mathematics

Article

Single- and Multi-Objective Modified Aquila Optimizer for Optimal Multiple Renewable Energy Resources in Distribution Network

Mohammed Hamouda Ali [1], **Ahmed Tijani Salawudeen** [2], **Salah Kamel** [3], **Habeeb Bello Salau** [4], **Monier Habil** [5,*] and **Mokhtar Shouran** [5]

[1] Department of Electrical Engineering, Faculty of Engineering, Al-Azhar University, Cairo 11651, Egypt; eng_mohammedhamouda@azhar.edu.eg
[2] Department of Electrical and Electronics Engineering, University of Jos, Jos 930222, Nigeria; atsalawudeen@unijos.edu.ng
[3] Department of Electrical Engineering, Faculty of Engineering, Aswan University, Aswan 81542, Egypt; skamel@aswu.edu.eg
[4] Department of Computer Engineering, Ahmadu Bello University Zaria, Zaria 810107, Nigeria; bellosalau@abu.edu.ng
[5] Wolfson Centre for Magnetics, School of Engineering, Cardiff University, Cardiff CF24 3AA, UK; shouranma@cardiff.ac.uk
* Correspondence: habilmm@cardiff.ac.uk

Abstract: Nowadays, the electrical power system has become a more complex, interconnected network that is expanding every day. Hence, the power system faces many problems such as increasing power losses, voltage deviation, line overloads, etc. The optimization of real and reactive power due to the installation of energy resources at appropriate buses can minimize the losses and improve the voltage profile, especially for congested networks. As a result, the optimal distributed generation allocation (ODGA) problem is considered a more proper tool for the processes of planning and operation of power systems due to the power grid changes expeditiously based on the type and penetration level of renewable energy sources (RESs). This paper modifies the AO using a quasi-oppositional-based learning operator to address this problem and reduce the burden on the primary grid, making the grid more resilient. To demonstrate the effectiveness of the MAO, the authors first test the algorithm performance on twenty-three competitions on evolutionary computation benchmark functions, considering different dimensions. In addition, the modified Aquila optimizer (MAO) is applied to tackle the optimal distributed generation allocation (ODGA) problem. The proposed ODGA methodology presented in this paper has a multi-objective function that comprises decreasing power loss and total voltage deviation in a distribution system while keeping the system operating and security restrictions in mind. Many publications investigated the effect of expanding the number of DGs, whereas others found out the influence of DG types. Here, this paper examines the effects of different types and capacities of DG units at the same time. The proposed approach is tested on the IEEE 33-bus in different cases with several multiple DG types, including multi-objectives. The obtained simulation results are compared to the Aquila optimizer, particle swarm optimization algorithm, and trader-inspired algorithm. According to the comparison, the suggested approach provides a superior solution for the ODGA problem with faster convergence in the DNs.

Keywords: radial distribution network; multi-objective optimization; renewable distributed generation; modified Aquila optimizer; real loss reduction; voltage deviation minimization

MSC: 13P25; 37N40; 65K10

check for updates

Citation: Ali, M.H.; Salawudeen, A.T.; Kamel, S.; Salau, H.B.; Habil, M.; Shouran, M. Single- and Multi-Objective Modified Aquila Optimizer for Optimal Multiple Renewable Energy Resources in Distribution Network. *Mathematics* **2022**, *10*, 2129. https://doi.org/10.3390/math10122129

Academic Editor: Xavier Blasco

Received: 3 May 2022
Accepted: 15 June 2022
Published: 19 June 2022

1. Introduction

Optimization can simply be described as the process of finding the best or optimal solution among several possible solutions for a given function [1]. Often times, this requires finding the best combination of different specific parameters in the design process that gives the best value for the function at hand. Optimization processes have found widespread applications to real-world problems, ranging from structural engineering problems, transportation problems, computer vision/image processing problems, power system problems, etc. Several optimization algorithms have been developed and categorized based on their design processes to solve these real-life problems. Although tremendous success has been achieved, we note that towards ensuring proper performance of these algorithms, specifically the metaheuristic-based approaches, there is a need to maintain a balance between the exploitation and exploration phase of the search process in the algorithms [2,3], as the skewness of either of the processes leads to convergence to the local optimal solution. In this regard, several metaheuristic-based optimization algorithms have been proposed as documented in [4] and, more recently, an AO was developed based on different hunting schemes of the Aquila bird for different kinds of prey [2]. Despite the considerable performance recorded by the AO, attributed to its good global exploration capability, it sometimes converges to local optimal solution due to its insufficient local exploitation capability [3]. To address this limitation, this paper modifies the AO using a quasi-oppositional-based learning operator. The proposed variant of AO, called the modified Aquila optimizer, is used to solve the optimal distributed generation problem (ODGP) in power systems.

The electrical grid is currently altering and maturing into a more responsive, and perhaps more controlled, system than in the past. This opens the door to new developments and upgrades. One of the most noticeable characteristics of smart grids is the widespread incorporation of DGs into electric networks. If DGs are not adequately planned for, they disrupt the network's voltage stability conditions and power flow. Thus, voltage regulation and lessening the power loss are the major issues to be addressed. However, the inclusion of renewable energy sources in the traditional system poses several obstacles [5]. The size and position of DGs are critical elements in using the DG strategy for loss reduction [6].

The balancing of active and reactive power and the demand to maintain acceptable system voltages is a critical need for successful electric power system operation. The peripheral technique of balancing the system's power and voltage is to install DG units at critical nodes. The grid-connected DG units serve as a power source to balance the system's demand for power. The optimal functioning of a power system must come before the optimal planning of facilities such as production plants, and transmission and distribution networks. To treat the large-scale optimal power flow problem, it is divided into two parts: real power (P) optimization and reactive power (Q) optimization [7]. The P problem intends to reduce production costs while the system voltages are kept the same, whereas the Q problem seeks to lessen transmission loss while keeping real generated power fixed.

The network reconfiguration is carried out by adding multiple DG units with dissimilar PQ capacitiesto the traditional grid to turn it into a smart grid. Ref. [8] addresses several instances involving the actual and reactive power incorporation of RESs. Both the placement and scale of DGs have a considerable influence on system losses in a distribution network when including RESs.

Finding the best DG size and location is the most important aspect of DG deployment. Many studies focus solely on reducing power loss as a primary goal for addressing the DG deployment problem, utilizing a variety of analytic and computational intelligence methods [9–14]. Metaheuristic techniques are ideally adequate for the DG placement challenge. Voltage augmentation and loss reduction are now essential considerations in power system planning. Loss reduction and voltage increase are based on proper DG placement, which is achieved by using the best DG position configuration and DG size.

Many objectives are simultaneously optimized in the multi-objective DG allocation (MODGA) problem, resulting in a convenient decision. The e-constrained technique or the weighting sum approach is used to address MO optimization issues. The MO rendering indicator is determined as a weighted sum of several technical aspects such as the reduction of actual power loss, minimization of voltage deviation, and operational cost reduction [15–18]. Many studies have given reviews as well as polls on the DG layout issue in recent years [9–18], which are based on a DG short caption, objective functions (OFs), restrictions, and distinctive strategies.

