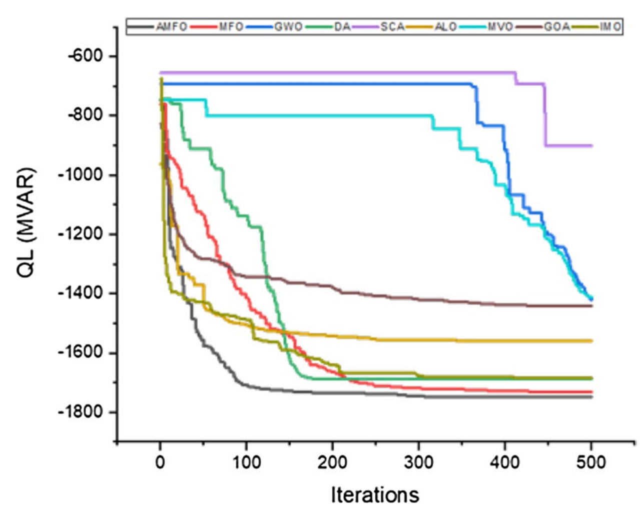

Fig. 12 Convergence curve for case 13

Table 11 Comparison of various algorithms based on a statistical test on the IEEE 118-bus test system

Position	Algorithm	Friedman	Algorithm	Quade	Algorithm	Friedman aligned
1	AMFO	1.6077	AMFO	1.3467	AMFO	240.0038
2	MFO	2.8462	MFO	2.5318	MFO	287.4577
3	GWO	3.5462	GWO	3.1504	GWO	362.9231
4	ALO	4.9269	ALO	4.8824	ALO	513.8731
5	GOA	5.6308	GOA	5.6459	GOA	666.4308
6	MVO	5.7923	DA	5.9544	DA	683.8538
7	DA	5.8538	MVO	6.1984	MVO	748.0885
8	IMO	5.9038	IMO	6.3157	IMO	765.7385
9	SCA	8.8923	SCA	8.9743	SCA	1001.13

$$f = \sum_{i=1}^{NB} Q_i = \sum_{i=1}^{NB} Q_{Gi} - \sum_{i=1}^{NB} Q_{Di} \tag{29}$$

It is noticeable that reactive power losses are negative. As derived from Table 8, AMFO achieves the best solution, while SCA lands into the worst solution. GWO and GOA provide the lowest and highest simulation time. Figure 12 presents a convergence trend of all the algorithms. AMFO exhibits smooth convergence, while SCA, GWO and MVO present partially stagnant characteristics.

2.14 Robustness Test

To assess the performance of an optimization algorithm, most of the approaches focus only on best values achieved for each case to voice their verdicts. This approach is not correct considering the stochastic nature of algorithms. In this article, the evaluation of the algorithm is based on ranking achieved by the Friedman test, Quade test and Friedman aligned test to detect whether any significant statistical differences occur. Moreover, these methods rank the algorithms from the best in performance to the poorest one.

The Friedman test (Friedman 1937) aims to decide whether there are noteworthy variances between the algorithms considered over given sets of information. The test determines the positions of the algorithms for each discrete data set, i.e., the best performing algorithm is ranked 1, the second best 2, etc.; in the case of ties, average rank is assigned. This test equates the average ranks of algorithms, and the null hypothesis asserts that all the algorithms perform equally, and hence, ranks of all algorithms should be equal.

Friedman rank test permits only intra-set comparisons. Therefore, this may become a drawback when the number of algorithms for comparison is small, as inter-set correlations may not be meaningful. In Friedman aligned test (Friedman 1940), a value of location is calculated as the average performance attained by all metaheuristics in each problem. Then, the alteration amid the performance achieved by an algorithm and the value of location is acquired. This step is replicated for each blend of metaheuristics and problems.

The Quade test (Quade 1979) in contrast to Friedman's test (Friedman test assumes that all problems are equally hard) takes into consideration the datum that a few problems are harder or that the changes recorded on the run of numerous algorithms over them are higher. Moreover, the ranking calculated for each problem relies on the changes noted in the algorithms' behavior.

All three statistical tests are done for each objective function, and the average rank for a given test system is found. Algorithms are positioned according to their average ranking from lowest to highest. Table 11 compares different algorithms based on various statistical tests. It is visible that AMFO stands first in all statistical analyses as well demonstrating its effectiveness to solve the OPF problem.

