*algorithms*

MDPI

*Article*

# Solving Optimal Power Flow Using New Efficient Hybrid Jellyfish Search and Moth Flame Optimization Algorithms

**Chiva Mayouf [1,2], Ahmed Salhi [2], Fanta Haidara [1], Fatima Zahra Aroua [2], Ragab A. El-Sehiemy [3,*], Djemai Naimi [2], Chouaib Aya [1] and Cheikh Sidi Ethmane Kane [1,*]**

[1] Unité de Recherche des Nouvelles Technologies des Energies et Systèmes Thermo-Fluides (nTEST), Département de Physique, Faculté des Sciences et Technologie, Université de Nouakchott, Nouakchott BP 888, Mauritania; mayoufchiva89@gmail.com (C.M.); fantise@gmail.com (F.H.); chouaib@somelec.mr (C.A.)

[2] Electrical Engineering Départment, LGEB Laboratory, University of Biskra, BP 145, Biskra 07000, Algeria; a.salhi@univ-biskra.dz (A.S.); zohra.aroua@univ-biskra.dz (F.Z.A.); d.naimi@univ-biskra.dz (D.N.)

[3] Faculty of Engineering, Kafrelsheikh University, Kafrelsheikh 33516, Egypt

* Correspondence: elsehimey@eng.kfs.edu.eg (R.A.E.-S.); ottmane6@hotmail.com (C.S.E.K.)

**Abstract:** This paper presents a new optimization technique based on the hybridization of two meta-heuristic methods, Jellyfish Search (JS) and Moth Flame Optimizer (MFO), to solve the Optimal Power Flow (OPF) problem. The JS algorithm offers good exploration capacity but lacks performance in its exploitation mechanism. To improve its efficiency, we combined it with the Moth Flame Optimizer, which has proven its ability to exploit good solutions in the search area. This hybrid algorithm combines the advantages of both algorithms. The performance and precision of the hybrid optimization approach (JS-MFO) were investigated by minimizing well-known mathematical benchmark functions and by solving the complex OPF problem. The OPF problem was solved by optimizing non-convex objective functions such as total fuel cost, total active transmission losses, total gas emission, total voltage deviation, and the voltage stability index. Two test systems, the IEEE 30-bus network and the Mauritanian RIM 27-bus transmission network, were considered for implementing the JS-MFO approach. Experimental tests of the JS, MFO, and JS-MFO algorithms on eight well-known benchmark functions, the IEEE 30-bus, and the Mauritanian RIM 27-bus system were conducted. For the IEEE 30-bus test system, the proposed hybrid approach provides a percent cost saving of 11.4028%, a percent gas emission reduction of 14.38%, and a percent loss saving of 50.60% with respect to the base case. For the RIM 27-bus system, JS-MFO achieved a loss percent saving of 50.67% and percent voltage reduction of 62.44% with reference to the base case. The simulation results using JS-MFO and obtained with the MATLAB 2009b software were compared with those of JS, MFO, and other well-known meta-heuristics cited in the literature. The comparison report proves the superiority of the JS-MFO method over JS, MFO, and other competing meta-heuristics in solving difficult OPF problems.

**Keywords:** optimization; hybrid JS-MFO; power flow; hybrid method; grid

check for updates

**Citation:** Mayouf, C.; Salhi, A.; Haidara, F.; Aroua, F.Z.; El-Sehiemy, R.A.; Naimi, D.; Aya, C.; Kane, C.S.E. Solving Optimal Power Flow Using New Efficient Hybrid Jellyfish Search and Moth Flame Optimization Algorithms. *Algorithms* **2024**, *17*, 438. https://doi.org/10.3390/a17100438

Academic Editor: Grammati Pantziou

Received: 10 August 2024
Revised: 22 September 2024
Accepted: 26 September 2024
Published: 1 October 2024

## 1. Introduction

Today, planners and operators in the field of electrical engineering are interested in solving the Optimal Power Flow (OPF) problem as an important tool to determine the optimal operating states of electrical power systems. This tool plays one of the most important roles in modern power system operation and has received a lot of interest over recent decades [1,2]. Furthermore, the problem of economic power generation can be evoked by considering the security of the power system through simultaneously minimizing the total fuel cost and the load bus voltage deviation as a mono-objective optimization problem [3].

In addition to economic and security issues, environmental safety is a major concern that has attracted the attention of many researchers in the OPF field, where the objective function to be minimized is given by the total quantity of pollutant gasses emitted from thermal generating units based on fossil fuel burning. In this case, the environmental dispatch problem is evoked by considering the complex function of gas emission [4].

One of the OPF problems cited previously requires the optimal adjustment of power system control variables, including active power from generating units, the voltage magnitude of the generating buses, the tap ratio of regulating transformers, and the generated reactive powers from VAR compensators. The active power and voltage magnitude of generating buses are continuous variables, while the tap settings of transformers and reactive power from VAR compensators are discrete control variables, which yield to a Mixed-Integer Nonlinear Programming (MINLP) optimization problem. Various classical optimization methods have been applied in the literature to solve OPF problems, such as the gradient-based method [5], linear programming, the Newton method, and the interior point method [6], among others. These methods based on gradient concepts present some drawbacks in dealing with MINLP and highly dimensional OPF problems, particularly their convergence to a local minimum failing in achieving the global optimal solution with highly nonlinear constraints and discontinuous objective functions. To overcome these drawbacks, meta-heuristic optimization methods based on intelligent soft computing and evolutionary algorithms have been developed, especially in recent decades, such as Genetic Algorithms (GAs) [7], Particle Swarm Optimization (PSO) [8], Ant Colony Optimization (ACO) [9], Artificial Bee Colony (ABC) [10], Gravitational Search Algorithm (GSA) [11], Grey Wolf Optimizer (GWO) [12], Dragonfly Algorithm (DA) [13], Teaching–Learning-based Optimization Algorithm (TLBO) [14], Firefly Algorithm (FFA) [15], Harris Hawks Optimization [16], Turbulent Flow of Water-Based Optimization (TFWO) [17], and so on. They have been successfully applied to solve MINLP-OPF problems without considering the derivability and continuity of the objective functions and constraints.

The two main phases of a meta-heuristic algorithm that determine its performance in achieving a global optimal solution are exploration and exploitation [18]. In the past decade, various strategies have been developed to enhance the effectiveness of meta-heuristic techniques by balancing these two phases, particularly when the technique shows weaknesses in one of them. The first strategy focuses on improving the basic meta-heuristic optimization algorithm by adjusting the parameters of the search equation that guide exploration, exploitation, or both. In [19], a Modified Sine Cosine Algorithm (MSCA) is introduced to address the issue of premature convergence in the basic SCA, with two key modifications highlighted. The first modification involves replacing the existing linear update mechanism in the standard SCA with a new nonlinear update mechanism. The second modification adjusts the updated design variable of each agent to include an average design variable between the current design variable of each agent and the best current design variable. In [20], the author proposes an improved Whale Optimization Algorithm (WOA) based on a Nonlinear Adaptive Weight and Golden Sine Operator (NGS-WOA). First, NGS-WOA introduces a nonlinear adaptive weight, allowing search agents to adaptively explore the search space and balance the exploration and exploitation phases. Second, the improved Golden Sine Operator is integrated into the WOA. The second strategy for improving the performance of meta-heuristics involves combining or hybridizing two meta-heuristic techniques to leverage their strengths in managing both the exploration and exploitation phases [18].

To improve the solution quality and speed up the convergence characteristics of meta-heuristics-based OPF problems, various strategies using hybrid meta-heuristic techniques have been proposed in the literature. In [21], a hybrid PSO and Pattern Search (PS) algorithm named PSO-PS is considered to improve the OPF solution by merging the advantages of both basic algorithms with the inclusion of FACTS devices. In [22], a combination of Differential Evolution and Harmony Search algorithms (DE-HS) is elaborated to improve the OPF solution by considering various scales of test systems. In [23], the authors proposed a hybrid meta-heuristic optimization approach based on the Dragonfly Algorithm (DA) and PSO to enhance the search capability to achieve the best global solution quality for solving the single- and multi-objective OPF problems. As reported in [24], a new hybrid approach using the Firefly and PSO algorithms (HFPSO) has been applied to the OPF solution to enhance the exploration and exploitation mechanisms with fast convergence to deal with various complexities of the OPF problem. The recently published work in [25] presents a novel hybrid optimization technique combining ABC and Salp Swarm Algorithms (SSAs) named ABC-SSA to solve the Optimal Reactive Power Dispatch (ORPD) as a sub-problem of OPF. The efficiency of ABC-SSA in balancing between exploration and exploitation mechanisms has been proven for the large-scale test system IEEE 30-bus. The authors of [26] developed a new hybrid fruit fly-based artificial bee colony (HFABC) algorithm to solve single- and multi-objective OPF problems using a fuzzy decision-based mechanism strategy to obtain the best compromise solution from the Pareto front. A Firefly and JAYA Algorithm based on the hybrid evolutionary technique (HFAJAYA) is suggested in [27] to solve the OPF problem by efficiently considering renewable energy sources. Various scenarios in the standard IEEE 30-bus test system network have been implemented to validate the simulation results of the HFAJAYA.

The Jellyfish Search optimizer (JS) algorithm is a modern meta-heuristic optimization technique recently developed by Jui-Sheng Chou and Dinh-Nhat Truong [28] in 2020; it is inspired by the behavior of jellyfish when looking for food in the ocean. The JSO reaffirms its effectiveness by using a set of unimodal and multimodal benchmark functions and various engineering problems. In [29], the JS optimizer algorithm was applied to solve the economic load dispatch (ELD) in the electric power system.

In order to obtain the best solution quality for the OPF problem, this paper proposes a new hybrid approach based on the JS and Moth Flame Optimization (MFO) algorithms. The MFO algorithm was developed by Mirjalili in 2015 [30] and is inspired by the flying behavior of moths with respect to the moon, as they can become trapped in spiral paths around artificial lights. Later, a spiral function was proposed to enhance the MFO algorithm's performance, as in [31]. The JS optimizer shows an adequate search in the exploration stage, but it suffers during the exploitation phase because it lacks a proficient exploitation operator of the best solutions in the search space. Inversely, the MFO algorithm, by imitating the flying spiral path performed by moths around flame lights, can be useful in the exploitation phase by exploiting the best solutions; nevertheless, it can become trapped in local optima. The main contribution issues can be summarized as follows:

- Providing a hybrid JS-MFO algorithm to associate the advantages of both the JS optimizer in exploration and MFO in exploitation.
- The effectiveness of the proposed hybrid approach has been investigated by solving various cases related to the OPF problem.
- Applications were carried out on two test systems, the IEEE 30-bus and the Mauritanian RIM 27-bus electric power systems, to verify the JS-MFO's performance.

This is an assessment study based on comparing our simulation results with those of other meta-heuristic methods reported in the literature.

The rest of the structure of this work is ordered as follows: Section 2 presents the formulation of the OPF problem. The mathematical representation of the proposed optimization problem is given in Section 3. Section 4 provides the simulation results and discussions. Finally, a brief conclusion is given in Section 5.

## 2. Optimal Power Flow Problem

### A. Formulation of OPF problem

The OPF problem aims to determine the optimal adjustments of the control equipment by assessing their optimal control variables in the power system network to obtain an optimized objective function, while simultaneously guaranteeing various operating restrictions. These restrictions are given by power flow equations as equality constraints and the functional limits of control equipment are given as inequality constraints. The most recognized objective function to be minimized in the OPF area is that of the total generation cost, with various complexities, including the quadratic, non-quadratic, and piecewise quadratic of the fuel cost function besides the valve point effects, affecting the fuel cost curves of the generating units [2,5,6]. In general, the conventional OPF problem can be mathematically formulated as follows:

$$\begin{cases} \min f(x,u) \\ subject\ to\ : g(x,u) \ = \ 0 \\ h(x,u) \leq 0 \end{cases} \tag{1}$$

where $f$ is the objective function to be optimized, $g$ is the set of equality constraints represented by the nonlinear power flow equations, $h$ is the set of inequality constraints reflecting the operating limits of control equipment, $u$ is the vector of control variables, and $x$ is the vector of state variables. The vector $u$ can be expressed as

$$u \ = \ \left[P_{G2} \ldots . P_{GN_g},\ V_{G1} \ldots . V_{GNg},\ T_1 \ldots . T_{Nt},\ Q_{C1} \ldots . Q_{CNc}\right]^T \tag{2}$$

Vector $x$ is given as follows:

$$x \ = \ \left[P_{G1},\ V_{L1} \ldots . V_{LNpq},\ \ Q_{G1} \ldots . Q_{GN_g},\ S_{L1} \ldots . S_{LNl}\right]^T \tag{3}$$

The equality constraints represent the balance equations of the active and reactive powers assigned to each bus, and the expression of $g(x,u)$ can be formulated in terms of control and state variables as follows:

$$\begin{cases} P_{Gi} - P_{Di} - V_i \sum_{j\,=\,1}^{N_B} V_j \left(G_{ij} cos\delta_{ij} + B_{ij} sin\delta_{ij}\right) \ = \ 0 \\ Q_{Gi} - Q_{Di} - V_i \sum_{j\,=\,1}^{N_B} V_j \left(G_{ij} sin\delta_{ij} - B_{ij} cos\delta_{ij}\right) \ = \ 0 \end{cases} \tag{4}$$

The inequality constraints expressed as $h(x,u)$ are as follows:

Limits of Generators:

The capability constraints for each generation unit are presented for limiting the active and reactive power outputs within the operational limits as follows:

$$\begin{cases} P_{Gi,min} \leq P_{Gi} \leq P_{Gi,max} \\ Q_{Gi,min} \leq Q_{Gi} \leq Q_{Gi,max} \end{cases} \quad i \ = \ 1, \ldots . N_g \tag{5}$$

Similar to the power capability constraints, the voltage at the generator terminals is kept within the minimum and maximum limits as follows:

$$V_{Gi,min} \leq V_{Gi} \leq V_{Gi,max} \ \ i \ = \ 1, \ldots . N_g \tag{6}$$

Limits of tap setting transformers:

In Equation (7), the transformer tap settings are kept between the accepted boundaries that are able to control the voltage at the transformer terminals as follows:

$$T_{i,min} \leq T_i \leq T_{i,max} \ \ i \ = \ 1, \ldots . N_t \tag{7}$$

Limits of reactive power compensators:

Reactive power compensators are bounded in Equation (8) as follows:

$$Q_{Ci,min} \leq Q_{Ci} \leq Q_{Ci,max} \ \ i \ = \ 1, \ldots . N_C \tag{8}$$

Limits of voltage magnitude for load buses

$$V_{Li,min} \leq V_{Li} \leq V_{Li,max} \ \ i \ = \ 1, \ldots . N_{PQ} \tag{9}$$

Power flow limits of transmission lines:

$$S_{Li,min} \leq S_{Li} \leq S_{Li,max} \ \ i \ = \ 1, \ldots . N_l \tag{10}$$

To handle an inequality constraint of a state variable, a penalty function is introduced in the augmented objective function as follows [6]:

$$F_{aug} = f(x,u) + \sum_{k=1}^{Ns} \lambda_{pk} \left(x_k - x_k^{lim}\right)^2 \quad (11)$$

where $x_k$ is the variable of the $k$-th violation of the state, $x^{lim}{}_k$ is the limit of the variable of the $k$-th violation of the state, $f(x,u)$ is the objective function, and $\lambda_{Pk}$ is the penalty factor for the penalty function of the variable of the $k$-th violation of the state.