Authors of [19] merged three goal functions (i.e., load shedding, total power, and voltage deviations) and presented them as a multi-objective index with different predilection weightages by using particle swarm optimization and genetic algorithm approaches to find the best position and size of DGs. The studies in [20] used PSO, GSA, and hybrid PSO-GSA algorithms to address OFs such as actual power loss and voltage profile adjustment. Ref. [21] examined power loss decreasing, voltage profile boosting, and operating cost decline as MO, and the DG allocation issue was handled using a whale optimization (WOA) algorithm.

In [22], the objective function to be reduced was obtained by power loss reduction, and a new analytical approach was utilized to minimize the objective function through determination of the ideal DG's position and size. The quasi-oppositional swine influenza model-based optimization with quarantine (QOSIMBO-Q) [23] method was used to address the ODGA problem to reduce real power loss, better voltage stability, and foster the voltage profile.

In [24] a backtracking search technique (BS) was used to optimize the multi-objective function and that included lessening power losses, meliorating the voltage profile, and combining static voltage stability indices. Fuzzy expert rules were used to identify DG sites based on bus voltages and loss sensitivity factors (LSFs). Ref. [25] employed LSFs to determine the placement of DGs, and the invasive weed optimization (IWO) method was used to determine the size of DGs; using this method, the power loss lowering percent was close to optimal when compared to other approaches. Ref. [26] employed a multi-objective performance index (MOPI) to determine the appropriate DG allocation for voltage stability improvement in a DS using a voltage sensitivity index and bus participation factors derived from the continuation power flow for voltage stability enhancement and energy loss minimization. The authors of [27] introduced a novel version called optimal RES placement (ORESP) for optimal placement and sizing of distributed generation. In [28], the authors created a multi-objective shuffled bat (MOSB) method using the shuffled frog-leaping algorithm and the bat algorithm to select DG location and sizing by minimizing the multi-objective function while accounting for power losses, voltage variance, and cost.

In [29], an improved differential search algorithm (IDSA) was employed to address the optimization issue using a Pareto optimal strategy, with the technological and economic benefits of DGs serving as targets. Refs. [30,31] examined power system expansion and total investment and expansion costs, which were handled using a novel heuristic, namely, the shuffled frog-leaping algorithm (SFLA). Authors of [32] introduced the MO opposition-based chaotic differential evolution (MOCDE) method for tackling the MO issue through power loss reduction, annual energy loss, and voltage drift as multi-objectives. In [33], an improved raven roosting optimization (IRRO) approach was established for handling the ODGA problem considering power losses, line loading, voltage stability, voltage profile, and net saving cost as a weighted MO in the DS. The authors of [34] used an enhanced metaheuristic technique, the quasi-oppositional chaotic symbiotic organisms search (QOCSOS) algorithm to choose the DG's best location and size considering power loss reduction, voltage profile refinement, and raising of the voltage stability. The authors of [35] presented a novel technique called the trader-inspired algorithm (TIA), which was used to discover the optimal selection of DG and power factor for lowering power losses and improving the voltage stability by using a multi-objective function. The key disadvantages of all of the preceding approaches are slow convergence and finding near-optimal solutions. RESP problems in the previous studies are elaborated in Table 1.

Table 1. Applied approaches for RESP techniques.

Method	Objective Function	Approach Discussion	Test Systems	Ref.	Year
ALO	Lessen real power losses	LSF and VSF were utilized for finding the optimal bus locations for multiple DGs. Then, the optimal size of DGs was identified using ALO algorithm by minimizing the total real power loss of the DN.	IEEE 33-, 69- and 119-bus	[8]	2021
SA-PSO	Active power loss minimization	A new hybrid SAPSO algorithm was used to determine the correct DG installation.	IEEE 33-bus	[9]	2020
IWHO	Active power loss and reliability indices	A novel IWHO algorithm was used to define the optimal DG siting and sizing.	IEEE 33-, 69- and 119-bus	[10]	2022
MOHSA	Total losses and voltage deviation	For DG allocation, total losses and TVD were employed as OF indices.	Debre Markos feeder 3	[11]	2019
Hybrid (TSA-SCA)	Active power loss, TVD, and VSI	Multi-objective method for determining the best DG location and size.	IEEE 69-bus	[12]	2021
MFO	Active power loss, emission, and VPI	Developing RES to maintain sustainable development.	IEEE 33-bus	[14]	2021
MOPSO	Active power loss and TVD	MOPSO was utilized to determine the optimal size and location of DGs before and after reconfiguration of the radial network.	IEEE 33-bus	[15]	2021
Hybrid (EGWO-PSO)	Active power loss, VDI, emission, cost, and VSI	MOF was used for optimal placement and sizing of DGs and CBs.	IEEE 33- and 69-bus	[17]	2021
Hybrid (GA-PSO)	Active power loss and voltage regulation	Convert the conventional distribution systems into resilient autonomous microgrid networks by inserting optimal DGs.	IEEE 33- and 69-bus	[19]	2019
Hybrid (PSO-GSA)	Total losses and voltage deviation	Firstly, LSF was used to determine the optimal location. Then, the optimal size of DGs was determined by PSO, GSA, and hybrid PSO-GSA algorithms.	IEEE 33-bus	[20]	2016
WOA	Active power loss, voltage profile, and operating cost	WOA was used to find the optimal DGs for MOF.	IEEE 33- and 69-bus	[21]	2018
Analytical approach	Active power loss minimization	A new simplified analytical approach was introduced to find the optimal choice of DGs.	IEEE 12-, 33- and 69-bus	[22]	2019
BS-FLC	Total power losses, TVD, and VSI	Backtracking search algorithms were demonstrated and marked on the DN to determine the best solution. Among the best solutions, fuzzy was a decision-maker to decide the best compromise solution among the offered Pareto optimal solutions.	IEEE 33- and 94-bus	[24]	2016
IWO	Active power loss, voltage deviation, and operating cost	LSF was used to determine the bus location, whereas IWO algorithm was used to find the best size.	IEEE 33- and 69-bus	[25]	2016
Analytical	Total losses index, VSI, VSM, and AVDI	The choice of the correct type of DG unit and its optimal location to enhance the voltage stability and improve the voltage profile was based on the analysis for some indexes.	IEEE 33- and 136-bus	[26]	2018
UPSO	Active power loss minimization	ODGP was solved by selecting the optimal size and location of RES. UPSO was used for solving ODGP and ORESP.	IEEE 33- and 118-bus	[27]	2019
MOCDE	Active power losses, VD, and cost.	The proposed technique used logistic mapping to generate a chaotic sequence for control parameters for solving the MO problem.	IEEE 33- and 69-bus	[32]	2019
IRRO	Total losses, VPI, VSI, and LLI	MOF considered to improve the technical aspects and net economical saving cost with DG units integrated on RDS. Pareto was used to make a set of the best solutions.	IEEE 33- and 69-bus	[33]	2020
QOCSOS	Active power loss, VD, and VSI	QOCSOS was utilized to determine the ODGA by solving the MOF.	IEEE 33-, 69- and 118-bus	[34]	2020
TIA	Active power loss and VSI	TIA was employed to determine the optimal allocation of multiple types of RESs-DGs, considering MOF.	IEEE 33-bus	[35]	2021

Finally, the main contribution of this paper can be summarized below:

- This paper proposes an improved variant of the Aquila optimizer (AO), called the modified Aquila optimizer, to beat all of the aforementioned drawbacks in terms of the complexity of the algorithms and slower convergence and stagnation at the end of the optimization process, as it has only a few control parameters in comparison with other intelligent algorithms.
- The new variant integrates the quasi-oppositional operator with the AO for im-proved convergence and quality of solutions.
- The efficacy of the outcomes of this approach are demonstrated as the effectiveness of the MAO; the authors first test the algorithm performance on twenty-three competition on evolutionary computation (CEC) benchmark functions considering different dimensions.
- For the first time, a modified Aquila optimizer (MAO) is introduced in the power system to tackle the optimal distribution generation allocation (ODGA) problem, augment search quality, and shun an early convergence to a local minimum.
- This research assesses the work in two directions in addition to optimal allocation and size. The first part examines the influence of expanding the number of DGs, whereas the second part discusses and compares the different PQ capabilities of DGs producing real power only, reactive power only, and combined real and reactive power.
- The efficacy of the outcomes of this approach are also demonstrated in terms of lowering losses and meliorating the voltage deviation considering distinct restrictions such as voltage limits, real power boundaries, DG capacity limit, and DG location to obtain the optimal solution.
- Smart grid concepts based on network reconfiguration, obtained through the integration of DGs by using soft computing approaches, are all treated under a single umbrella and numerous scenarios are studied and compared.
- The performance of the methodology proposed is verified using the typical IEEE 33-bus test system to detect its superiority for handling the problems and compare to other published approaches.

This paper is organized as follows: Section 2 illustrates the suggested methodology, formulates the load flow issue, and formulates the objective problems. Section 3 displays the simulated results and compares the proposed methodology to the other methods. The conclusions are driven in Section 4.

2. Materials and Methods

This section describes the detailed procedure adopted in this research. The relevant mathematical requirement for the implementation of the quasi-oppositional-based Aquila optimizer is presented.

2.1. The Aquila Optimizer

The Aquila optimizer was proposed in [2] to model the hunting behavior of Aquila birds. This entailed mimicking the agile nature, speed, large sturdy feet, and sharp talons of Aquila birds used in grabbing their prey during hunting. The four major types of prey hunting schemes devised by the Aquila birds were modeled, which are used depending on the hunting situation [2]. The first hunting scheme popularly used for hunting flying birds involves the Aquila bird flying at a high soar with a vertical stoop, in which success depends on the Aquila bird having a height advantage over its prey [2,3]. The second scheme involves the bird hovering around in a contour-like pattern at a low distance to the ground level and using a short glide-like attack on its prey, often used to hunt seabirds [2]. The third scheme involves a slow, low flight descent attack on prey, such as foxes, with slow speed and response. The last hunting scheme by the Aquila bird involves the direct grabbing of prey while walking on land. The intelligence of the Aquila bird gave it the flexibility of quickly varying its hunting scheme back and forth between the four hunting techniques. The success of this nature-inspired hunting process displayed by the Aquila bird serves as the motivation for the development of the Aquila optimizer algorithm for the purpose of optimization. Further details on the hunting movement pattern and theoretical mathematical formulation of it can be found in [2,3]. The optimization process of the AO begins by randomly generating the initial population, which comprises a set of randomly generated candidate solutions. A search scheme that tries to balance the exploration and exploitation of the search space towards converging at the best optimal solution using four strategies along the path repetition was adopted. These search strategies are expanded diversification, narrowed diversification, expanded intensification, and narrowed intensification.

The expanded exploration (x_1) utilizes the high soar with the vertical stoop hunting scheme. This involves the Aquila bird flying at a high soar to explore and determine its search space area while identifying its prey and choosing the best hunting area. This is modeled mathematically [2] as shown in Equation (1).

$$x_1^{(t+1)} = x_{best}^{(t)} \times \left(1 - \frac{t}{T}\right) + \left(x_M^{(t)} - rand \times X_{best}^{(t)}\right) \tag{1}$$

where the maximum number of iterations is represented as T and t is the current iteration whose next iteration solution $x_1^{(t+1)}$ is obtained through the first search in the population of the candidate solution (x_1), and $x_{best}^{(t)}$ denotes the best solution obtained until the tth iteration. The number of iterations is employed to regulate the scope of the exploration of the search space using $(1 - t/T)$ and $x_M^{(t)}$ is the mean value of the locations of connected tth iteration current solutions, computed using Equation (2) for a given population size N of dimension size D.

$$x_M^{(t)} = \frac{1}{N}\sum_{i=1}^{N} x_i(t), \forall_j = 1, 2, \cdots, D \tag{2}$$

The narrowed exploration (x_2) is characterized by the contour-like flight pattern with short glide attack in a narrowly explored search space for prey by the Aquila birds. This search method is used to obtain a solution $X_2^{(t+1)}$ for the next iteration of t as expressed mathematically in Equation (3).

$$x_2^{(t+1)} = x_{best}^{(t)} \times Levy(D) + x_R^{(t)} + (y - x) \times rand \tag{3}$$

where $Levy(D)$ is the levy flight distribution for the dimension space D and the random solution is $x_R^{(t)}$ obtained at the ith iteration in the range of $[1\ N]$. The levy flight distribution is computed based on a fixed constant value s usually set to a value of 0.01, as well as a number picked at random between 0 and 1 for the parameter u and v as expressed in Equation (4).

$$Levy(D) = s \times \frac{u \times \sigma}{|v|^{\frac{1}{\beta}}} \tag{4}$$

where the parameter σ is computed using Equation (5), with a fixed constant parameter β set to 1.5.

$$\sigma = \left(\frac{\Gamma(1+\beta) \times \sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left(\frac{1+\beta}{2}\right) \times \beta \times 2^{\left(\frac{\beta-1}{2}\right)}}\right) \tag{5}$$

The spiral shape in the search space as contained in Equation (3) is denoted by y and x and expressed in Equations (6) and (7), respectively.

$$y = r_1 + UD_1 \cos\left(-\omega D_1 + \frac{3\pi}{2}\right) \tag{6}$$

$$x = r_1 + UD_1 \sin\left(-\omega D_1 + \frac{3\pi}{2}\right) \tag{7}$$

where r_1 is assigned values between the range of 1–20 for a specified number of search iterations, and U and ω are assigned fixed values of 0.00565 and 0.005, respectively. D_1 is an integer number 1 to the length of the dimension D of the search space.

A slow, low flight descent attack is used by the Aquila bird to attack its prey, having accurately mapped out the prey area through the exploration process. This expanded exploitation x_3 is mathematically expressed as shown in Equation (8).

$$x_3^{(t+1)} = \left(x_{best}^{(t)} - x_M^{(t)}\right) \times \alpha - rand + ((ub - lb) \times rand + lb) \times \delta \tag{8}$$

where the next iteration of the t solution is denoted by $x_3^{(t+1)}$, the best solution obtained until the ith iteration is denoted by $x_{best}^{(t)}$, and $x_M^{(t)}$ is the mean value of the present solution computed based on Equation (2). The adjustment parameters α and δ are usually assigned a value of 0.1 each and the random number generated between 0 and 1 is assigned to rand. The upper and the lower bound are denoted by ub and lb, respectively.