3 Conclusion

As a first contribution, an enhanced version of basic MFO, i.e., adaptive MFO, is proposed. The suggested method is employed on thirteen different benchmark functions. Performance of AMFO is equated with MFO for different attributes on these single-objective benchmark functions. As discussed, for majority attributes AMFO performed better than MFO. After validating the performance of AMFO on benchmark functions, nine different algorithms including AMFO are implemented to optimize the OPF problem. To generalize the assessment of the performance of all the algorithms, thirteen different test cases are optimized having complex and practical restraints. Evaluation is done established on the best solution, average simulation time and average solution. It can be inferred that AMFO, MFO, GWO and MVO perform well as compared to the remaining algorithms on most attributes. To validate the results, three statistical checks are performed on results obtained by each algorithm. We can conclude from the results of statistical tests that AMFO performs the best irrespective of the complexity of the objective function.

As a future work, enhanced variants of MFO and other contemporary algorithms can be proposed and applied to solve complex real-world problems. Moreover, design, development and application of multi-objective versions of enhanced algorithms can also be a promising field.

Acknowledgements The authors would like to thank Prof. Seyedali Mirjalili and Shri. Pradeep Jangir for their valuable support.

Compliance with Ethical Standards

Conflict of interest In compliance with the journal's policy and our ethical obligation as researchers, no potential conflict of interest should be reported. The authors certify that they are not involved in any organization or entity with any financial interest or non-financial interest in the subject matter discussed in this manuscript.

References

Abido MMAM (2002a) Optimal power flow using particle swarm optimization. J Electr Power Energy Syst 24(7):563–571

Abido MA (2002b) Optimal power flow using tabu search algorithm. Electr Power Components Syst 30:469–483

Abou El Ela AAA, Abido MAA, Spea SRR (2010) Optimal power flow using differential evolution algorithm. Electr Power Syst Res 80(7):878–885

AlRashidi MR, El-Hawary ME (2009) Applications of computational intelligence techniques for solving the revived optimal power flow problem. Electr Power Syst Res 79(4):694–702

Bhesdadiya RH, Trivedi IN, Jangir P, Kumar A, Jangir N, Totlani R (2017) A novel hybrid approach particle swarm optimizer with moth-flame optimizer algorithm. Springer, Singapore, pp 569–577

Bouchekara HREH (2014) Optimal power flow using black-hole-based optimization approach. Appl Soft Comput 24:879–888

Bouchekara HREHEH, Abido MA, Boucherma M (2014a) Optimal power flow using Teaching-Learning-Based Optimization technique. Electr Power Syst Res 114:49–59

Bouchekara HREH, Abido MA, Chaib AE, Mehasni R (2014b) Optimal power flow using the league championship algorithm: a case study of the Algerian power system. Energy Convers Manag 87:58–70

Bouchekara HREH, Chaib AE, Abido MA, El-Sehiemy RA (2016) Optimal power flow using an Improved Colliding Bodies Optimization algorithm. Appl Soft Comput 42:119–131

Buch H, Trivedi IN (2018) On the efficiency of metaheuristics for solving the optimal power flow. Neural Comput Appl 1–19

Buch H, Trivedi IN, Jangir P (2017) Moth flame optimization to solve optimal power flow with non-parametric statistical evaluation validation. Cogent Eng 4:1. https://doi.org/10.1080/23311916.2017.1286731

Carpentier J (1962) Contribution to the economic dispatch problem. Bull la Soc Fr des Electr 3(1):431–447

Chaib AEAAEA, Bouchekara HREHREH, Mehasni R, Abido MAA (2016) Optimal power flow with emission and non-smooth cost functions using backtracking search optimization algorithm. Int J Electr Power Energy Syst 81:64–77

Frank S, Steponavice I, Rebennack S (2012) Optimal power flow: a bibliographic survey II. Energy Syst 3(3):259–289

Friedman M (1937) The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J Am Stat Assoc

Friedman M (1940) A comparison of alternative tests of significance for the problem of m rankings. Ann Math Stat

Ghanizadeh GB, Mokhtari AJ, Abedi G, Gharehpetian M (2011) Optimal power flow based on imperialist competitive algorithm. Int Rev Electr Eng 6(4):1847–1852