**B. Objective functions:**

The mono-objective optimization problem is considered for eight cases of objective function $f(x,u)$ in Equation (1):

Case 1: minimization of total fuel cost

The total cost of electrical power generation is expressed as an objective function (quadratic function) [9] by:

$$f = F_C = \sum_{i=1}^{Ng} F_i(P_{Gi}) = \sum_{i=1}^{Ng} \left(a_i + b_i P_{Gi} + c_i P_{Gi}^2\right) \quad (12)$$

where $a_i$, $bi$, and $c_i$ are the cost coefficients of the $i$-$th$ generating unit.

Case 2: minimizing total gas emissions

The gasses emitted by each generating unit can be expressed as a combination of the quadratic and exponential functions of the generated active power [10], and the total amount of gas emissions produced by all generating units:

$$f = F_{Em} = \sum_{i=1}^{Ng} \left(\alpha_i + \beta_i P_{Gi} + y_i P_{Gi}^2 + \delta_i exp(\varepsilon_i P_{Gi})\right) \ \text{(ton/h)} \quad (13)$$

where $F_{Em}$ is the total gas emissions (ton/h) and $\alpha_i$, $\beta_i$, $\gamma_i$, $\delta_i$, and $\varepsilon_i$ are the emission coefficients of the $i$-th unit.

Case 3: minimization of total fuel cost considering the valve point effect

A sine component is added to the expression of the objective function in (12) to consider the effect of the valve point to evaluate the total fuel cost as follows [11]:

$$f = F_C = \sum_{i=1}^{Ng} F_i(P_{Gi}) = \sum_{i=1}^{Ng} \left(a_i + b_i P_{Gi} + C_i P_{Gi}^2 + f_{vpi}\right) \quad (14)$$

where $f_{vpi} = |\, d_i \cdot \sin\,[e_i(P_{Gi,min} - P_{Gi})]\,|$, while $e_i$ and $d_i$ are the fuel cost coefficients of the since component.

Case 4: minimization of total active losses

The total active transmission losses for the power system can be expressed as an objective function:

$$f = F_{Plosses} = \sum_{K=1}^{N_l} g_k \left[V_i^2 + V_j^2 - 2V_i V_j \cos(\delta_i - \delta_j)\right] \quad (15)$$

where $g_k$ is the conductance of the $k$-th transmission line between buses $i$ and $j$.

Case 5: minimization of the total load bus voltage deviation

This optimization case is devoted to the minimization of total voltage magnitude deviation related to load buses with the goal of enhancing power system security; the total voltage deviation function for all load buses is given by:

$$f(x,u) = F_{VD} = \sum_{k=1}^{N_{PQ}} |V_{Lk} - 1| \quad (16)$$

Case 6: improvement of voltage stability index

The stability of a power system is its capacity to maintain all bus voltages within normal limits. The power system reaches a voltage instability state due to severe perturbations, rising electric load, or network configuration changes, which occasions a progressive and unexpected voltage drop. The improvement of the system voltage stability for each bus is a fundamental measure to ensure a safe and stable power system using voltage stability

(L-index). Its value varies between 0 and 1 (0 corresponds to a zero-load situation, while 1 means a voltage drop). If the power system has $N_{PQ}$ load buses (PQ buses) and $N_{PV}$ generator buses (PV buses), the L-index is evaluated by [32,33]:

$$L_j = \left|1 - \sum_{i=1}^{N_{PV}} F_{ji} \frac{V_{Gi}}{V_{Lj}}\right| \quad j = 1, 2, \ldots, N_{PQ} \tag{17}$$

where $V_{Gi}$ is the complex voltage of the generation bus $i$ and $V_{Lj}$ is the complex voltage of the load bus $j$.

$$F_{ji} = -[Y_{LL}]^{-1} \cdot [Y_{LG}] \tag{18}$$

where the sub-matrices $Y_{LL}$ and $Y_{LG}$ have been obtained from the $\mathrm{Y_{BUS}}$ matrix after rearranging nodal current injections with respect to nodal voltages, as defined as:

$$\begin{bmatrix} I_L \\ I_G \end{bmatrix} = \begin{bmatrix} Y_{LL} Y_{LG} \\ Y_{GL} Y_{GG} \end{bmatrix} \begin{bmatrix} V_L \\ V_G \end{bmatrix} \tag{19}$$

The maximum value $L_{max}$ of $L_j$ among all load buses is a system stability indicator that can be considered as an objective function presented in the following expression:

$$F_{Lmax} = \max(L_j) \quad where \quad j = 1, 2, \ldots \ldots, N_{PQ} \tag{20}$$

Case 7: quadratic total fuel cost function and load bus voltage deviation

Minimizing both the total cost of fuel in (6) and the total voltage deviation in (14) to achieve the economical and secure state of the power system, these two functions are introduced in one objective function, as stated in the following expression:

$$f = F_{CVD} = \left(\sum_{i=1}^{N_g} a_i + b_i P_{Gi} + c_i P_{Gi}^2\right) + Y_{VD} \times F_{TVD} \tag{21}$$

where $Y_{VD}$ is a weighting factor, which is selected as 100.

Case 8: quadratic total fuel cost function and voltage stability index

In order to reduce the total fuel cost and improve the voltage stability of the system simultaneously, the two objective functions in (6) and (13) are implemented into a single-objective function, formulated as follows:

$$f = F_{Lmax} = \left(\sum_{i=1}^{N_g} a_i + b_i P_{Gi} + c_i P_{Gi}^2\right) + Y_L \times F_{Lmax} \tag{22}$$

The selected value of the weighting factor $Y_L$ is 6000.

## 3. Mathematical Representation of the Optimization Technique

### A. Jellyfish Optimization Algorithm:

The main inspiration of the JS algorithm is the search behavior of jellyfish for food sources in the ocean, considering their intelligent movement behavior when detecting the best location of aliments [28]. Three rules are adopted in JS modeling:

- Two types of jellyfish movement can be handled by the time control mechanism: moving in the jellyfish swarm or following the ocean current.
- When searching for food sources in the ocean, jellyfish move in an attractive manner towards the best locations of nutrients.
- The quantity of food found by jellyfish and its location correspond to the objective function of the problem and the corresponding solution, respectively.

(a) Ocean current

The ocean current includes a considerable amount of food sources, and jellyfish are drawn to its flow. In such a case, the jellyfish update their positions based on the following equation:

$$X_i(t+1) = X_i(t) + rand(0,1)\mu(X^* - \beta \times rand(0,1)),\ \mathrm{i} = 1,\ 2,\ 3, \mathrm{N_{pop}} \tag{23}$$

where $X^*$: the current best global position of jellyfish in the swarm, $rand(0,1)$: the uniform random number in the interval [0, 1], $\mu$: the mean location of all jellyfish, and $\beta$: the distribution coefficient.

The determination of $\beta$ and the ocean current direction are given in [28], with $\beta = 3$ referring to numerical tests. $X_i(t)$ and $X_i(t+1)$ are the jellyfish positions at the iterations t and (t + 1), respectively, and $N_{pop}$: the number of jellyfish.

(b) Jellyfish swarm

In swarms, jellyfish show passive (type A) and active (type B) motions. In the beginning when the swarm is created, jellyfish exhibit the passive motion type (A). Over time, they increasingly exhibit a type B motion. Type A motion is the motion of jellyfish around their own locations, and the consistent updated location of each jellyfish is given by:

$$X_i(t+1) = X_i(t) + \gamma \times rand(0,1) \times (U_b - L_b) \tag{24}$$

In order to simulate type B motion, a jellyfish (j) is selected randomly, other than the current jellyfish (i), and a distance vector is formed between jellyfish (i) and jellyfish (j) to establish the step and the direction of the movement. When the quantity of food at the location of the selected jellyfish (j) exceeds that at the location of the jellyfish (i) of interest, the latter moves toward the former; it moves directly away from it if the quantity of food available to the selected jellyfish (j) is lower than that available to the jellyfish of interest (i). So, each jellyfish moves toward a better direction to find the food in a swarm [28]. This behavior can be simulated by the following equations:

$$\overrightarrow{Step} = X_i(t+1) - X_i \tag{25}$$

where

$$\overrightarrow{Step} = rand\ (0,\ 1) \times \overrightarrow{Direction} \tag{26}$$

$$\overrightarrow{Direction} = \begin{cases} \mathrm{X}_j(\mathrm{t})\text{–}\mathrm{X}_i(\mathrm{t}) & if\ f(X_i) \geq f\left(X_j\right) \\ X_i(t) - X_j(t) & if\ f(X_i) < f\ \left(X_j\right) \end{cases} \tag{27}$$

Here, $f$ is an objective function of location $X$, and hence the active motions (type B) are accomplished using the following expression:

$$X_i(t+1) = X_i(t) + \overrightarrow{Step} \tag{28}$$

A control parameter, time control mechanism, is employed to switch between the passive and active types of motion. It can also be used to manage the movement of jellyfish in the direction of the ocean current.

(c) Time control mechanism

At each level of changes in wind or temperature, the jellyfish change their positions to another ocean current and a new jellyfish swarm is created. Inside this swarm, the jellyfish favor the passive motion type A in the beginning of the search phase. Over time, the jellyfish begin to prefer active motion (type B). In order to simulate this behavior, a time control mechanism is introduced by regulating the movement, switching between jellyfish moving in a swarm and following the ocean current. This time control mechanism is modeled as a time control function, which randomly fluctuates over the time between 0 and 1. Equation (29) formulates the time control function as below:

$$C(t) = \left[\left(1 - \frac{t}{MaxIt}\right) \times (2 \times \mathrm{rand}(0,\ 1) - 1)\right] \tag{29}$$

where *MaxIt* is the maximal number of iterations and *rand* is a random number in the interval [0, 1].

(d) Initialization of population

The initial population of jellyfish is generated randomly. To obtain more diversification of the population, a logistic map is developed, which provides more diverse initial populations than random selection and provides a lower probability of premature convergence [28]. Consequently, the logistic map can be used in the initial population based on the following:

$$X_{i+1} \ = \ \eta \times X_i \times (1 - X_i) \ with \ 0 \leq X_o \leq 1 \tag{30}$$

where $X_i$ is the chaotic logistic value of the location of the $i$-th jellyfish, $X_o$ is used to generate the initial population of jellyfish, and the parameter $\eta$ is set to 4.

(e) Restriction of boundaries

During the swarm movement, jellyfish can move outside of the boundaries of the search space; in this case, the location is updated by returning to the opposite bound, referring to Equation (31):

$$\begin{cases} X'_{i,d} & = \ (X_{i,d} \quad - U_{b,d}) + L_b(\mathrm{d}) \quad if \ X_{i,d} > U_{b,d} \\ X'_{i,d} & = \ (X_{i,d} \quad - L_{b,d}) + U_b(\mathrm{d}) \quad if \ X_{i,d} < L_{b,d} \end{cases} \tag{31}$$

where $X_{i,d}$ is the location of the $i$-th jellyfish in the $d$-th dimension; $X'_{i,d}$ is the updated location after checking the boundary constraints; and $U_{b,d}$ and $X_{b,d}$ are the upper and lower bounds of the $d$-th dimension in search spaces, respectively.

**B. Moth Flame Optimization Algorithm**

As insects, moths share important characteristics with butterflies. The navigation strategy used by moths is a particularly intriguing aspect of their behavior and deserves closer examination. To cover great distances in a straight line, they fly at a fixed angle relative to the moon. This efficient technique is known as transverse orientation [30]. The distance from the light source substantially affects the efficacy of transverse orientation. When the light source is close, the moth spirals around it. This spiral flight path guides the moth to light [31]. The Moth Flame Optimization method was introduced by Mirjalili [30] based on observed behaviors and mathematical models. This method employs moths as potential solutions and their locations are presented as a decision variable vector, allowing each moth to move freely within the problem's solution space.

$$MO \ = \begin{bmatrix} mo_{1,1} mo_{1,2} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots .. mo_{1,D-1} mo_{1,D} \\ mo_{2,1} mo_{2,2} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots .... mo_{2,D-1} mo_{2,D} \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots .. \\ mo_{N_{POP}-1,1} mo_{N_{POP}-1,2} \ldots\ldots\ldots .. mo_{N_{POP}-1,D-1} mo_{N_{POP}-1,D} \\ mo_{N_{POP},1} mo_{N_{POP},2} \ldots\ldots\ldots\ldots\ldots .. mo_{N_{POP},D-1} mo_{N_{POP},D} \end{bmatrix} \tag{32}$$

where $N_{POP}$ is the initial population size of moths and $D$ denotes the number of decision variables. The flame matrix is another fundamental element of the MFO algorithm. Given that each moth exhibits a tendency to navigate in close proximity to its individual flame, it may be inferred that the sizes of the flame matrix are the same as those of the moth's matrix.

$$Fl \ = \begin{bmatrix} Fl_{1,1} Fl_{1,2} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots .. Fl_{1,D-1} Fl_{1,D} \\ Fl_{2,1} Fl_{2,2} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots .... Fl_{2,D-1} Fl_{2,D} \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots .. \\ Fl_{N_{POP}-1,1} Fl_{N_{POP}-1,2} \ldots\ldots\ldots .. Fl_{N_{POP}-1,D-1} Fl_{N_{POP}-1,D} \\ Fl_{N_{POP},1} Fl_{N_{POP},2} \ldots\ldots\ldots\ldots\ldots .. Fl_{N_{POP},D-1} Fl_{N_{POP},D} \end{bmatrix} \tag{33}$$

The distinction between a moth and a flame is based on their respective roles within a problem-solving context. The moth actively engages in a search process, using its flight capabilities to explore various options in order to identify more optimal solutions. In contrast, the flame represents the most optimal solution that has been discovered so far by the moth. To illustrate the helical flight trajectory of moths in proximity to a certain light source, a logarithmic spiral function is used [31].

$$\mathrm{MO}_{\mathrm{i}}^{x+1} = |\mathrm{MO}_{\mathrm{i}}^{x} - Fl_i|\cdot e^{bv}\cdot\cos(2\pi v) + Fl_i \tag{34}$$

where $v$ represents a uniformly distributed random variable ranging from −1 to 1. This parameter quantifies the proximity of the moth's subsequent location to the associated flame. To improve the efficiency of exploring the solution space during the first iterations and exploiting the solution space during the final iterations, an adaptive approach is suggested to decrease the parameter t throughout the iterations [31].

$$a = -1 + t\left(\frac{-1}{MaxIt}\right) \tag{35}$$

$$v = (a-1)\times rand + 1 \tag{36}$$

The constant, $a$, exhibits a linear reduction from −1 to −2 during the iterative process. This well-constructed temporal change ensures both exploration and exploitation in the MFO algorithm.