The last hunting scheme used by the Aquila birds involves the direct grabbing of prey based on their random movement pattern while working on the land. This hunting scheme was used for the design of the narrowed exploitation (x_4) used to generate the tth iteration of the next solution $x_4^{(t+1)}$, as expressed in Equation (9). A balance between the search scheme was ensured by the introduction of a quality function, QF, as expressed in Equation (10).

$$x_4^{(t+1)} = QF \times x_{best}^{(t)} - \left(G_1 \times rand \times x_1^{(t)}\right) - G_2 \times Levy(D) + rand \times G_1 \tag{9}$$

where G_1 and G_2 denotes the Aquila prey tracking motion pattern and the flight slope to attack during elopement from the initial location to the last location, respectively, computed using Equations (11) and (12) where t is the current iteration and the maximum iterations are represented as T.

$$QF(t) = t^{\frac{2\times rand()-1}{(1-T)^2}} \tag{10}$$

$$G_1 = 2 \times rand - 1 \tag{11}$$

$$G_2 = 2 \times \left(1 - \frac{t}{T}\right) \tag{12}$$

The Aquila optimization process is summarized in the pseudocode in Algorithm 1 [15].

2.2. The Proposed Modified Aquila Optimizer

The search operator quasi-oppositional-based learning (QOBL), which was used to improve the initialization of the AO and local searching at the end of the AO algorithm, is described as follows.

2.2.1. Oppositional and Quasi-Oppositional-Based Learning

The OBL is a mathematical operator which was introduced for improving the performance of modern optimization algorithms. As proposed by Tizhoosh, the OBL is based on the idea that the probability of the opposite numbers to obtain a fitter solution is higher than random numbers [36]. Various studies have shown that OBL has improved the performance of numerous population-based optimization algorithms in terms of solution accuracy and convergence speed. Despite the encouraging results of OBL in various studies, the method had been modified to become quasi-oppositional-based learning [34]. The QOBL shows that using a quasi-opposite number is more effective than the opposite number in determining optimal results to an optimization problem. In QOBL, both the present nominee population, its opposite population, and its quasi-opposite population are used to form the fittest population. Assuming x is a 1-dimensional real number in the hyperspace. Let the opposite number be x^0 and the quasi-opposite number be x^{q0} of x, defined as:

$$x^0 = lb + ub - x \tag{13}$$

where $x \in \mathbb{R}$ and $x \in [lb, ub]$.

$$x^{q0} = rand\left(\frac{lb + ub}{2}, x^0\right) \tag{14}$$

Algorithm 1: Aquila Optimizer

```
Parameter Initialization (nPop, nVar, $\alpha$, $\delta$, maxIter, etc.)
Randomly generate initial positions
while (Termination is not met) do:
    Compute the fitness of initial positions
    Obtain the best individual with the best fitness values as $x_{best}(t)$
    for ($i$ = 1: nPop) do:
        The current solution's mean value if updated.
        Update $x$, $y$, $G_1$, $G_2$, $Levy(D)$
        if $t \leq \left(\frac{2}{3}\right) \times T$ then:
            if $rand \leq 0.5$ then:
                Update the current using Equation (1) %% Step 1: Expanded Diversification ($x_1$)
                if $f(x_1(t+1)) < f(x(t))$ then
                    $x(t) = x_1(t+1)$
                    if $f(x_1(t+1)) < f(x_{best}(t))$ then
                        $x_{best}(t) = x_1(t+1)$
                    end if
                end if
            else
                Update the current Solution Using Equation (5) % Step 2: Narrowed Diversification ($x_2$)

                if $f(x_2(t+1)) < f(x(t))$ then
                    $x(t) = x_2(t+1)$
                    if $f(x_2(t+1) < f(x_{best}(t))$ then
                        $x_{best} = x_2(t+1)$
                    end if
                end if
            end if
        else
            if $rand \geq 0.5$ then
                Use Equation (8) to update the current solution %% Expanded Intensification ($x_3$)
                if $f(x_3(t+1)) < f(x(t)$ then
                    $x(t) = x_3(t+1)$
                    if $f(x_3(t+1) < f(x_{best}(t))$ then
                        $x_{best}(t) = x_3(t+1)$
                    end if
                end if
            else
                Use Equation (9) to update the current solution %% Narrowed Exploitation
                if $f(x_4(t+1)) < f(x(t))$ then
                    $x(t) = x_4(t+1)$
                    if $f(x_4(t+1) < f(x_{best}(t))$ then
                        $x_{best}(t) < x_4(t+1)$
                    end if
                end if
            end if
        end if
    end for
end while
Return best solution ($x_{best}$)
```

2.2.2. Opposite Points and Quasi-Opposite Points

Assuming $x(x_1, x_2, \ldots, x_n)$ is a 1-dimensional point in the optimization hyperspace, then, the opposite point $x^0\left(x_1^0, x_2^0, \ldots, x_n^0\right)$ and quasi-opposite point $x^{q0}\left(x_1^{q0}, x_2^{q0}, \ldots, x_n^{q0}\right)$ can be computed by:

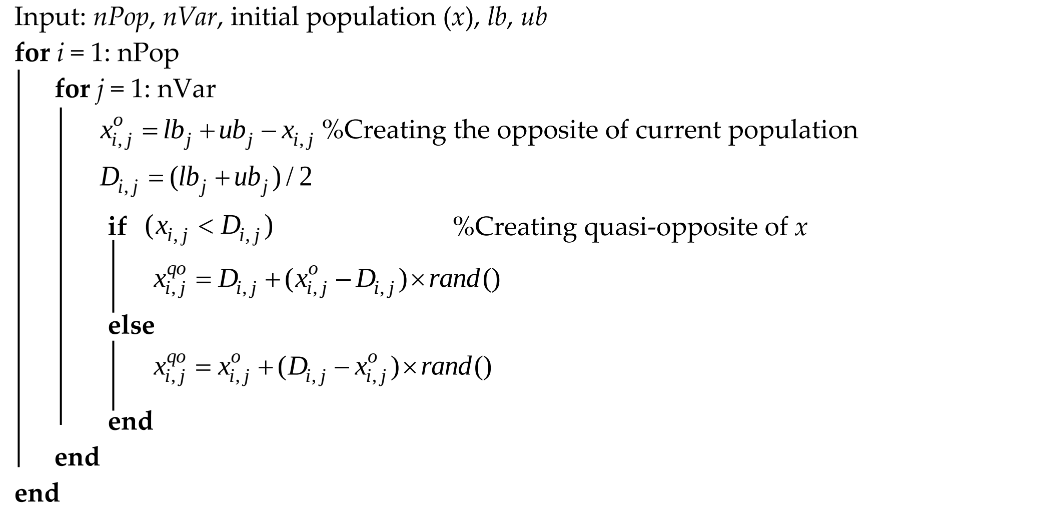
$$x_i^0 = lb_i + ub_i - x_i \tag{15}$$

where $x_i \in \mathbb{R}$ and $x_i \in [lb_i, ub_i]\ \forall_i\, 1,\, 2,\, \ldots, n$.

$$x_i^{q0} = rand\left(\frac{lb_i + ub_i}{2}, x_i^0\right) \tag{16}$$

In this paper, the QOBL was used in the population initialization of the original Aquila optimization algorithm. Concurrently, both the present population and the quasi-opposite population were utilized to generate a fitter population. This enables the AO to hurdle to a new nominee of the solution that has a fitter fitness value. The pseudocode in Algorithm 2 describes the procedures of the implemented quasi-oppositional-based learning.