Group, IIT Power One-line Diagram of IEEE 118-bus Test System. [Online]. http://motor.ece.iit.edu/data/IEEE118bus_inf/IEEE118bus_figure.pdf. Accessed 15 Jan 2017

Güçyetmez M, Çam E (2016) A new hybrid algorithm with genetic-teaching learning optimization (G-TLBO) technique for optimizing of power flow in wind-thermal power systems. Electr Eng 98(2):145–157

Javidy B, Hatamlou A, Mirjalili S (2015) Ions motion algorithm for solving optimization problems. Appl Soft Comput J 32:72–79

Kessel P, Glavitsch H (1986) Estimating the voltage stability of a power system. IEEE Trans Power Deliv 1(3):346–354

Kumar NM, Wunnava A, Sahany S, Panda R (2015) A new adaptive Cuckoo search algorithm. In: 2015 IEEE 2nd int. conf. recent trends inf. syst., no. December, pp 1–5

Lai LL, Ma JT, Yokoyama R, Zhao M (1997) Improved genetic algorithms for optimal power flow under both normal and contingent operation states. Int J Electr Power Energy Syst 19(5):287–292

Li Z, Zhou Y, Zhang S, Song J (2016) Lévy-flight moth-flame algorithm for function optimization and engineering design problems. Math Probl Eng

Mirjalili S (2015a) Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl Based Syst 89:228–249

Mirjalili S (2015b) SCA: a Sine Cosine Algorithm for solving optimization problems. Knowl Based Syst

Mirjalili S (2015c) The ant lion optimizer. Adv Eng Softw 83:80–98

Mirjalili S (2016) Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural Comput Appl 27(4):1053–1073

Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61

Mirjalili S, Mirjalili SM, Hatamlou A (2016) Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Comput Appl 27(2):495–513

Mohamed A-AAAA, Mohamed YS, El-Gaafary AAM, Hemeida AM (2017) Optimal power flow using moth swarm algorithm. Electr Power Syst Res 142:190–206

Mukherjee A, Mukherjee V (2015) Solution of optimal power flow using chaotic krill herd algorithm. Chaos Solitons Fractals

Niu M, Wan C, Xu Z (2014) A review on applications of heuristic optimization algorithms for optimal power flow in modern power systems. J Mod Power Syst Clean Energy 2(4):289–297

Ong P (2014) Adaptive cuckoo search algorithm for unconstrained optimization. Sci World J 2014:943403

Pandya KS, Joshi SK (2005) A survey of Optimal Power Flow methods. J Appl Inf Technol 4(5):450–458

Paranjothi SR, Anburaja K (2002) Optimal power flow using refined genetic algorithm. Electr Power Components Syst 30(10):1055–1063

Quade D (1979) Using weighted rankings in the analysis of complete blocks with additive block effects. J Am Stat Assoc

Roa-Sepulveda CAA, Pavez-Lazo BJJ (2001) A solution to the optimal power flow using simulated annealing. Int J Electr Power Energy Syst 25(1):47–57

Saremi S, Mirjalili S, Lewis A (2017) Grasshopper optimisation algorithm: theory and application. Adv Eng Softw 105:30–47

Sinsuphan N, Leeton U, Kulworawanichpong T (2013) Optimal power flow solution using improved harmony search method. Appl Soft Comput J 13(5):2364–2374

Soliman G, Khorshid M, Abou-El-Enien T (2016) Modified moth-flame optimization algorithms for terrorism prediction. Int J Appl Innov Eng Manag 5(7):47–59

Surender Reddy S et al (2014) Faster evolutionary algorithm based optimal power flow using incremental variables. Int J Electr Power Energy Syst

Trivedi IN, Bhoye M, Jangir P, Parmar SA, Jangir N, Kumar A (2016a) Voltage stability enhancement and voltage deviation minimization using BAT optimization algorithm. In: 2016 3rd International conference on electrical energy systems (ICEES), pp 112–116

Trivedi IN, Jangir P, Jangir N, Parmar SA, Bhoye M, Kumar A (2016b) Voltage stability enhancement and voltage deviation minimization using multi-verse optimizer algorithm. In: 2016 International conference on circuit, power and computing technologies (ICCPCT), pp 1–5

Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1(1):67–82

Yao X, Liu Y, Lin G (1999) Evolutionary programming made faster. IEEE Trans Evol Comput 3(2):82–102