**C. Hybrid Jellyfish Search–Moth Flame Optimizer (JS-MFO)**

In the artificial JS optimizer, the movement toward an ocean current represents the exploration phase, while the movement within a jellyfish swarm depicts the exploitation phase. A time control mechanism alternates between these two processes [28]. For passive motion (type A) within the jellyfish swarm, each jellyfish updates its position (new solution) relative to its current location based on (24), exploiting the current solution without focusing on the best solutions found nearby. However, exploiting the best solutions in the vicinity of the current solution can help to discover more promising solutions. For active motion (type B), referring to the initial position of jellyfish (i) (current solution (i)), a jellyfish (j) is randomly selected to determine the step size and direction of movement. This updates the position of jellyfish (i) (solution (i)) using (28). This random selection of jellyfish (j) neglects the focus on the best positions (i.e., best solutions corresponding to the highest food availability), causing the algorithm to explore newly discovered solutions randomly without exploiting the best solutions or the best global solution found so far, potentially limiting its ability to find promising solutions. These two shortcomings of passive (A) and active (B) motions reduce the efficiency of the JS algorithm in the exploitation phase.

On the other hand, and based on the literature review, the MFO algorithm signals a useful exploitation capability by carrying out a good search in the proximity of the best solutions. This performance is achieved by using a spiral path around each exploited solution. However, the MFO algorithm presents a drawback; it is unable to share the best global solution obtained so far between agents during the search stage. These characteristics lead to a powerful exploitation mechanism and weak exploration process of the MFO algorithm.

Achieving a new efficient optimization approach is the reason behind our hybridization of the JS and MFO algorithms, which benefits from both advantages of the two algorithms, as shown in Figure 1. In this work, the exploitation capability of the basic JS algorithm is improved by introducing the spiral movement of MFO around the best solutions during the search process. In the early iterations of JS and just after the swarm is formed, the search space is not sufficiently explored, so in this case we try to exploit the good solutions found so far (in order to obtain promising solutions) in the first half-course of iterations (t varies from 0 to MaxIt/2); the elite solutions (which represent 30% of the sorted population of solutions, from the best to the worst) are selected to be exploited, and hence they may be imposed on the spiral movement using (37) and (38). After the first half-course of iterations, the search space is relatively well explored, so in the second half-course of iterations (t changes from MaxIt/2 to MaxIt) we concentrate on the exploitation of the best global solution found so far, which must be imposed on the spiral movement using (39) and (40).

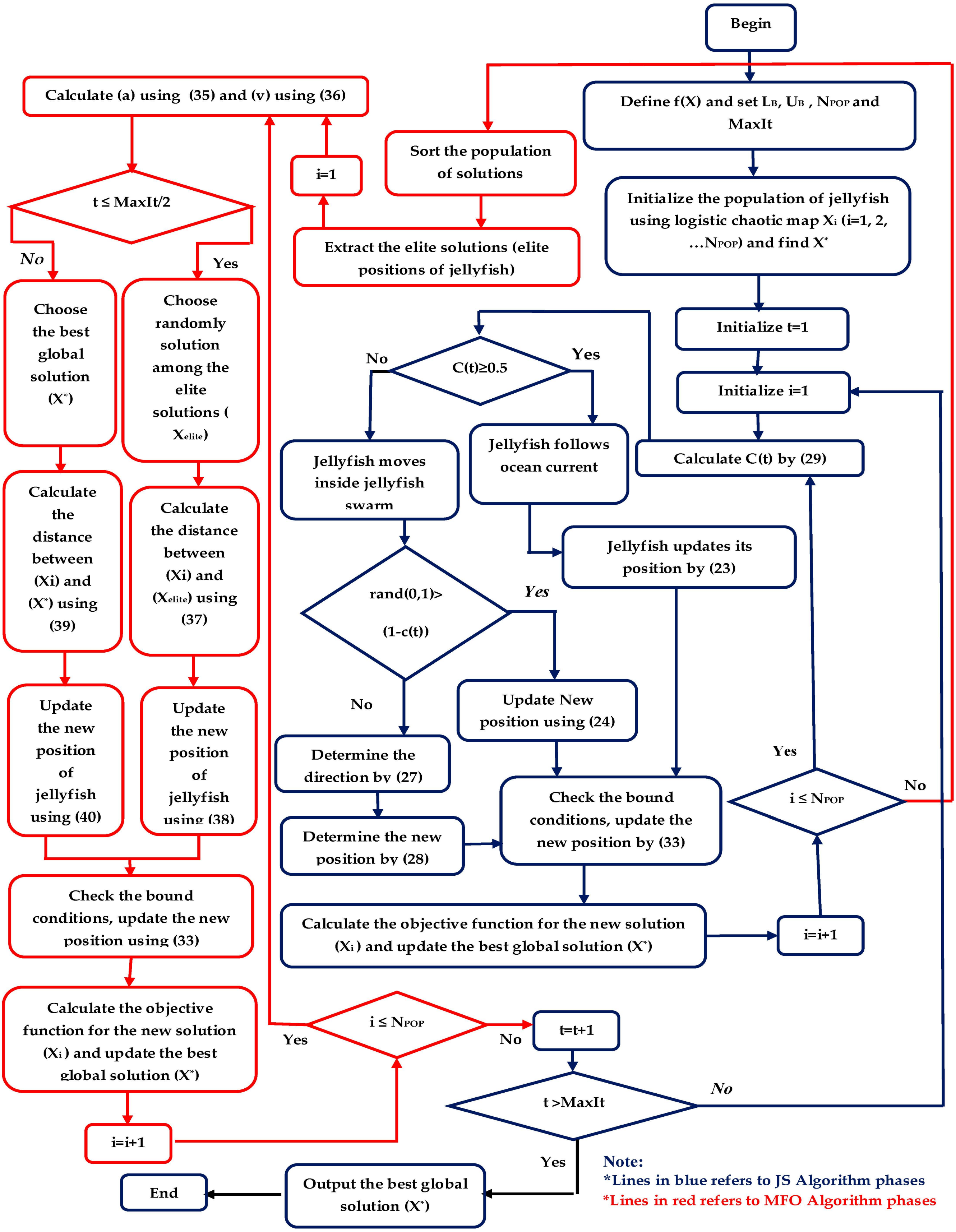
Begin
Define f(X) and set LB, UB , NPOP and MaxIt
Initialize the population of jellyfish using logistic chaotic map Xi (i=1, 2, …NPOP) and find X*
Initialize t=1
Initialize i=1
Calculate C(t) by (29)
C(t)≥0.5
Yes
No
Jellyfish follows ocean current
Jellyfish updates its position by (23)
Jellyfish moves inside jellyfish swarm
rand(0,1)> (1-c(t))
Yes
No
Update New position using (24)
Determine the direction by (27)
Determine the new position by (28)
Check the bound conditions, update the new position by (33)
Calculate the objective function for the new solution (Xi ) and update the best global solution (X*)
i=i+1
i ≤ NPOP
Yes
No
Sort the population of solutions
Extract the elite solutions (elite positions of jellyfish)
i=1
Calculate (a) using (35) and (v) using (36)
t ≤ MaxIt/2
No
Yes
Choose the best global solution (X*)
Choose randomly solution among the elite solutions ( Xelite)
Calculate the distance between (Xi) and (X*) using (39)
Calculate the distance between (Xi) and (Xelite) using (37)
Update the new position of jellyfish using (40)
Update the new position of jellyfish using (38)
Check the bound conditions, update the new position using (33)
Calculate the objective function for the new solution (Xi ) and update the best global solution (X*)
i=i+1
i ≤ NPOP
Yes
No
t=t+1
t >MaxIt
No
Yes
Output the best global solution (X*)
End
Note:
*Lines in blue refers to JS Algorithm phases
*Lines in red refers to MFO Algorithm phases

**Figure 1.** Flowchart of the JS-MFO algorithm.

In such a manner, the elite solutions are exploited to achieve promising solutions in the early iterations, and inversely the best global solution is more exploited in the later iterations to improve the quality of the best solution.

$$D_i = |X_i(t) - X_{elite}| \tag{37}$$

$$X_i(t+1) = D_i.e^{bt}.cos(2\pi t) + X_{elite} \tag{38}$$

$$D_i = |X_i(t) - X^*| \tag{39}$$

$$X_i(t+1) = D_i\ .e^{bt}.cos(2\pi t) + X^* \tag{40}$$

## 4. Simulation Results and Discussion

### *4.1. Performance Evaluation of JS-MFO for Benchmark Functions*

The numerical efficiency of JS-FO was analyzed on eight popular benchmark functions to extensively evaluate its performance. These functions are divided into four unimodal functions and four multimodal functions, which are listed in Table 1 [28]. The unimodal functions were used to verify the exploitation of algorithms. The multimodal functions were used to check the exploration of algorithms. Experimental tests were performed for JS-MFO, MFO, and JS separately in order to verify the effectiveness of the hybrid approach. This task was accomplished for 30 trials, with the maximum number of iterations (MaxIt) = 500 and the population size equals 50 ($N_{POP}$ = 50). Statistical indices such as the average (mean), standard deviation (SD), and minimum (best) of 30 trials were determined for each benchmark function and are shown in Table 2. The findings in the previous table distinctly exhibit the capacity of the proposed JS-MFO in overcoming the basic algorithms JS and MFO in terms of solution quality (in reaching the near-global solution with slightly more execution time).

**Table 1.** Benchmark function characteristics.

| Fun | Equation | B | Min | D | T |
|---|---|---|---|---|---|
| $f_1$ | $f(x) = \sum_{i=1}^{n} x_i^2$ | [−100, 100] | 0 | 30 | Unimodal |
| $f_2$ | $f(x) = \sum_{i=1}^{n} ix_i^2$ | [−10, 10] | 0 | 30 | Unimodal |
| $f_3$ | $f(x) = \sum_{i=1}^{n-1} \left[100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2\right]$ | [−30, 30] | 0 | 30 | Unimodal |
| $f_4$ | $f(x) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$ | [−10, 10] | 0 | 30 | Unimodal |
| $f_5$ | $f(x) = -20exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - exp\left(\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi x_i)\right) + 20 + e$ | [−32, 32] | 0 | 30 | Multimodal |
| $f_6$ | $f(x) = \sum_{i=1}^{n} \left[x_i^2 - 10\cos(2\pi x_i) + 10\right]^2$ | [−5.12, 5.12] | 0 | 30 | Multimodal |
| $f_7$ | $f(x) = \frac{\pi}{n}\left\{10sin^2(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2\left[1 + 10\sin^2(\pi y_{i+1})\right] + (y_n - 1)^2\right\}$<br>$+\sum_{i=1}^{n} u(x_i, 10, 100, 4), \quad y_i = 1 + \frac{x_i+1}{4}$ | [−50, 50] | 0 | 30 | Multimodal |
| $f_8$ | $f(x) = 0.1\left\{\sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2\left[1 + \sin^2(3\pi x_{i+1})\right] + (x_n - 1)^2\left[1 + \sin^2(2\pi x_n)\right]\right\}$<br>$+\sum_{i=1}^{n} u(x_i,\ 5,\ 100,\ 4)$ | [−50, 50] | 0 | 30 | Multimodal |