Algorithm 2: Pseudocode of quasi-oppositional-based learning



```
Input: nPop, nVar, initial population (x), lb, ub
for i = 1: nPop
    for j = 1: nVar
        $x^{o}_{i,j} = lb_j + ub_j - x_{i,j}$ %Creating the opposite of current population
        $D_{i,j} = (lb_j + ub_j)/2$
        if ($x_{i,j} < D_{i,j}$)                %Creating quasi-opposite of x
            $x^{qo}_{i,j} = D_{i,j} + (x^{o}_{i,j} - D_{i,j}) \times rand()$
        else
            $x^{qo}_{i,j} = x^{o}_{i,j} + (D_{i,j} - x^{o}_{i,j}) \times rand()$
        end
    end
end
```

The QOBL strategy described in Algorithm 2 was incorporated into the original AO algorithm to establish the modified Aquila optimizer. After randomizing the initial population of the AO, the QOBL was implemented to obtain a new, improved initial population. Thereafter, each step of the AO was implemented. For each step of the AO, the QOBL was reimplemented to guarantee a better balance amid diversification and intensification and to ensure that the best solution was obtained. This process was repeated until termination criteria were met. The detailed procedure of the MAO is given in Algorithm 3.

The flowchart for implementing the quasi-oppositional-based Aquila optimizer is given in Figure 1 as follows.

Algorithm 3: Pseudocode of the MAO algorithm

```
Parameter initialization ($nPop$, $nVar$, $\alpha$, $\delta$, $maxIter$, etc.)
Randomly initialized population
Compute $x_{i,j}^{qo}$ using **Algorithm 2.**
**while** (termination is not met) **do:**
    Compute fitness value
    Obtain the best individual with the best fitness values as $X_{best}(t)$
    **for** (i =1: $nPop$) **do**:
        **if** $t \le \left(\frac{2}{3}\right) \times T$ **then:**
            **if** $rand() \le 0.5$ **then:**
                Perform expanded exploration
                Perform QOBL
            **else**
                Perform narrowed exploration
                Perform QOBL
            end if
        **else**
            **if** $rand() \le 0.5$ **then:**
                Perform expanded exploitation
                Perform QOBL
            **else**
                Perform narrowed exploitation
                Perform QOBL
            **end if**
        **end if**
    **end for**
**end while**
**return** best solution $X_{best}$
```

2.3. Methods of Evaluation

This section is further separated into two subsections. In the first subsection, the procedures used to evaluate the performance of the modified Aquila optimizer on twenty-three competition on evolutionary computation (CEC) benchmark functions in addition to the original Aquila optimizer are presented. The second subsection provides detailed procedures on the application of the modified Aquila optimizer on engineering problems.

2.3.1. The CEC Benchmark Functions' Description

Performance of the developed modified Aquila optimizer was evaluated using twenty-three benchmark functions of diverse properties. The twenty-three commonly used benchmark functions used in this paper were organized into three separate groups. The first group which comprises seven multidimensional, unimodal benchmark functions in the range of F1–F7 is given in Table 2. The second group given in Table 3 comprises six multidimensional multimodal benchmark functions in the range of F8–F13. The last group comprises benchmark functions of different dimensionality ranging from F14–F23 as shown in Table 4. Detailed information about the test functions can be found in [2].

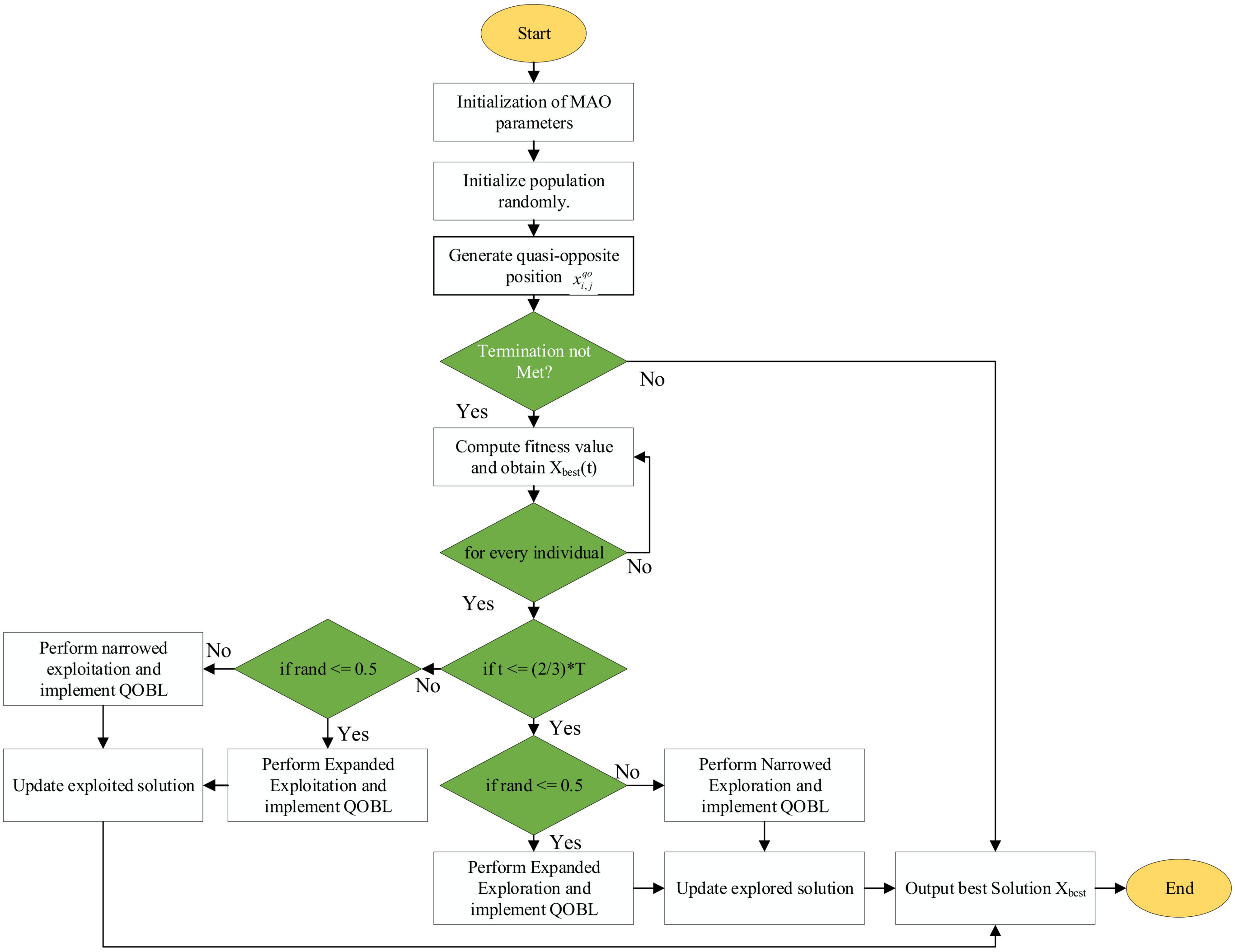

Start
Initialization of MAO parameters
Initialize population randomly.
Generate quasi-opposite position $x_{i,j}^{qo}$
Termination not Met?
No
Yes
Compute fitness value and obtain $X_{best}(t)$
for every individual
No
Yes
if t <= (2/3)*T
No
if rand <= 0.5
No
Perform narrowed exploitation and implement QOBL
Yes
Perform Expanded Exploitation and implement QOBL
Update exploited solution
Yes
if rand <= 0.5
No
Perform Narrowed Exploration and implement QOBL
Yes
Perform Expanded Exploration and implement QOBL
Update explored solution
Output best Solution X_{best}
End

Figure 1. Flowchart of modified Aquila optimizer.