**Table 2.** Comparison results between JS, MFO, and JS-MFO for $f_1$–$f_8$.

| Algorithm | Index | Unimodal Functions | | | | Multimodal Functions | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | $f_1$ | $f_2$ | $f_3$ | $f_4$ | $f_5$ | $f_6$ | $f_7$ | $f_8$ |
| JS | Mean | $2.00 \times 10^{-18}$ | $2.03 \times 10^{-19}$ | 0.7100 | $2.67 \times 10^{-10}$ | $3.49 \times 10^{-10}$ | 12.585 | $8.68 \times 10^{-5}$ | 0.0098 |
| | SD | $2.84 \times 10^{-18}$ | $2.68 \times 10^{-19}$ | 0.8835 | $2.65 \times 10^{-10}$ | $1.49 \times 10^{-10}$ | 9.0885 | $1.94 \times 10^{-4}$ | 0.0368 |
| | Best | $9.14 \times 10^{-20}$ | $1.57 \times 10^{-20}$ | 0.0179 | $3.97 \times 10^{-11}$ | $1.50 \times 10^{-10}$ | 3.0574 | $1.36 \times 10^{-6}$ | $1.57 \times 10^{-5}$ |
| | Time (s) | 1.2488 | 1.2332 | 1.0741 | 1.3384 | 1.4780 | 1.4203 | 2.4955 | 2.7288 |
| MFO | Mean | $2.67 \times 10^{3}$ | 383.527 | $1.88 \times 10^{4}$ | 32.0936 | 11.1412 | 139.56 | 4.5098 | $1.36 \times 10^{7}$ |
| | SD | $5.20 \times 10^{3}$ | 505.098 | $3.64 \times 10^{4}$ | 20.3943 | 8.7387 | 27.52 | 2.8850 | $7.48 \times 10^{7}$ |
| | Best | 0.3104 | 0.0524 | 178.5437 | 0.1066 | 0.1323 | 77.67 | 0.4706 | 0.2722 |
| | Time (s) | 0.7218 | 0.7127 | 0.8212 | 0.8202 | 1.2363 | 0.9198 | 2.0499 | 2.2922 |
| JS-MFO | Mean | $6.53 \times 10^{-83}$ | $4.94 \times 10^{-60}$ | 0.0520 | $2.59 \times 10^{-39}$ | $3.13 \times 10^{-15}$ | $3.89 \times 10^{-13}$ | $5.08 \times 10^{-29}$ | $2.41 \times 10^{-25}$ |
| | SD | $3.43 \times 10^{-82}$ | $1.79 \times 10^{-59}$ | 0.1296 | $1.40 \times 10^{-38}$ | $1.74 \times 10^{-15}$ | $1.98 \times 10^{-12}$ | $1.88 \times 10^{-28}$ | $1.31 \times 10^{-24}$ |
| | Best | $2.38 \times 10^{-89}$ | $1.19 \times 10^{-72}$ | $1.00 \times 10^{-8}$ | $1.25 \times 10^{-46}$ | $8.88 \times 10^{-16}$ | 0 | $3.63 \times 10^{-32}$ | $1.62 \times 10^{-32}$ |
| | Time (s) | 1.7842 | 1.7561 | 1.9401 | 1.9434 | 2.6328 | 2.0638 | 4.1697 | 4.6286 |

### *4.2. Implementation of JS-MFO for Solving OPF Problems*

Two test systems were used to investigate the aptitude of the proposed JS-MFO algorithm to reach the near-global OPF solution for different cases of mono-objective optimization problems: the IEEE 30-bus, a real part of the USA network, and the RIM 27-bus. The modeling of the tested networks was carried out in the MATLAB 2009b environment using a personal computer running Windows XP Professional, Pentium P-IV CPU 3 GHz processor, and RAM of 1 GB.

*A. IEEE 30-bus test system*

The IEEE 30-bus model has 6 generators, 41 branches (37 lines and 4 transformers with off-nominal tap ratios), and 24 load buses as shown in Figure 2. Shunt Volt Ampere Reactive (VAR) compensators are installed in buses 10, 12, 15, 17, 20, 21, 23, 24, and 29, where the reactive power injection is controlled between 0 and 5 MVAR as the lower and upper limits, respectively [11]. The total system demand was (2.834 + j1.262) p.u. for the apparent power at the 100 MVA base. Bus 1 was taken as the slack bus. The upper and lower limits of active power generation, the reactive power limits, and the generator cost coefficients are taken from [34].

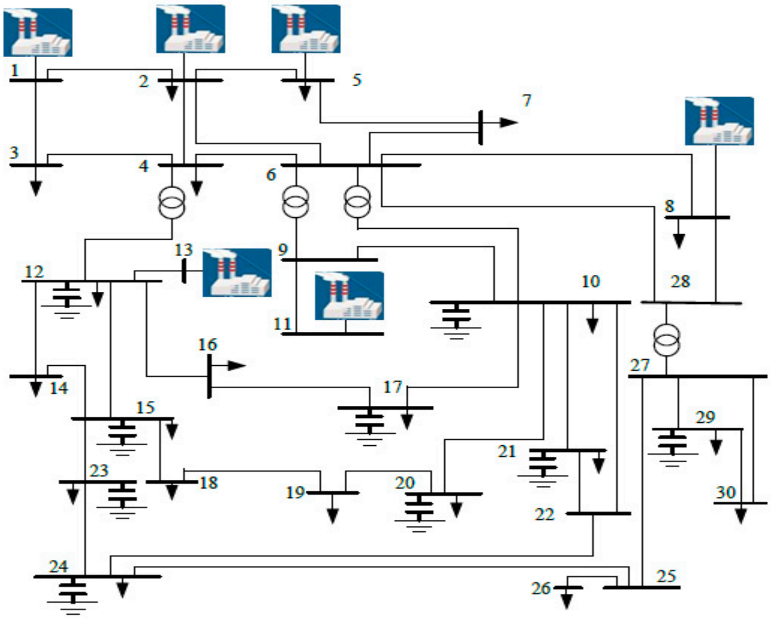

**Figure 2.** IEEE 30-bus single-line diagram.

In the base case, the system presents the following data: Fc (quadratic fuel cost) = 901.935 USD/h, $F_{Em}$ = 0.23908 ton/h, $F_{Plosses}$ = 5.8332 MW, $F_{VD}$ = 1.1663 p.u., and $L_{maxr}$ = 0.1731. The parameters of the JS, MFO. and JS-MFO algorithms are as follows: the population number (population size) $N_{POP}$ is 50, the maximum number of iterations *MaxIt* = 100, and the number of control variables is *D* = 25. For the present test system, six cases (six objective functions) were studied to evaluate the performance of the proposed JS-MFO compared to that of JS and MFO in order to reach the optimal solution. In order to validate the proposed JS-MFO in dealing with and solving complex OPF problems, 30 independent trial runs were performed for each case study by implementing JS, MFO, and JS-MFO. Three complex cases, particularly 1, 3, and 4, were considered to provide the minimum, average, maximum, and standard deviation (SD) for each objective function as shown in Table 3. A comparative report of each index between the JS, MFO, and JS-MFO algorithms, shown in the previous table, can clearly prove the superiority of JS-MFO over JS and MFO in providing the best solution quality and stability over the trial runs. The optimal control variable settings and the corresponding objective function value are depicted for each case in Tables 4 and 5. Executing 30 independent runs for each case led to extracting the solution for the run that provided the best solution.

**Table 3.** Comparative statistical study between JSO, MFO, and JS-MFO for case 1, case 3, and case 4 of OPF problem.

| **Study Case** | **Index** | **Method** | | |
|---|---|---|---|---|
| | | **JSO** | **MFO** | **JS-MFO** |
| Case 1<br>Quadratic total fuel cost (USD/h) | Minimum | 801.023 | 800.79 | **799.08** |
| | Average | 801.84 | 801.92 | **799.09** |
| | Maximum | 803.41 | 802.87 | **799.14** |
| | SD | 0.781 | 1.040 | **0.0133** |
| Case 3<br>Total fuel cost with valve point effect (USD/h) | Minimum | 929.48 | 924.02 | **918.07** |
| | Average | 943.60 | 929.06 | **925.61** |
| | Maximum | 959.14 | 984.18 | **952.54** |
| | SD | 12.07 | 13.59 | **12.05** |
| Case 4<br>Total active losses (MW) | Minimum | 3.232 | 3.049 | **2.881** |
| | Average | 3.189 | 3.443 | **2.909** |
| | Maximum | 3.266 | 3.75 | **3.095** |
| | SD | 0.053 | 0.2147 | **0.033** |

**Table 4.** Optimal tuning of control variables and objectives for Cases 1–3—IEEE 30-bus system.

| **Control Variables/ Objectives** | **Case 1** | | | **Case 2** | | | **Case 3** | | |
|---|---|---|---|---|---|---|---|---|---|
| | **JS** | **MFO** | **JS-MFO** | **JS** | **MFO** | **JS-MFO** | **JS** | **MFO** | **JS-MFO** |
| $P_{G1(MW)}$ | 177.68 | 178.13 | 177.15 | 64.529 | 63.804 | 63.921 | 199.60 | 199.60 | 199.589 |
| $P_{G2(MW)}$ | 48.689 | 49.146 | 48.727 | 67.393 | 67.79 | 67.490 | 20.144 | 20.000 | 20.002 |
| $P_{G5(MW)}$ | 21.448 | 21.824 | 21.329 | 49.958 | 50.00 | 49.999 | 19.625 | 22.290 | 22.027 |
| $P_{G8(MW)}$ | 19.436 | 19.307 | 20.897 | 34.974 | 34.99 | 34.999 | 24.920 | 27.904 | 23.029 |
| $P_{G11(MW)}$ | 13.069 | 11.740 | 11.924 | 29.983 | 30.00 | 29.999 | 15.038 | 10.003 | 14.982 |
| $P_{G13(MW)}$ | 12.276 | 12.426 | 12.005 | 39.993 | 40.00 | 39.999 | 14.528 | 14.576 | 13.192 |
| $V_{G1(p\cdot u)}$ | 1.0804 | 1.1000 | 1.1000 | 1.0600 | 1.059 | 1.099 | 1.0660 | 1.023 | 1.097 |
| $V_{G2(p\cdot u)}$ | 1.0607 | 1.0811 | 1.0876 | 1.0551 | 1.054 | 1.095 | 1.0352 | 0.999 | 1.076 |
| $V_{G5(p\cdot u)}$ | 1.0268 | 1.0491 | 1.0605 | 1.0198 | 1.035 | 1.076 | 0.9867 | 0.950 | 1.043 |
| $V_{G8(p\cdot u)}$ | 1.0353 | 1.0503 | 1.0684 | 1.0406 | 1.047 | 1.083 | 1.0032 | 0.966 | 1.058 |
| $V_{G11(p\cdot u)}$ | 1.0669 | 1.0363 | 1.1000 | 1.0613 | 1.100 | 1.097 | 1.0606 | 1.099 | 1.096 |
| $V_{G13(p\cdot u)}$ | 1.0633 | 0.9500 | 1.0991 | 1.0418 | 1.100 | 1.099 | 0.9937 | 1.087 | 1.098 |
| $T_{11(6\_9)}$ | 1.0092 | 1.0919 | 1.0666 | 0.9850 | 1.019 | 1.050 | 1.0399 | 0.921 | 0.988 |
| $T_{12(6\_10)}$ | 0.9600 | 1.0405 | 0.9061 | 1.0259 | 0.901 | 0.914 | 0.9871 | 0.985 | 1.022 |
| $T_{15(4\_12)}$ | 1.0236 | 1.0600 | 1.0099 | 1.0353 | 0.972 | 0.991 | 1.0016 | 1.082 | 1.031 |
| $T_{36(28\_27)}$ | 1.0080 | 1.0745 | 0.9748 | 0.9918 | 0.950 | 0.972 | 0.9932 | 0.931 | 0.968 |
| $Q_{C10(MVar)}$ | 3.6054 | 5.0000 | 4.9914 | 1.8390 | 3.675 | 4.984 | 1.6452 | 0 | 2.657 |
| $Q_{C12(MVar)}$ | 3.8140 | 3.4313 | 4.9341 | 3.6168 | 0 | 4.739 | 1.9352 | 0.017 | 3.497 |
| $Q_{C15(MVar)}$ | 2.0583 | 4.9744 | 4.9263 | 3.5515 | 5.0000 | 4.764 | 2.1570 | 0 | 1.655 |
| $Q_{C17(MVar)}$ | 2.7145 | 0 | 4.9544 | 1.8804 | 4.716 | 4.900 | 1.5654 | 0.137 | 2.442 |
| $Q_{C20(MVar)}$ | 2.8758 | 4.9988 | 4.8532 | 2.7821 | 2.839 | 4.906 | 3.1875 | 4.931 | 4.492 |
| $Q_{C21(MVar)}$ | 2.5908 | 4.7493 | 4.9983 | 3.8469 | 4.999 | 4.963 | 2.8867 | 5.00 | 1.828 |
| $Q_{C23(MVar)}$ | 2.3790 | 4.9460 | 4.0653 | 3.8902 | 1.710 | 4.369 | 4.2100 | 4.193 | 3.439 |
| $Q_{C24(MVar)}$ | 2.9189 | 5.0000 | 4.9839 | 2.9350 | 5.000 | 4.7945 | 3.4187 | 3.224 | 2.184 |
| $Q_{C29(MVar)}$ | 1.7836 | 5.0000 | 3.2182 | 3.7881 | 3.714 | 2.9024 | 4.0090 | 0.043 | 3.073 |
| $F_C$ (\$/h) | 801.02 | 800.79 | 799.085 | 944.30 | 944.6 | 943.690 | 922.48 | 924.02 | 918.07 |
| $F_{Em}$ (ton/h) | 0.3673 | 0.3689 | 0.3663 | 0.2049 | 0.2048 | 0.2047 | 0.4395 | 0.4403 | 0.439 |
| $F_{Ploss}$ (MW) | 9.2026 | 9.179 | 8.6368 | 3.4324 | 3.195 | 3.0095 | 10.466 | 10.975 | 9.424 |
| $F_{VD}$ (p.u) | 0.532 | 0.9889 | 1.6486 | 0.3994 | 1.486 | 1.9257 | 0.4706 | 0.329 | 1.195 |
| $F_{Lmax}$ | 0.1383 | 0.1145 | 0.1182 | 0.1339 | 0.120 | 0.1158 | 0.1433 | 0.1431 | 0.124 |

Case 1: Minimization of the quadratic total fuel cost function

The objective function in Equation (12) was evaluated for the optimal settings of the control variables by running the JS, MFO, and JS-MFO algorithms, and the simulation results are indicated in Table 4 for case 1. The optimal total fuel costs obtained were 799.085 USD/h, 801.02 USD/h, and 800.79 USD/h using JS-MFO, JS, and MFO, respectively. The best solution of the total fuel cost for JS-MFO was compared with that of the JS and MFO algorithms and other optimization techniques in the literature shown in Table 6, such as the Novel Improved Social Spider Optimization algorithm (NISSO) [35], Firefly Algorithm (FFA) [15], Moth Swarm Algorithm (MSA) [15], Efficient Sine Cosine Algorithm

(ESCA) [36], Symbiotic Organism Search algorithm (SOS) [35], Harris Hawks Optimization (HHO) [16], Dragonfly Algorithm (DA) [13], Adaptive Gaussian Teaching–Learning-based Optimization (AGLBO) [14], PSO-PS [21], DA-PSO [23], HF-PSO [24], Artificial Bee Colony based on hybrid fruit fly (HF-ABC) [25] and HFA-JAYA [27]. It is very clear from Table 6 that the JS-MFO approach is able to reduce the fuel cost to the lowest value of 799.085 USD/h. The percent cost savings with respect to the base cases of JS, MFO and JS-MFO are 11.188%, 11.2137%, and 11.4028 %, respectively, which lead to annual cost savings for JS-MFO of 16,951 USD/year compared to JS and 14,936 USD/year compared to MFO. The evolution of the total fuel cost during the simulation is indicated in Figure 3. It can be seen that the optimal total fuel cost converges towards the best optimal solution using JS-MFO.

**Table 5.** Optimum tuning of control variables and objectives for Cases 4, 7, and 8—IEEE 30-bus system.

| Control Variables and Objectives | Case 4 | | | Case 7 | | | Case 8 | | |
|---|---|---|---|---|---|---|---|---|---|
| | JS | MFO | JS-MFO | JS | MFO | JS-MFO | JS | MFO | JS-MFO |
| $P_{G1(MW)}$ | 63.591 | 51.483 | 51.349 | 178.73 | 177.89 | 176.174 | 174.09 | 179.01 | 177.466 |
| $P_{G2(MW)}$ | 68.150 | 80.00 | 79.988 | 48.711 | 48.76 | 48.826 | 48.904 | 49.483 | 48.746 |
| $P_{G5(MW)}$ | 49.993 | 50.00 | 49.997 | 22.006 | 21.03 | 21.970 | 21.632 | 22.102 | 21.785 |
| $P_{G8(MW)}$ | 34.962 | 35.00 | 34.950 | 18.213 | 21.49 | 21.356 | 22.195 | 16.766 | 20.639 |
| $P_{G11(MW)}$ | 29.969 | 29.965 | 29.998 | 12.433 | 12.239 | 12.722 | 13.182 | 11.959 | 11.458 |
| $P_{G13(MW)}$ | 39.965 | 39.999 | 39.997 | 13.246 | 12.008 | 12.133 | 12.351 | 13.317 | 12.015 |
| $V_{G1(p\cdot u)}$ | 1.087 | 1.064 | 1.099 | 1.0497 | 1.033 | 1.044 | 1.086 | 1.092 | 1.100 |
| $V_{G2(p\cdot u)}$ | 1.076 | 1.056 | 1.096 | 1.0429 | 1.009 | 1.026 | 1.066 | 1.07 | 1.084 |
| $V_{G5(p\cdot u)}$ | 1.055 | 1.039 | 1.077 | 1.0134 | 0.996 | 1.010 | 1.023 | 1.024 | 1.053 |
| $V_{G8(p\cdot u)}$ | 1.070 | 1.045 | 1.084 | 1.0000 | 1.004 | 1.0065 | 1.045 | 1.044 | 1.058 |
| $V_{G11(p\cdot u)}$ | 1.069 | 1.0993 | 1.100 | 1.0038 | 1.073 | 1.0575 | 1.083 | 1.100 | 1.099 |
| $V_{G13(p\cdot u)}$ | 1.081 | 1.0994 | 1.099 | 1.0446 | 1.0187 | 0.9899 | 1.078 | 1.093 | 1.097 |
| $T_{11(6\_9)}$ | 0.976 | 0.9735 | 1.072 | 1.0780 | 1.1000 | 1.0779 | 1.078 | 0.940 | 0.990 |
| $T_{12(6\_10)}$ | 1.025 | 0.9323 | 0.900 | 1.0690 | 0.9110 | 0.9010 | 1.069 | 0.900 | 0.933 |
| $T_{15(4\_12)}$ | 0.996 | 0.9693 | 0.999 | 1.0320 | 0.9987 | 0.9416 | 1.032 | 0.953 | 0.969 |
| $T_{36(28\_27)}$ | 0.988 | 0.9448 | 0.978 | 1.0680 | 0.9615 | 0.9671 | 1.068 | 0.921 | 0.948 |
| $Q_{C10(MVar)}$ | 3.388 | 3.7029 | 4.870 | 2.9344 | 5.000 | 4.1739 | 4.119 | 4.989 | 4.979 |
| $Q_{C12(MVar)}$ | 3.112 | 5.0000 | 4.984 | 0.6209 | 0.468 | 0.3890 | 3.651 | 4.995 | 4.999 |
| $Q_{C15(MVar)}$ | 3.713 | 5.0000 | 4.946 | 4.7753 | 5.000 | 4.9032 | 4.494 | 5.000 | 4.983 |
| $Q_{C17(MVar)}$ | 3.072 | 5.0000 | 4.964 | 2.8538 | 5.000 | 0.0844 | 3.350 | 4.660 | 4.648 |
| $Q_{C20(MVar)}$ | 2.838 | 3.9215 | 4.815 | 4.4568 | 4.781 | 4.9859 | 2.208 | 5.000 | 4.552 |
| $Q_{C21(MVar)}$ | 2.560 | 4.0782 | 4.810 | 2.5250 | 5.000 | 4.9713 | 3.417 | 1.153 | 4.985 |
| $Q_{C23(MVar)}$ | 2.860 | 3.435 | 3.893 | 3.8349 | 5.000 | 4.9970 | 4.128 | 4.899 | 4.800 |
| $Q_{C24(MVar)}$ | 3.114 | 4.742 | 4.977 | 4.0013 | 4.901 | 4.9836 | 3.787 | 4.964 | 4.897 |
| $Q_{C29(MVar)}$ | 2.915 | 3.915 | 2.865 | 2.7291 | 1.756 | 2.3740 | 4.334 | 0 | 4.680 |
| $F_C\ (\$/h)$ | 945.10 | 967.46 | 967.02 | 803.57 | 803.93 | 803.554 | 801.16 | 800.96 | 799.318 |
| $F_{Em}$ (ton/h) | 0.2048 | 0.207 | 0.207 | 0.3700 | 0.3681 | 0.3631 | 0.357 | 0.3712 | 0.367 |
| $F_{Ploss}$ (MW) | 3.232 | 3.049 | 2.881 | 9.9491 | 10.043 | 9.7845 | 8.958 | 9.244 | 8.711 |
| $F_{VD}$ (p.u) | 1.174 | 1.639 | 1.8806 | 0.1480 | 0.1136 | 0.0993 | 1.282 | 1.872 | 1.8981 |
| $F_{Lmax}$ | 0.126 | 0.118 | 0.1166 | 0.1371 | 0.1361 | 0.1370 | 0.117 | 0.1182 | 0.1141 |

**Table 6.** Comparisons of the results obtained for case 1 of the IEEE 30-bus system.

| Technique | Total Fuel Cost (USD/h) | Technique | Total Fuel Cost (USD/h) |
|---|---|---|---|
| JS-MFO | 799.085 | DA [13] | 802.316 |
| JS | 801.02 | AGTLBO [14] | 800.481 |
| MFO | 800.79 | PSO-PS [21] | 799.8723 |
| NISSO [35] | 799.762 | DA-PSO [23] | 802.124 |
| FFA [15] | 802.130 | HF-PSO [24] | 799.123 |
| MSA [15] | 802.223 | HF-ABC [26] | 800.212 |
| ESCA [36] | 800.219 | HFA-JAYA [27] | 800.480 |
| SOS [35] | 801.5733 | | |
| HHO [16] | 801.829 | | |

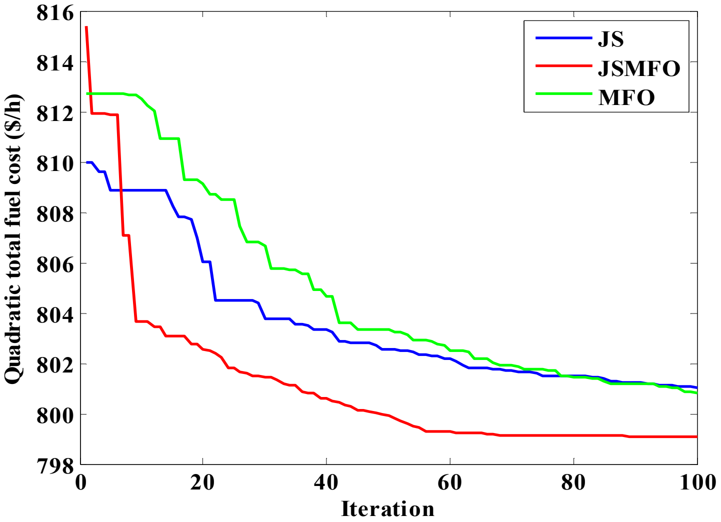

**Figure 3.** Convergence curves of JS, MFO and JS-MFO algorithms for case 1.

Case 2: Minimization of Total Gas Emissions

The total gas emission function in (13) was minimized using the proposed JS-MFO, MFO, and JS algorithms. The gas emission coefficients are given in [34]. The optimal solutions achieved using JS-MFO, MFO, and JS are 0.2047, 0.2048, and 0.2049 ton/h, respectively, based on the optimal setting of the control variables shown in Table 4 for case 2. Table 7 gives a statistical comparison report considering other optimization methods applied to the same test system and the same number of control variables as the Harris Hawks optimization algorithm (HHO) [16], Turbulent Flow Of Water-Based Optimization algorithm (TFWO) [17], hybrid DA-PSO [23], Adaptive Gaussian AGTLBO [14], and Social Spider Optimization algorithm (SSO) [37]. By examining the previous table, the minimum total gas emissions obtained by the proposed JS-MFO algorithm was shown to be better than that of the JS and MFO algorithms and the comparison optimization techniques reported in the literature. The percent gas emission reductions were 14.30%, 14.34%, and 14.38% for JS, MFO, and JS-MFO, respectively, referring to the base case. It can be seen that the proposed hybrid technique JS-MFO achieves total gas emission reductions of 1.752 tons/year and 0.876 tons/year compared to JS and MFO, respectively. The convergence of the total gas emission function with respect to the number of iterations is depicted in Figure 4, showing that the proposed JS-MFO algorithm gives faster convergence to the optimal solution with a better solution quality than the JS and MFO algorithms.

**Table 7.** Comparison report provided for case 2 of the IEEE 30-bus system.

| Technique | Total Gas Emissions (ton/h) | Technique | Total Gas Emissions (ton/h) |
|---|---|---|---|
| JS-MFO | 0.2047 | AGTLBO [14] | 0.2048 |
| JS | 0.2049 | SSO [35] | 0.2315 |
| MFO | 0.2048 | NISSO [35] | 0.2048 |
| HHO [16] | 0.2850 | SSA [37] | 0.205 |
| TFWO [17] | 0.2050 | PSO-SSA [37] | 0.205 |
| DA-PSO [23] | 0.2048 | | |

Case 3: Minimization of total fuel cost considering valve point effect

The objective function given by (14) is provided for the non-smooth curve of the total fuel cost function and that for the two generators in buses 1 and 2, where their cost coefficients are taken from [10]. The fuel cost curves of the remaining generators keep the same characteristics as in case 1. The minimum total fuel costs obtained using the proposed JS-MFO, JS, and MFO algorithms are 918.073 USD/h, 922.48 USD/h, and 924.02 USD/h, respectively, considering the valve point effect. The optimal solution reflects the optimal control variables presented in Table 4 for case 3. The percent cost savings considering the valve point effect for JS, MFO, and JS-MFO are 15.01%, 14.87%, and 15.42%, respectively, compared to the base case. It is clearly shown that JS-MFO gives the highest cost saving. The annual cost savings of JS-MFO correspond to 38,632 USD/year and 52,122 USD/year with respect to those of JS and MFO, respectively. The JS-MFO algorithm shows a better solution than that of the JS and MFO algorithms, and the other methods displayed in Table 8 such as the Improved Salp Swarm Algorithm (ISSA) [38], Salp Swarm Algorithm (SSA) [38], Teaching–Learning-based Optimization Algorithm (TLBO) [39], Gbest-guided ABC (Gbest-ABC) [40], and ABC-based Grenade Explosion Method (GABC) [41]. The progress of the non-quadratic total fuel cost with respect to the iteration evolutions is given in Figure 5 for the JS, MFO, and JS-MFO algorithms. These curves confirm the superiority of JS-MFO over JS and MFO in attaining the best global solution.

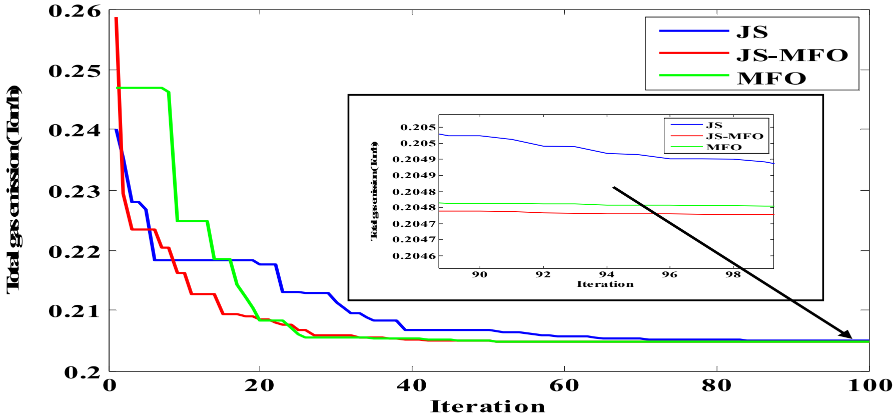

**Figure 4.** Convergence rates of JS, MFO, and JS-MFO algorithms for case 2.

**Table 8.** Comparison of the results obtained for case 3 of the IEEE 30-bus system.

| Technique | Total Fuel Cost (USD/h) | Technique | Total Fuel Cost (USD/h) |
|---|---|---|---|
| JS-MFO | 918.093 | TLBO [39] | 919.394 |
| JS | 922.48 | Gbest-ABC [40] | 931.745 |
| MFO | 924.02 | GABC1 [41] | 919.597 |
| ISSA [38] | 919.191 | GABC2 [41] | 918.435 |
| SSA [38] | 920.706 | | |
| IHS [38] | 919.843 | | |

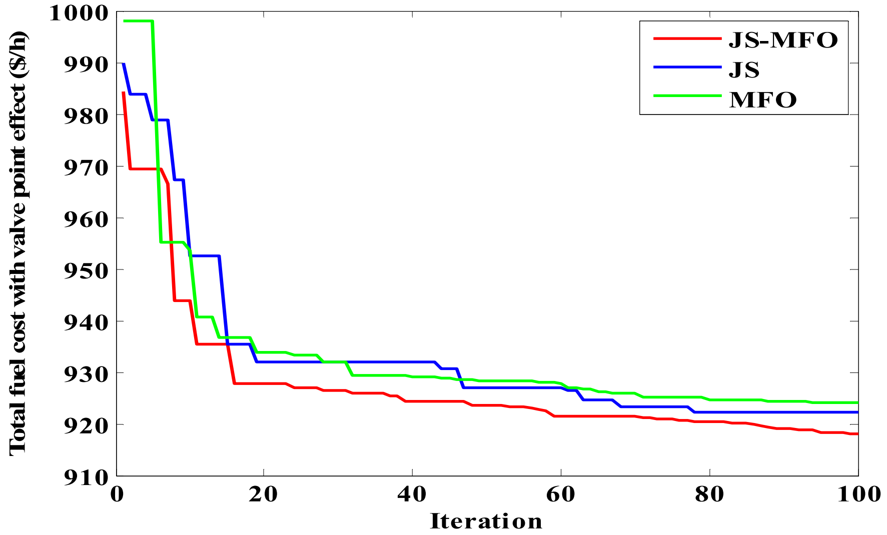

**Figure 5.** Convergence rates of JS, MFO, and JS-MFO algorithms for case 3.

Case 4: Minimization of Total Active Losses

Active transmission losses in Equation (15) are adopted as an objective function in this case and are minimized by carrying out the proposed JS-MFO, JS, and MFO techniques. The optimal total active losses resulting from the simulation are given in Table 5 for case 4, along with their corresponding control and state variables. It is reported that JS-MFO is able to reduce the active transmission losses to the lowest value of 2.8810 MW, while JS and MFO lead to 3.232 MW and 3.049 MW, respectively. The percent loss savings of the presented techniques JS, MFO, and JS-MFO are, respectively, 44.59%, 47.72%, and 50.60% with reference to the base case.