Table 2. Unimodal benchmark functions.

Fn	Description	Dimension	Range	F_{min}
F1	$f(x) = \sum_{i=1}^{n} x_i^2$	10,50,100,500	[−100,100]	0
F2	$f(x) = \sum_{i=0}^{n}\lvert x_i\rvert + \prod_{i=0}^{n}\lvert x_i\rvert$	10,50,100,500	[−10,10]	0
F3	$f(x) = \sum_{i=1}^{d}\left(\sum_{j=1}^{i} x_j\right)^2$	10,50,100,500	[−100,100]	0
F4	$f(x) = \max_i\{\lvert x_i\rvert, 1 \le i \le n\}$	10,50,100,500	[−100,100]	0
F5	$f(x) = \sum_{i=1}^{n-1}\left[100\left(x_i^2 - x_{i+1}\right)^2 + (1 - x_i)^2\right]$	10,50,100,500	[−30,30]	0
F6	$f(x) = \sum_{i=1}^{n}([x_i + 0.5])^2$	10,50,100,500	[−100,100]	0
F7	$f(x) = \sum_{i=0}^{n} ix_i^4 + rand(0,1]$	10,50,100,500	[−128,128]	0

Table 3. Multimodal benchmark functions.

Fn	Description	Dimension	Range	$F_{\min}$
F8	$f(x) = \sum_{i=1}^{n}\left(-x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)\right)$	10,50,100,500	[−500,500]	$-418.9829n$
F9	$f(x) = \sum_{i=1}^{n}\left[x_i^2 - 10\cos(2\pi x_i) + 10\right]$	10,50,100,500	[−5.12,5.12]	0
F10	$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	10,50,100,500	[−32,32]	0
F11	$f(x) = 1 + \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)$	10,50,100,500	[−600,600]	0
F12	$f(x) = \frac{\pi}{n}\{10\sin(\pi y_1)\} + \sum_{i}^{n-1}(y_i - 1)^2\left[1 + 10\sin^2(\pi y_{i+1})\right] +$ $\sum_{i=1}^{n} u(x_i, 10, 100, 4),\ where,$ $\frac{x_i+1}{4}, u(x_i, a, k, m)\begin{cases} K(x_i - a)^m & if\, x_i > a \\ 0 & -a \le x_i \ge \\ K(-x_i - a)^m & -a \le x_i \end{cases}$	10,50,100,500	[−50,50]	0
F13	$f(x) = 0.1\left(\begin{array}{c}\sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2[1+ \\ \sin^2(3\pi x_i + 1)] + (x_n - 1)^2 + \sin^2(2\pi x_n)\end{array}\right) + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	10,50,100,500	[−50,50]	0

Table 4. Fixed-dimensional multimodal benchmark functions.

Fn	Description	Dimension	Range	$F_{\min}$
F14	$f(x) = \left(\frac{1}{500} + \sum_{j}^{25}\frac{i}{j + \sum_{i}^{25}(x_i - a_{ij})}\right)^{-1}$	2	[−65,65]	1
F15	$f(x) = \sum_{i=1}^{11}\left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right]^2$	4	[−5,5]	0.00030
F16	$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[−5,5]	−1.0316
F17	$f(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10$	2	[−5,5]	0.398
F18	$f(x) = \left[1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2\right] \times$ $\left[30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)\right]$	2	[−2,2]	3
F19	$f(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2\right)$	3	[−1,2]	−3.86
F20	$f(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2\right)$	6	[0,1]	−3.2
F21	$f(x) = -\sum_{i=1}^{5}\left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	4	[0,1]	−10.1532
F22	$f(x) = -\sum_{i=1}^{7}\left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	4	[0,1]	−10.4028
F23	$f(x) = -\sum_{i=1}^{10}\left[(X - a_i)(X - a_i)^T + c_i\right]^{-1}$	4	[0,1]	−10.5363

2.3.2. Load Flow Formulation for Optimal Distributed Generation Allocation

For load flow evaluations, the BFS approach was used [9]. Figure 2 is a scale model of a radial distribution network. It is assumed that the Nth line connects nodes "i" and "j". The BFS algorithm's implementation may be inferred in two stages using Kirchhoff's law. The convergence is faster if the largest discrepancy within bus voltages is less than 0.00001. As a result, the distribution grid's active and reactive power loss is easily calculated. These stages are as follows:

A. The power flow from the ith to jth nodes is expressed as the following equations depending on the direction of the backward sweep from the last node.

$$P_{ij} = P'_j + R_{ij}\frac{\left(P'^2_j + Q'^2_j\right)}{V_j^2} \tag{17}$$

$$Q_{ij} = Q'_j + X_{ij}\frac{\left(P'^2_j + Q'^2_j\right)}{V_j^2} \tag{18}$$

$$P'_j = P_{Lj} + P_{j,\,eff} \tag{19}$$

$$Q'_j = Q_{Lj} + Q_{j,\,eff} \tag{20}$$

where P_{Lj} and Q_{Lj} are connected loads at the jth node.

The forwarding sweep direction can be calculated using the voltage at the ith and jth nodes, and the nodal current via the branch-sample with an impedance $Z_{ij} = R_{ij} + jX_{ij}$ connected with ith and jth nodes can be calculated as follows:

$$I_{ij} = \frac{V_i\angle\delta_i - V_j\angle\delta_j}{R_{ij} + jX_{ij}} \tag{21}$$

$$or\ I_{ij} = \frac{P_i - jQ_i}{V_i\angle - \delta_i} \tag{22}$$

The voltage at the jth node is provided by Equations (21) and (22), as follows:

$$V_j = \left[V_i^2 - 2\times\left(P_iR_{ij} + jQ_iX_{ij}\right) + \left(R_{ij}^2 + X_{ij}^2\right)\times\frac{\left(P_i^2 + Q_i^2\right)}{V_i^2}\right]^{0.5} \tag{23}$$

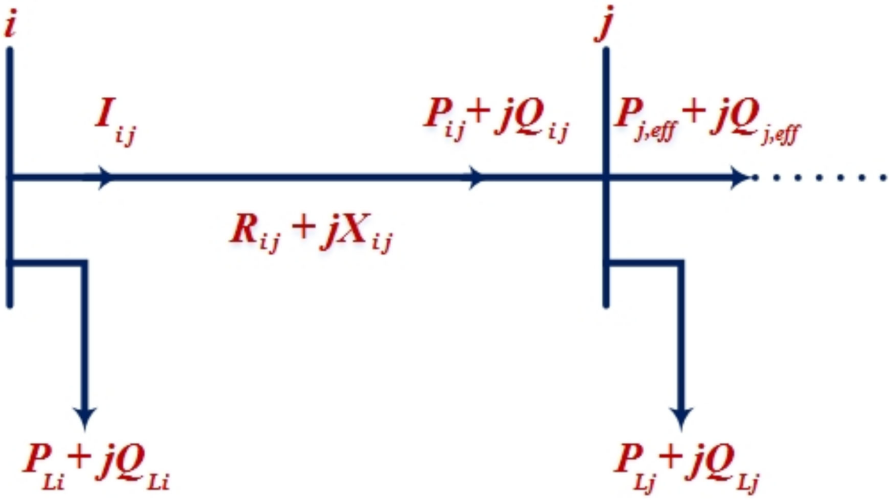


Figure 2. A radial distribution feeder.