For comparison purposes, Table 9 depicts the optimal active transmission losses for JS-MFO, JS, MFO, and other techniques inspired from previous works in the literature such as AGTLBO [14], FFA [15], MSA [15], DE-HS [22], DA [23], DA-PSO [23], HFA-JAYA [27], SS0 [35], NISSO [35], ESCA [36], and Adaptive Multiple-Team Perturbation-Guiding JAYA [42]. From this table, it can be observed that the results obtained using JS-MFO are better than those of the other methods presented in the current literature. Figure 6 shows the evolution of total active losses with respect to the progression of iterations during the simulation, depicting the convergence characteristics of the JS and JS-MFO algorithms for this case.

**Table 9.** Comparison of the results for Case 4 of the IEEE 30-bus system.

| Technique | Total Transmission Losses (MW) | Technique | Total Transmission Losses (MW) |
|---|---|---|---|
| JS-MFO | 2.881 | DA [23] | 3.198 |
| JS | 3.232 | DA-PSO [23] | 3.189 |
| MFO | 3.049 | HFA-JAYA [27] | 4.529 |
| AGTLBO [14] | 3.090 | SS0 [35] | 3.911 |
| FFA [15] | 3.643 | NISSO [35] | 2.945 |
| MSA [15] | 3.649 | ESCA [36] | 3.021 |
| DE-HS [22] | 3.054 | AMTPG-JAYA [42] | 3.080 |

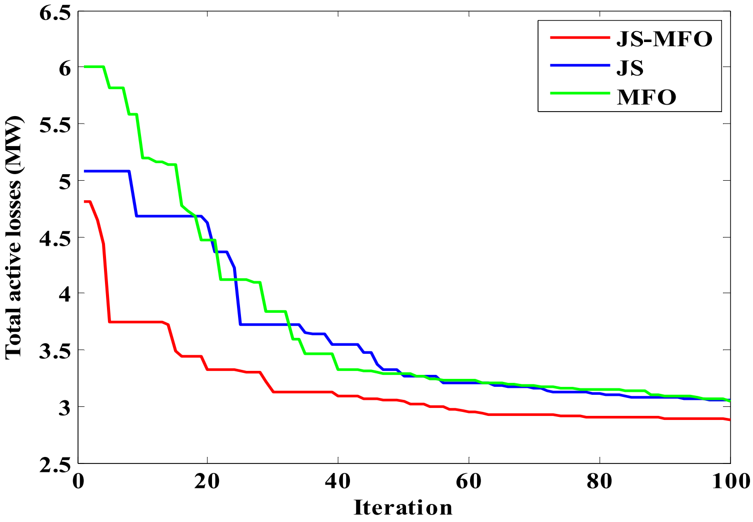

**Figure 6.** Convergence rates of JS, MFO, and JS-MFO for Case 4.

Case 7: Minimization of the total fuel cost along with the deviation of the total load bus voltage

This case study is devoted to the simultaneous minimization of the quadratic total fuel cost and the total load bus voltage deviation as the objective function depicted in Equation (21). The problem is solved by running the JS, MFO, and JS-MFO algorithms separately. The simulation results for case 7 are listed in Table 5 by showing the numerical optimal settings of the control and state variables. Table 10 is provided to compare the optimal total fuel cost and the total load bus voltage deviation of the JS-MFO method to those of the JS and MFO algorithms and the other techniques given by recently published works in the literature. It is clearly shown that JS-MFO dominates all other comparison techniques in terms of the optimal fuel cost with a slightly higher optimal voltage deviation compared to AGTLBO [14], HFA-JAYA [27], ESCA [36], PSO-SSA [37], MVO [43] and ECHT-DE [44]. The ESCA is reported to present the best optimal voltage deviation among all other techniques, but it provides the highest optimal voltage deviation.

**Table 10.** Comparisons of the results obtained for case 7 of the IEEE 30-bus system.

| Technique | Total Fuel Cost FC (USD/h) | Total Voltage Deviation FVD (p.u) |
|---|---|---|
| JS-MFO | 803.5549 | 0.0993 |
| JS | 803.5727 | 0.1480 |
| MFO | 803.93 | 0.1136 |
| AGTLBO [14] | 803.738 | 0.0947 |
| DA-PSO [23] | 803.66 | 0.1163 |
| HFA-JAYA [27] | 803.7036 | 0.0948 |
| ESCA [36] | 804.9968 | 0.09163 |
| PSO-SSA [37] | 803.989 | 0.0940 |
| MVO [43] | 803.908 | 0.1056 |
| ECHT-DE [44] | 803.719 | 0.0945 |

It is observed from the results that JS-MFO provides simultaneous reductions in quadratic total fuel cost and total voltage deviation of 10.90% and 91.48%, respectively, with reference to the base case, while the other existing techniques, JS, and MFO show less important results. By examining the convergence curves of the JS and JS-MFO algorithms given by Figure 7, it can be seen that the JS-MFO algorithm gives the best final global optimal solution compared to the JS and MFO algorithms.

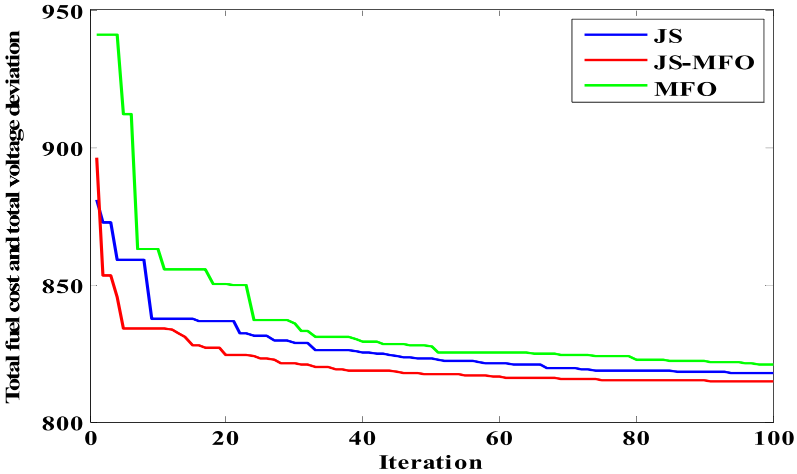

**Figure 7.** Convergence curves of JS, MFO, and JS-MFO algorithms for case 7.

Case 8: Minimization of Quadratic Total Fuel Cost Along With Vulnerability Stability Enhancement

The present case study examines the minimization of the total fuel cost along with the enhancement of voltage stability by optimizing the objective function mentioned in (22). The optimal settings obtained for the control and state variables after the JS, MFO, and JS-MFO algorithm running processes are illustrated in Table 5 for case 8. The optimal solution presented by JS-MFO gives the best simultaneous reductions in total fuel cost and voltage stability index than that of JS and MFO. By referring to the base case, JS-MFO presents, simultaneously, a percent cost saving of 11.37% and an improvement of the stability margin of 34.08%. A comparison in Table 11 with other meta-heuristics-based optimization techniques such as ESCA [36], PSO-SSA [37], AMTPG-JAYA [42], MVO [43], and ECHT-DE [44] shows that the proposed JS-MFO presents the best global solution among others.

**Table 11.** Comparison of the results obtained for case 8 of the IEEE 30-bus system.

| Technique | Total Fuel Cost FC (USD/h) | Voltage Stability Index $F_{Lmax}$ |
|---|---|---|
| JS-MFO | 799.3187 | 0.1141 |
| JS | 801.1621 | 0.1171 |
| MFO | 800.963 | 0.1182 |
| ESCA [36] | 800.410 | 0.1224 |
| PSO-SSA [38] | 830.352 | 0.1250 |
| AMTPG-JAYA [42] | 840.901 | 0.1240 |
| MVO [43] | 802.466 | 0.1146 |
| ECHT-DE [44] | 800.420 | 0.1374 |

B. Mauritanian electric power system RIM 27-bus:

B.1. Simulation of various cases for RIM 27-bus

The RIM 27-bus power grid is controlled by the SOMELEC society, involving three voltage levels: 225 kV, 90 kV, and 33 kV. This system model consists of six generation buses: 1 (slack bus), 8, 10, 23, 24, and 25, feeding a total load of (1.786 + j0.736) p.u. at a base power of 100 MVA, and three shunt compensators at buses 12, 14, and 16. Two solar power plants are installed on buses 8 and 23 with active power capacities of 50 MW and 15 MW, respectively, and a wind power plant on bus 10 with a capacity of 30 MW, as shown in Figure 8. Also, the RIM 27-bus system has 21 load buses and 36 branches. The voltage magnitude of the generation buses is controlled between 0.92 p.u. and 1.08 p.u. The number of control variables for this system is 15. In the base case, the RIM 27-bus system presents the following data: $F_{Plosses}$ = 6.6015 MW, $F_{VD}$ = 1.6358 p.u., and $L_{maxr}$ = 0. 522.

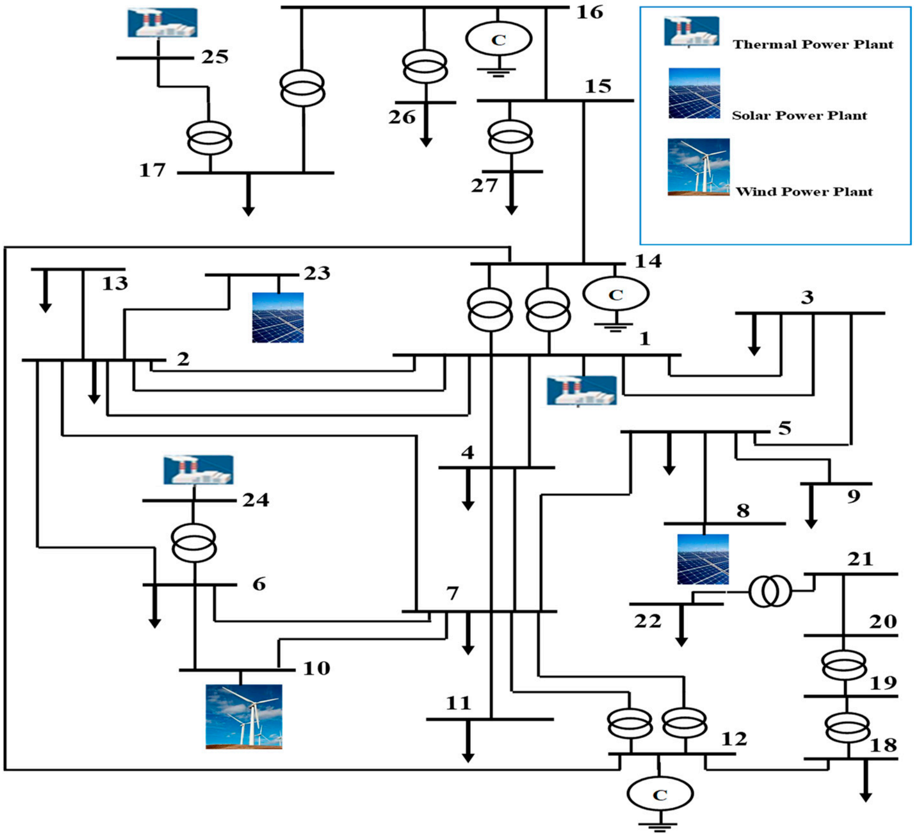

**Figure 8.** Single-line diagram of the Mauritanian electric power system 27 bus.

Case 4: Minimization of Total Active Losses for RIM 27-bus system:

In this case study, the aim is to minimize the transmission active losses of the RIM 27-bus system, adopting (15) as the objective function, and the simulation results are tabulated in

Table 12. The optimal active losses affected by JS, MFO, and JS-MFO are 3.35 MW, 3.56 MW, and 3.25 MW, which represent loss percent savings of 49.19%, 46.04%, and 50.67% compared to the base case, respectively. The best solution quality was found with the JS-MFO technique, which yielded the lowest total active power losses and the highest loss percent saving. The optimal total active transmission losses attributed to JS-MFO are 2.94% and 8.58% lower than those obtained by the JS and MFO algorithms. It is clearly illustrated in Figure 9 that the JS-MFO algorithm dominates the JS and MFO algorithms in terms of solution quality and convergence characteristics.

**Table 12.** Settings of control variables and objectives of RIM 27-bus system for cases 4–6.

| Control Variables and Objective | Case 4 | | | Case 5 | | | Case 6 | | |
|---|---|---|---|---|---|---|---|---|---|
| | JS | MFO | JS-MFO | JS | MFO | JS-MFO | JS | MFO | JS-MFO |
| $P_{G1(MW)}$ | 85.67 | 82.044 | 83.6279 | 90.667 | 49.75 | 36.9530 | 77.5546 | 108.26 | 109.0601 |
| $P_{G8(MW)}$ | 20.49 | 21.971 | 21.8570 | 20.005 | 50.00 | 49.8027 | 18.176 | 5.553 | 6.5210 |
| $P_{G10(MW)}$ | 19.27 | 19.327 | 17.8151 | 11.789 | 14.42 | 27.6150 | 25.6081 | 29.95 | 10.4073 |
| $P_{G23(MW)}$ | 9.345 | 8.581 | 9.0237 | 4.6829 | 9.936 | 3.0152 | 12.6120 | 14.66 | 3.5084 |
| $P_{G24(MW)}$ | 33.43 | 35.939 | 35.9926 | 30.700 | 36.00 | 35.9595 | 19.6385 | 10.71 | 23.6076 |
| $P_{G25(MW)}$ | 13.48 | 14.073 | 13.3123 | 24.333 | 2.745 | 29.9992 | 29.0962 | 12.86 | 29.9499 |
| $V_{G1(p\cdot u)}$ | 0.934 | 0.9618 | 0.9200 | 0.9570 | 0.954 | 0.9559 | 0.9518 | 0.9379 | 0.9445 |
| $V_{G8(p\cdot u)}$ | 0.941 | 0.9705 | 0.9271 | 0.9744 | 0.992 | 0.9936 | 0.9647 | 0.9398 | 0.9392 |
| $V_{G10(p\cdot u)}$ | 0.938 | 0.9663 | 0.9235 | 0.9648 | 0.966 | 0.9724 | 0.9646 | 0.9500 | 0.9492 |
| $V_{G23(p\cdot u)}$ | 0.935 | 0.9641 | 0.9219 | 0.9574 | 0.959 | 0.9574 | 0.9511 | 0.9422 | 0.9411 |
| $V_{G24(p\cdot u)}$ | 0.929 | 0.9504 | 0.9200 | 0.9374 | 0.941 | 0.9415 | 0.9421 | 0.9313 | 0.9303 |
| $V_{G25(p\cdot u)}$ | 0.970 | 0.9691 | 0.9431 | 0.9719 | 0.920 | 0.9993 | 0.9240 | 0.9200 | 0.9673 |
| $Q_{C12(MVar)}$ | −19.67 | −19.948 | −19.994 | −19.73 | −20 | −19.998 | −16.248 | −19.34 | −19.903 |
| $Q_{C14(MVar)}$ | −9.774 | −9.9932 | −9.9996 | −9.907 | −10 | −9.971 | −7.9491 | −9.760 | −7.072 |
| $Q_{C16(MVar)}$ | −24.97 | −25.000 | −24.998 | −24.944 | −25 | −24.999 | −24.216 | −25.00 | −15.7419 |
| $F_{Plosses}$ (MW) | 3.354 | 3.5621 | 3.2562 | 3.9357 | 4.470 | 4.9235 | 4.3556 | 3.8866 | 4.9284 |
| $F_{VD}$ (p.u) | 0.916 | 0.6045 | 1.2402 | 0.6547 | 0.732 | 0.6144 | 0.8005 | 0.9306 | 0.9650 |
| $F_{Lmax}$ | 0.426 | 0.3796 | 0.4597 | 0.3869 | 0.391 | 0.3876 | 0.3861 | 0.4201 | 0.3389 |

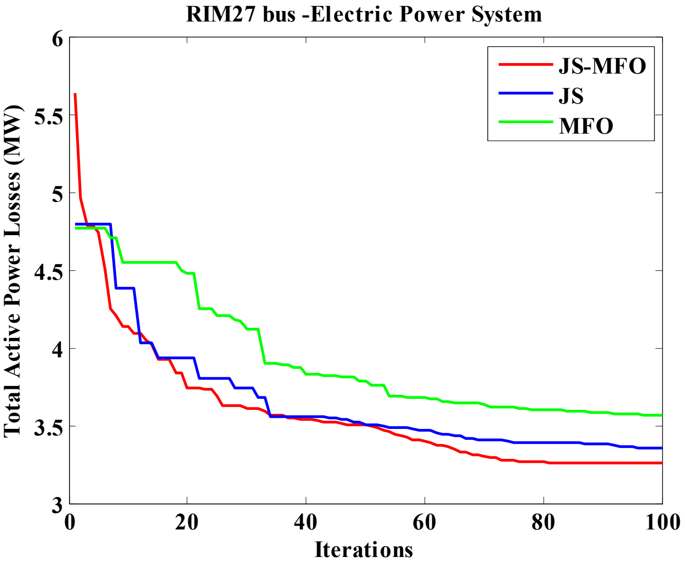

**Figure 9.** Convergence characteristics of JS, MFO, and JS-MFO algorithms for case 4.

Case 5: Minimizing of the total voltage deviation for the RIM 27-bus system:

The objective function presented in (16) is minimized in case 5 and depicts an optimal solution for the control and state variables shown in Table 12. The RIM 27-bus power system security is enhanced via the minimization of the total voltage deviation $F_{VD}$ using JS, MFO, and JS-MFO, signaling reductions of 59.97%, 55.25%, and 62.44%, respectively, compared to the base case. The JS-MFO pointed out the best optimal $F_{VD}$ compared to that of JS and MFO. The optimal total voltage deviation achieved by JS-MFO is 0.6144 p.u., which is better than that of the JS and MFO algorithms, which correspond to 0.6547 p.u. and 0.732 p.u., respectively. The best performance of the JS-MFO algorithm is verified in Figure 10, illustrating the convergence proprieties of JS-MFO overcoming those of the JS and MFO algorithms. Case 6: Voltage stability enhancement for the RIM 27-bus system:

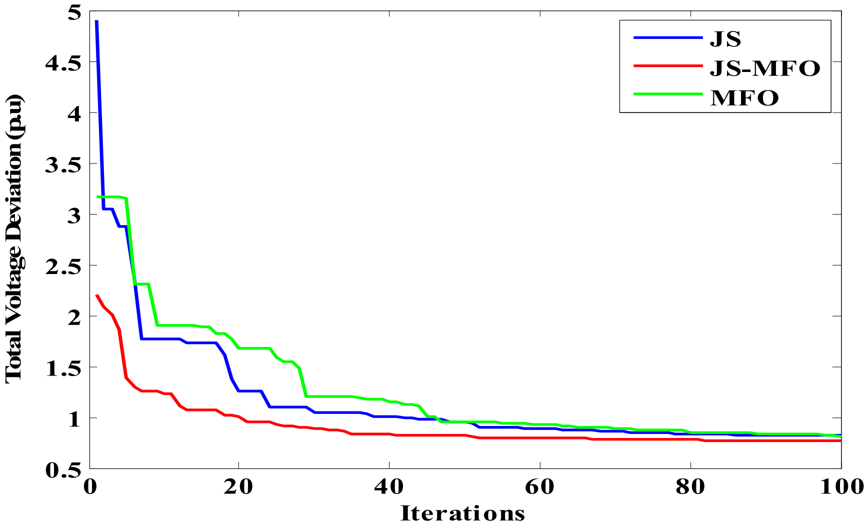

**Figure 10.** Convergence characteristics of JS, MFO, and JS-MFO algorithms for case 5.

The $F_{Lmax}$ voltage stability index is optimized for case 6 in order to enhance voltage stability using the JS, MFO, and JS-MFO algorithms. The stability margin of the RIM 27-bus system is improved using JS, MFO, and JS-MFO, where $F_{Lmax}$ reaches the optimal values of 0.3861, 0.4201, and 0.3389, respectively. The JS-MFO approach denoted a percent reduction of $F_{Lmax}$ of 12.22% compared to JSO and 19.33% compared to MFO, which proves the capacity of JS-MFO to overcome the JS and MFO algorithms.

It appears from Table 12 that the JS-MFO algorithm outperforms the JS and MFOs in solving the OPF problem. The JS-MFO simulation result provides 12.22% and 19.33% improvements in the stability margin compared to the JS and MFO algorithms, respectively.

B.2. Sizing of reactive power compensators for renewable sources

Table 13 points out the reactive power generation for each optimal state of cases 4, 5, and 6. The reactive power compensator sizing for each renewable energy source in generation buses 8, 10, and 23 is, respectively, −17.5/17.5 MVAR, −16/16 MVAR, and −15/15 MVAR. Figure 11 indicates the voltage magnitude profile for each optimal state for cases 4–6, describing that the voltage magnitude is within the predefined limits between 0.92 p.u. and 1.08 p.u.

**Table 13.** Reactive power generation in the optimal state for each case—RIM 27-bus power system.

| State Variables | Case 4 | Case 5 | Case 6 |
|---|---|---|---|
| **Generated Reactive Power** | **JS-MFO** | **JS-MFO** | **JS-MFO** |
| $Q_{G1(MVAR)}$ | −79.6379 | −104.3660 | −99.8009 |
| $Q_{G8(MVAR)}$ | −13.7852 | −6.2700 | −17.1219 |
| $Q_{G10(MVAR)}$ | −15.8118 | −6.1590 | −8.9077 |
| $Q_{G23(MVAR)}$ | −7.5879 | −5.8648 | −11.5592 |
| $Q_{G24(MVAR)}$ | −5.9939 | −25.0052 | −17.6271 |
| $Q_{G25(MVAR)}$ | 0.3272 | 5.1091 | 4.5488 |

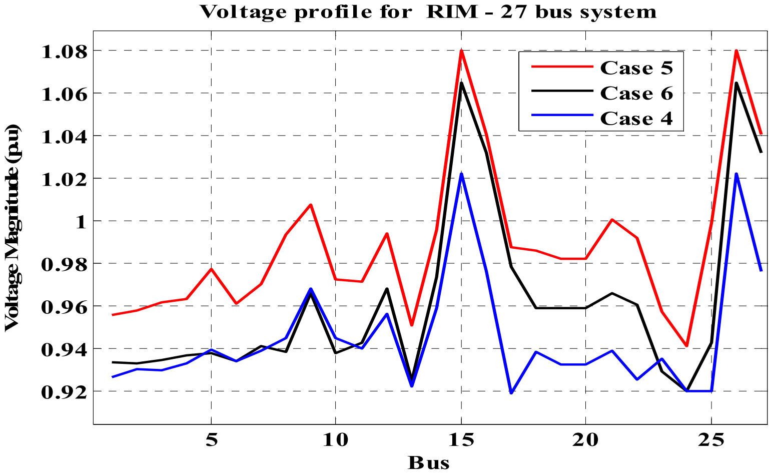

**Figure 11.** Voltage profile for each RIM 27-bus case using the JS-MFO algorithm.

## 5. Conclusions

A new meta-heuristic optimization technique has been developed in this paper, based on the hybridization of two methods, Jellyfish Search (JS) and Moth Flame Optimizer (MFO), called JS-MFO. This new hybrid approach is designed to improve the solution quality when dealing with complex Optimal Power Flow (OPF) problems. To improve the performance of the JS method in the exploitation mechanism, a spiral path of the MFO method on the best solutions found after the exploration phase is introduced. In this way, we can benefit from the advantages of the JS and MFO algorithms. The efficiency and precision of JS-MFO have been confirmed by applying it to several complex OPF optimization problems. These problems deal with the minimization of total power generation cost, gas emissions from power plants, total voltage deviation, and total transmission losses. Two test grids were employed to solve these problems: the popular IEEE 30-bus grid and the Mauritanian RIM 27-bus grid. For the IEEE 30-bus test system, JS-MFO presented a percent cost saving of 11.40% (optimization of the quadratic total fuel cost) with respect to the base case. This benefit is the highest when it is compared to those of the JS and MFO algorithms. It results in an annual cost saving of 16,951 USD/year with respect to the optimal value given by the JS algorithm. The percent gas emission reduction when dealing with the environmental optimization problem is about 14.38% for the JS-MFO algorithm (the highest value compared to those of JS and MFO) with reference to the base case. The JS-MFO algorithm shows a gas emission reduction of 1.752 tons/year with respect to the value given by the JS algorithm, and of 0.876 tons/year with respect to the amount given by the MFO algorithm. By optimizing the active transmission losses, a significant percent loss saving of 50.60% is assigned to the JS-MFO algorithm (greater than those of JS and MFO) with reference to the base case. For the RIM 27-bus power system, the loss percent saving using JS-MFO is 50.67% compared to the base case. The optimization of the total voltage deviation using JS-MFO showed a reduction of 62.44% with respect to the base case. The results obtained prove the capacity of the proposed approach to dominate the basic meta-heuristic techniques, JS, MFO, and others recently developed in the literature based on the comparison report for the same test network and the same objective functions. The aptitude of this hybrid JS-MFO technique to deal with complex OPF problems makes it competitive for (i) treating more problems in the power system, such as a variety of cases with complex objective functions, security constraints, prohibited zones, and different test systems, and (ii) offering prospects for improving the performance of the Mauritania RIM 27-bus grid by optimizing the location and sizing of new renewable sources. In view of the potential and superior characteristics of the proposed JS-MFO algorithm, a multi-objective algorithm based on JS-MFO is recommended to be developed and applied to solve multi-objective OPF problems.