2.3.3. Problem Formulation

The key issue here is determining the ideal sizes and locations of DGs to be installed on buses, which may be accomplished using the MAO algorithm. In load flow simulation, the DG units are considered as a negative PQ load. To begin, it is believed that all system buses, with the exemption of the substation bus, are candidates for DG placement. Using an objective function as in (24), the DG locations and sizes are considered as the control variables, and their optimal values are achieved through the proposed algorithm. The best selection of DG is found using an MO function, which is a weighted sum of a single-OF.

A. Objective function (OF)

The ODGA problem's goal or fitness function seeks to reduce power losses and voltage variations while adhering to equality and inequality restrictions [9]:

$$OF = min\{(W_1 \times P_L) + (W_2 \times VD)\} \tag{24}$$

where P_L is the real power loss, VD is the voltage deviation, and W_1 and W_2 are adaptive weights of power loss minimization and voltage deviation, respectively; $W_1 \geq 0$, $W_2 \geq 0$, and $W_1 + W_2 = 1$ for multi-objective function [26]. The specific variables, P_L, and VD, are as follows.

I. Real power loss (P_L)

The overall real power losses of the system are given by:

$$P_L = \sum\nolimits_{k=1}^{N_l} G_k \left(V_i^2 + V_j^2 - 2V_i V_j \cos\left(\delta_i - \delta_j\right) \right) \tag{25}$$

where N_l is the total of the transmission lines; G_k is the conductance of the line k; V_i and V_j are the line voltages at the sending and receiving ends, respectively; and δ_i and δ_j are voltages angles.

II. Voltage deviation (VD)

To guarantee quality service, the bus voltage magnitude is kept within the permitted limit. The voltage profile is enhanced by limiting the variance of the load bus voltage, as illustrated in Equation (26):

$$VD = \sum\nolimits_{i=1}^{n} |1 - V_i| \tag{26}$$

where n is the total system buses.

B. Constraints

The restrictions that ensure the system's exceptional performance are classified as technical and operational constraints, which determine the minimization of the objective function.

I. Operational constraints

They are known as equality constraints which formulate the active and reactive power balance by load flow equations, as given below:

$$P_{Gi} - P_{Di} - V_i \sum\nolimits_{j=1}^{n} V_j \left[G_{ij} \cos\theta_{ij} + B_{ij} \sin\theta_{ij} \right] = 0\,,\ \forall i\ \in\ n \tag{27}$$

$$Q_{Gi} - Q_{Di} - V_i \sum\nolimits_{j=1}^{n} V_j \left[G_{ij} \sin\theta_{ij} + B_{ij} \cos\theta_{ij} \right] = 0,\ \forall i\ \in\ n \tag{28}$$

where P_G and Q_G are the real power and reactive power generation, P_D and Q_D are the real and reactive load demand, G_{ij} and B_{ij} are the mutual conductance and susceptance between bus i and bus j, and θ_{ij} is the voltage angle difference between bus i and bus j.

II. Technical constraints

They are known as inequality constraints and can be divided into:

- Voltage constraints

$$\left| V_i^{min} \right| \leq |Vi| \leq \left| V_i^{max} \right|,\ \forall i\ \in\ n \tag{29}$$

- Current constraints

$$I_{li} \leq I_{li}^{rated},\ \forall i\ \in\ b \tag{30}$$

- DG size constraints

$$P_{DG}^{min} \leq P_{DG} \leq P_{DG}^{max} \tag{31}$$

$$Q_{DG}^{min} \leq Q_{DG} \leq Q_{DG}^{max} \tag{32}$$

- DG location constraints

$$2 \leq DG_{location} \leq n_{bus} \tag{33}$$

where $\left| V_i^{min} \right|$ and $\left| V_i^{max} \right|$ are the voltage boundaries at bus i, I_{li} is the line current flow of line i, and I_{li}^{rated} is the rated line current transferred.

3. Simulation Results

This section provides a detailed comparative study of the developed quasi-oppositional-based learning Aquila optimizer, called the MAO, and the original AO on twenty-three CEC benchmark functions and ODGA engineering problems. The results and discussion on the benchmark function, which comprises the solution accuracy and convergence analysis of both the MAO and AO on the CEC benchmark functions, is presented in Section 3.1, whereas the performance of both algorithms on the ODGA engineering problem is presented in Section 3.2. In both the CEC benchmark functions and the ODGA engineering problem, the MATLAB R2020b simulation platform was used on a personal computer running Windows 10, 64-bit operating system with Intel(R) Core (TM) i7-8550U CPU @ 1.80 GHz 1.99 GHz, 16 GB RAM.

3.1. Benchmark Functions Result Analysis

This section is sub sectioned into two parts. The first part describes the solution accuracy of the AO and MAO on the benchmark functions, whereas the convergence accuracy of the algorithms is described in the second subsection.

3.1.1. Solution Accuracy

The MAO and original AO algorithms were implemented on the twenty-three benchmark functions described in Section 2.3. The benchmark functions were grouped into three categories, namely, unimodal, multimodal, and fixed-dimensional benchmark functions. To ensure a good level of comparison, we retained the same algorithm parameters presented in the original AO algorithm. Both the MAO and AO were implemented for 200 iterations using 50 search agents. To examine the computational complexity of the MAO with respect to AO, we implemented the algorithms on the unimodal and multimodal functions by considering 10, 50, 100, and 500 dimensions. The results obtained for each dimension are presented in Tables 5–8. Table 5 gives the results obtained for both unimodal and multimodal benchmark functions considering 10 dimensions. The same results obtained for 50, 100, and 500 dimensions are presented in Tables 6–8, respectively.

Table 5. Results obtained for unimodal and multimodal functions with D = 10.