**Author Contributions:** Conceptualization: C.M. and A.S.; methodology: A.S., D.N., and C.S.E.K.; software: C.M., A.S., and F.Z.A.; validation: R.A.E.-S., C.A., and D.N.; formal analysis: C.M., R.A.E.-S., F.H., and F.Z.A.; investigation: A.S., R.A.E.-S., D.N., and C.S.E.K.; resources: C.A. and C.S.E.K.; data curation: C.M., R.A.E.-S., and C.A.; writing—original draft preparation: C.M., F.Z.A., and F.H.; writing—review and editing: C.M., F.Z.A., F.H., and R.A.E.-S.; visualization: F.H., D.N., and C.S.E.K. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Data Availability Statement:** All required data are included in the text.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Nomenclature

| | |
|---|---|
| $P_{G1}$ | Generated active power of slack bus |
| $P_{Gi}$ | Generated active power of i-th bus |
| $V_{Li}$ | Voltage magnitude of load bus i |
| $Q_{Gi}$ | Reactive power output of generator i |
| $P_{Gi,min}$ | Lower active generation capacity in i-th bus |
| $P_{Gi,max}$ | Upper active generation capacity in i-th bus |
| $P_{Di}$ | Active power of load demand on bus i |
| $Q_{Di}$ | Reactive power of load demand on bus i |
| $N_g$ | Number of generators (power plants) |
| $N_{PQ}$ | Number of load buses (PQ buses) |
| $S_{Li}$ | Apparent power in transmission line i |
| $V_i$, $V_j$ | Voltage magnitudes at node i and node j, respectively |
| $S_{Li,min}$ | Lower limit of apparent power in transmission line i |
| $S_{Li,max}$ | Upper limit of apparent power in transmission line i |
| $V_{Li,min}$ | Lower limit of voltage magnitude in load bus i |
| $V_{Li,max}$ | Upper limit of voltage magnitude in load bus i |
| $Q_{Gi,min}$ | Lower reactive generation capacity in i-th bus |
| $Q_{Gi,max}$ | Upper reactive generation capacity in i-th bus |
| $V_{Gi}$ | Voltage magnitude of generator connected to bus i |
| $Q_{ci}$ | Reactive power injected by shunt VAR compensator in i-th bus |
| $V_{Gi,min}$ | Minimum voltage magnitude of generator connected to bus i |
| $V_{Gi,max}$ | Maximum voltage magnitude of generator connected to bus i |
| $\delta_{i,}$ $\delta_j$ | Voltage angles of buses i and j, respectively |
| $\delta_{ij}$ | Difference angle between voltage angles $\delta_i$ and $\delta_j$, respectively |
| $B_{ij}$ | Susceptance of the transmission line between buses i and j (imaginary part of the admittance) |
| $G_{ij}$ | Conductance of the transmission line between buses i and j (real part of the admittance) |
| $N_t$ | Number of regulating transformers |
| $N_B$ | Number of buses |
| $N_l$ | Number of the transmission lines |
| $N_C$ | Number of shunt compensators |
| $N_{PV}$ | Number of generator buses (PV buses) |
| $T_i$ | Tap setting of i-th transformer |
| $T_{i,min}$ | Lower tap setting of i-th transformer |
| $T_{i,max}$ | Upper tap setting of i-th transformer |
| $P_{G1}$ | Generated active power of slack bus |
| $N_g$ | Number of generator buses |
| $V_{Lj}$ | Voltage magnitude of load bus j |
| $N_{PQ}$ | Number of load buses (PQ buses) |
| $Q_{Gi}$ | Reactive power output of generator i |
| $Q_{Di}$ | Active power of load demand on bus i |
| $S_{Li}$ | Apparent power in transmission line i |
| $V_i$, $V_j$ | Voltage magnitudes at node i and node j, respectively |

| | |
|---|---|
| $P_{Gi}$ | Active power output of generator i |
| $P_{Di}$ | Active power of load demand on bus i |
| $V_{Gi}$ | Voltage magnitude of generator connected to bus i |
| $Q_{ci}$ | Reactive power injected by shunt VAR compensator in i-th bus |
| $B_{ij}$ | Susceptance of the transmission line between buses i and j (imaginary part of the admittance) |
| $G_{ij}$ | Conductance of the transmission line between buses i and j (real part of the admittance) |
| NT | Number of regulating transformers |
| $N_B$ | Number of buses |
| $T_i$ | Tap setting of transformer i |
| $N_C$ | Number of shunt compensators |
| $N_{PV}$ | Number of generator buses (PV buses) |
| $N_l$ | Number of the transmission lines |
| $\delta_{ij}$ | Difference angle between voltage angles$\delta_i$ and $\delta_j$, respectively |
| $\delta_{i,}\delta_j$ | Voltage angles of buses i and j, respectively |

## References

1. Frank, S.; Steponavice, I.; Rebennack, S. Optimal power flow: A bibliographic survey I. *Energy Syst.* **2012**, *3*, 221–258. [CrossRef]
2. Niknam, T.; Azizipanah-Abarghooee, R.; Narimani, M.R. Reserve constrained dynamic optimal power flow subject to valve-point effects, prohibited zones and multi-fuel constraints. *Energy* **2012**, *47*, 451–464. [CrossRef]
3. El Ela, A.A.A.; El-Sehiemy, R.A.; Shaheen, A.M.; Shalaby, A.S. Application of the crow search algorithm for economic environmental dispatch. In Proceedings of the 2017 IEEE Nineteenth International Middle East Power Systems Conference (MEPCON), Cairo, Egypt, 19–21 December 2017; pp. 78–83.
4. Chaib, A.E.; Bouchekara, H.; Mehasni, R.; Abido, M.A. Optimal power flow with emission and non-smooth cost functions using backtracking search optimization algorithm. *Int. J. Electr. Power Energy Syst.* **2016**, *81*, 64–77. [CrossRef]
5. Dommel, H.W.; Tinney, W.F. Optimal power flow solutions. *IEEE Trans. Power Appar. Syst.* **1968**, *10*, 1866–1876. [CrossRef]
6. Momoh, J.A.; El-Hawary, M.E.; Adapa, R. A review of selected optimal power flow literature to 1993. II. Newton, linear programming and interior point methods. *IEEE Trans. Power Syst.* **1999**, *14*, 105–111. [CrossRef]
7. Osman, M.S.; Abo-Sinna, M.A.; Mousa, A.A. A solution to the optimal power flow using genetic algorithm. *Appl. Math. Comput.* **2004**, *155*, 391–405. [CrossRef]
8. Abido, M.A. Optimal power flow using particle swarm optimization. *Int. J. Electr. Power Energy Syst.* **2002**, *24*, 563–571. [CrossRef]
9. Bouktir, T.; Slimani, L. Optimal power flow of the Algerian electrical network using an ant colony optimization method. *Leonardo J. Sci.* **2005**, *6*, 43–57.
10. Adaryani, M.R.; Karami, A. Artificial bee colony algorithm for solving multi-objective optimal power flow problem. *Int. J. Electr. Power Energy Syst.* **2013**, *53*, 219–230. [CrossRef]
11. Duman, S.; Güvenç, U.; Sönmez, Y.; Yörükeren, N. Optimal power flow using gravitational search algorithm. *Energy Convers. Manag.* **2012**, *59*, 86–95. [CrossRef]
12. El-Fergany, A.A.; Hasanien, H.M. Single and multi-objective optimal power flow using grey wolf optimizer and differential evolution algorithms. *Electr. Power Compon. Syst.* **2015**, *43*, 1548–1559. [CrossRef]
13. Ouafa, H.; Linda, S.; Tarek, B. Multi-objective optimal power flow considering the fuel cost, emission, voltage deviation and power losses using Multi-Objective Dragonfly algorithm. In Proceedings of the International Conference on Recent Advances in Electrical Systems. In Proceedings of the International Conference on Recent Advances in Electrical Systems, Hammamet, Tunusia, 22–24 December 2017.
14. Alanazi, A.; Alanazi, M.; Memon, Z.A.; Mosavi, A. Determining Optimal Power Flow Solutions Using New Adaptive Gaussian TLBO Method. *Appl. Sci.* **2022**, *12*, 7959. [CrossRef]
15. Vijaya Bhaskar, K.; Ramesh, S.; Abudhahir, A. Modern swarm intelligence-based algorithms for solving optimal power flow problem in a regulated power system framework. *Turk. J. Comput. Math. Educ. (TURCOMAT)* **2021**, *12*, 1786–1793. [CrossRef]
16. Islam, M.Z.; Wahab, N.I.; Veerasamy, V.; Hizam, H.; Mailah, N.F.; Guerrero, J.M.; Nasir, M.N.M. A Harris Hawks optimization based single-and multi-objective optimal power flow considering environmental emission. *Sustainability* **2020**, *12*, 5248. [CrossRef]
17. Sarhan, S.; El-Sehiemy, R.; Abaza, A.; Gafar, M. Turbulent flow of water-based optimization for solving multi-objective technical and economic aspects of optimal power flow problems. *Mathematics* **2022**, *10*, 2106. [CrossRef]
18. Mirjalili, S. (Ed.) *Handbook of Whale Optimization Algorithm: Variants, Hybrids, Improvements, and Applications*; Academic Press Elsevier: Amsterdam, The Netherlands, 2023.
19. Shaheen, A.M.; Elattar, E.E.; El-Sehiemy, R.A.; Elsayed, A.M. An improved sunflower optimization algorithm-based Monte Carlo simulation for efficiency improvement of radial distribution systems considering wind power uncertainty. *IEEE Access* **2020**, *9*, 2332–2344. [CrossRef]
20. Zhang, J.; Wang, J.S. Improved whale optimization algorithm based on nonlinear adaptive weight and golden sine operator. *IEEE Access* **2020**, *8*, 77013–77048. [CrossRef]
21. Berrouk, F.; Bounaya, K. Optimal power flow for multi-FACTS power system using hybrid PSO-PS algorithms. *J. Control. Autom. Electr. Syst.* **2018**, *29*, 177–191. [CrossRef]

22. Reddy, S.S. Optimal power flow using hybrid differential evolution and harmony search algorithm. *Int. J. Mach. Learn. Cybern.* **2019**, *10*, 1077–1091. [CrossRef]
23. Khunkitti, S.; Siritaratiwat, A.; Premrudeepreechacharn, S.; Chatthaworn, R.; Watson, N.R. A hybrid DA-PSO optimization algorithm for multi-objective optimal power flow problems. *Energies* **2018**, *11*, 2270. [CrossRef]
24. Khan, A.; Hizam, H.; Wahab, N.I.B.A.; Othman, M.L. Optimal power flow using hybrid firefly and particle swarm optimization algorithm. *PLoS ONE* **2020**, *15*, e0235668. [CrossRef] [PubMed]
25. Salhi, S.; Naimi, D.; Salhi, A.; Abujarad, S.; Necira, A. A novel hybrid approach based artificial bee colony and salp swarm algorithms for solving ORPD problem. *Indones. J. Electr. Eng. Comput. Sci.* **2021**, *23*, 1825–1837. [CrossRef]
26. Mallala, B.; Papana, V.P.; Sangu, R.; Palle, K.; Chinthalacheruvu, V.K. multi-objective optimal power flow solution using a non-dominated sorting hybrid fruit fly-based artificial bee colony. *Energies* **2022**, *15*, 4063. [CrossRef]
27. Alghamdi, A.S. A Hybrid Firefly–JAYA Algorithm for the Optimal Power Flow Problem Considering Wind and Solar Power Generations. *Appl. Sci.* **2022**, *12*, 7193. [CrossRef]
28. Chou, J.-S.; Truong, D.-N. A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean. *Appl. Math. Comput.* **2021**, *389*, 125535. [CrossRef]
29. Rai, H.; Verma, H.K. Economic load dispatch using jellyfish search optimizer. In Proceedings of the 2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT), Bhopal, India, 18–19 June 2021. [CrossRef]
30. Mirjalili, S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowl.-Based Syst.* **2015**, *89*, 228–249. [CrossRef]
31. Khalilpourazari, S.; Pasandideh, S.H. Multi-item EOQ model with nonlinear unit holding cost and partial backordering: Moth-flame optimization algorithm. *J. Ind. Prod. Eng.* **2016**, *34*, 42–51. [CrossRef]
32. Kessel, P.; Glavitsh, H. Estimating the voltage stability of power systems. *IEEE Trans. Power Syst.* **1986**, *1*, 346–354. [CrossRef]
33. Shaheen, A.M.; El-Sehiemy, R.A.; Farrag, S.M. A novel adequate bi-level reactive power planning strategy. *Int. J. Electr. Power Energy Syst.* **2016**, *78*, 897–909. [CrossRef]
34. Bouktir, T.; Slimani, L. Object-Oriented Economic Power Dispatch of Electrical Power System with minimum pollution using a Genetic Algorithm. *J. Electr. Syst.* **2005**, *1*, 19–34.
35. Almabsout, E.A.; El-Sehiemy, R.A.; An, O.N.U.; Bayat, O. A Hybrid Local Search-Genetic Algorithm for Simultaneous Placement of DG Units and Shunt Capacitors in Radial Distribution Systems. *IEEE Access* **2020**, *8*, 54465–54481. [CrossRef]
36. Messaoudi, A.; Belkacemi, M. Optimal Power Flow Solution using Efficient Sine Cosine Optimization Algorithm. *J. Intell. Syst. Appl.* **2020**, *2*, 34–43. [CrossRef]
37. El Sehiemy, R.A.; Selim, F.; Bentouati, B.; Abido, M.A. A novel multi-objective hybrid particle swarm and salp optimization algorithm for technical-economical-environmental operation in power systems. *Energy* **2020**, *193*, 116817. [CrossRef]
38. Abd el-sattar, S.; Kamel, S.; Ebeed, M.; Jurado, F. An improved version of salp swarm algorithm for solving optimal power flow problem. *Soft Comput.* **2021**, *25*, 4027–4052. [CrossRef]
39. Xing, H.; Song, F.; Yan, L.; Pan, W. A modified artificial bee colony algorithm for load balancing in network-coding-based multicast. *Soft Comput.* **2019**, *23*, 6287–6305. [CrossRef]
40. Roy, R.; Jadhav, H.T. Optimal power flow solution of power system incorporating stochastic wind power using Gbest guided artificial bee colony algorithm. *Int. J. Electr. Power Energy Syst.* **2015**, *64*, 562–578. [CrossRef]
41. Salhi, A.; Naimi, D.; Bouktir, T. Optimal power flow resolution using artificial bee colony algorithm-based grenade explosion method. *J. Electr. Syst.* **2016**, *12*, 734–756.
42. Warid, W. Optimal power flow using the AMTPG-Jaya algorithm. *Appl. Soft Comput.* **2020**, *91*, 106252. [CrossRef]
43. Bentouati, B.; Chettih, S.; Jangir, P.; Trivedi, I.N. A solution to the optimal power flow using multi-verse optimizer. *Electr. Syst.* **2016**, *12*, 716–733.
44. Biswas, P.P.; Suganthan, P.N.; Mallipeddi, R.; Amaratunga, G.A. Optimal power flow solutions using differential evolution algorithm integrated with effective constraint handling techniques. *Eng. Appl. Artif. Intell.* **2018**, *68*, 81–100. [CrossRef]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.