Unimodal Benchmark Functions with D = 10													
Fn	**AO**				**MAO**				**AO**		**MAO**		
	Best	**Avg**	**Worst**	**STD**	**Best**	**Avg**	**Worst**	**STD**	**nTI**	**Time(s)**	**nTI**	**Time(s)**	**AR**
F1	6.42×10^{-75}	1.43×10^{-62}	6.57×10^{-61}	9.28×10^{-62}	**5.00×10^{-148}**	3.30×10^{-123}	1.47×10^{-121}	2.09×10^{-122}	23	0.4238	10	**0.3118**	0.43
F2	2.39×10^{-37}	2.36×10^{-32}	6.60×10^{-31}	1.03×10^{-31}	**1.54×10^{-73}**	7.00×10^{-63}	$1.95 \times 10-61$	3.19×10^{-62}	60	**0.2963**	27	0.3685	0.45
F3	1.22×10^{-72}	$2.02 \times 10-56$	9.97×10^{-55}	1.41×10^{-55}	**6.66×10^{-146}**	2.05×10^{-121}	9.59×10^{-120}	1.36×10^{-120}	23	0.3866	6	**0.3581**	0.26
F4	1.24×10^{-38}	3.67×10^{-27}	1.84×10^{-25}	2.60×10^{-26}	**2.51×10^{-72}**	6.85×10^{-62}	3.42×10^{-60}	4.83×10^{-61}	60	**0.2912**	30	0.3630	0.50
F5	3.49×10^{-05}	0.001681	0.012943	0.002854	**1.12×10^{-05}**	2.21×10^{-02}	0.173778	3.34×10^{-02}	153	**0.3554**	197	0.3853	1.29
F6	2.89×10^{-08}	1.08×10^{-04}	1.20×10^{-03}	2.31×10^{-04}	**9.88×10^{-09}**	5.05×10^{-04}	3.60×10^{-03}	8.34×10^{-04}	137	**0.2986**	137	0.3655	1.00
F7	1.79×10^{-06}	1.33×10^{-04}	6.07×10^{-04}	1.34×10^{-04}	**1.27×10^{-06}**	1.57×10^{-04}	1.05×10^{-03}	2.07×10^{-04}	157	**0.2429**	16	0.3949	0.10
Multimodal Benchmark Functions with D = 10													
F8	$-6.78 \times 10^{+04}$	$-3.63 \times 10^{+04}$	$-1.76 \times 10^{+04}$	$1.39 \times 10^{+04}$	**$-4.77 \times 10^{+04}$**	$-1.96 \times 10^{+04}$	$-9.27 \times 10^{+03}$	$9.17 \times 10^{+03}$	137	**0.273260**	32	0.344875	0.23
F9	***$0.00 \times 10^{+00}$***	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	***$0.00 \times 10^{+00}$***	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	27	**0.270444**	10	0.396144	0.37
F10	***8.88×10^{-16}***	8.88×10^{-16}	8.88×10^{-16}	$0.00 \times 10^{+00}$	***8.88×10^{-16}***	8.88×10^{-16}	8.88×10^{-16}	$0.00 \times 10^{+00}$	64	**0.251370**	26	0.384963	0.41
F11	***$0.00 \times 10^{+00}$***	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	***$0.00 \times 10^{+00}$***	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	30	**0.283189**	10	0.434610	0.33
F12	**2.78×10^{-09}**	1.27×10^{-05}	1.17×10^{-04}	2.16×10^{-05}	8.86×10^{-09}	3.94×10^{-06}	6.83×10^{-05}	1.03×10^{-05}	191	**0.535182**	138	0.642867	0.72
F13	7.86×10^{-09}	5.23×10^{-05}	5.73×10^{-04}	9.84×10^{-05}	**4.70×10^{-09}**	4.82×10^{-05}	3.70×10^{-04}	8.10×10^{-05}	148	0.593632	152	**0.590624**	1.03

Table 6. Results obtained for unimodal and multimodal functions with D = 50.

	Unimodal Benchmark Functions with D = 50												
Fn	**AO**				**MAO**				**AO**		**MAO**		
	Best	**Avg**	**Worst**	**STD**	**Best**	**Avg**	**Worst**	**STD**	**nTI**	**Time(s)**	**nTI**	**Time(s)**	**AR**
F1	7.91×10^{-76}	5.27×10^{-48}	1.05×10^{-49}	7.45×10^{-49}	$\mathbf{1.24 \times 10^{-157}}$	4.34×10^{-131}	2.16×10^{-129}	3.06×10^{-130}	20	**0.346058**	7	0.446555	0.35
F2	5.83×10^{-38}	1.84×10^{-29}	4.30×10^{-31}	2.61×10^{-30}	$\mathbf{5.38 \times 10^{-76}}$	1.29×10^{-66}	5.70×10^{-65}	8.09×10^{-66}	14	**0.321249**	8	0.510822	0.57
F3	1.77×10^{-70}	8.58×10^{-52}	3.00×10^{-53}	1.39×10^{-52}	$\mathbf{2.29 \times 10^{-155}}$	2.10×10^{-130}	9.62×10^{-129}	1.36×10^{-129}	12	0.782950	2	**0.716966**	0.166
F4	1.08×10^{-37}	5.06×10^{-28}	1.02×10^{-29}	7.15×10^{-29}	$\mathbf{4.39 \times 10^{-78}}$	4.29×10^{-67}	2.02×10^{-65}	2.86×10^{-66}	67	**0.314908**	23	0.407632	0.34
F5	$\mathbf{5.17 \times 10^{-07}}$	0.087	0.0128	0.0209	2.01×10^{-05}	0.1339	0.0134	2.78×10^{-02}	10	**0.323443**	2	0.497856	0.20
F6	$\mathbf{4.88 \times 10^{-08}}$	2.20×10^{-03}	3.57×10^{-04}	5.51×10^{-04}	5.45×10^{-07}	4.72×10^{-04}	0.0045	8.44×10^{-04}	11	**0.415805**	2	0.563604	0.18
F7	$\mathbf{7.66 \times 10^{-07}}$	7.85×10^{-04}	1.15×10^{-04}	1.31×10^{-04}	1.87×10^{-06}	1.38×10^{-04}	4.21×10^{-04}	1.01×10^{-04}	113	0.555977	136	**0.547667**	1.20
	Multimodal Benchmark Functions with D = 50												
F8	$\mathbf{-2.09 \times 10^{+04}}$	$-4.01 \times 10^{+03}$	$-8.35 \times 10^{+03}$	$5.01 \times 10^{+03}$	$-4.98 \times 10^{+03}$	$-2.80 \times 10^{+03}$	$-3.75 \times 10^{+03}$	483.5646	172	**0.415364**	183	0.674151	1.06
F9	$\boldsymbol{0.00 \times 10^{+00}}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$\boldsymbol{0.00 \times 10^{+00}}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	30	**0.432322**	4	0.482665	0.13
F10	$\boldsymbol{8.88 \times 10^{-16}}$	8.88×10^{-16}	8.88×10^{-16}	$0.00 \times 10^{+00}$	$\boldsymbol{8.88 \times 10^{-16}}$	8.88×10^{-16}	8.88×10^{-16}	$0.00 \times 10^{+00}$	74	**0.416677**	5	0.455358	0.01
F11	$\boldsymbol{0.00 \times 10^{+00}}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$\boldsymbol{0.00 \times 10^{+00}}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	24	**0.374171**	21	0.490126	0.88
F12	6.33×10^{-08}	2.71×10^{-06}	1.60×10^{-05}	3.32×10^{-06}	$\mathbf{5.67 \times 10^{-08}}$	5.92×10^{-06}	5.92×10^{-05}	1.05×10^{-05}	7	0.990779	181	**0.878485**	25.86
F13	3.94×10^{-07}	2.85×10^{-04}	4.41×10^{-05}	6.44×10^{-05}	$\mathbf{3.35 \times 10^{-08}}$	1.04×10^{-04}	1.30×10^{-03}	2.60×10^{-04}	10	1.027384	196	**0.878522**	19.60

Table 7. Results obtained for unimodal and multimodal functions with D = 100.

	Unimodal Benchmark Functions with D = 100												
Fn	**AO**				**MAO**				**AO**		**MAO**		
	Best	**Avg**	**Worst**	**STD**	**Best**	**Avg**	**Worst**	**STD**	**nTI**	**Time(s)**	**nTI**	**Time(s)**	**AR**
F1	1.25×10^{-77}	4.93×10^{-59}	1.07×10^{-60}	6.98×10^{-60}	$\mathbf{7.17 \times 10^{-152}}$	2.84×10^{-128}	5.78×10^{-130}	4.02×10^{-129}	22	**0.284541**	8	0.538532	0.36
F2	2.24×10^{-39}	1.42×10^{-24}	3.52×10^{-26}	2.06×10^{-25}	$\mathbf{1.51 \times 10^{-76}}$	2.43×10^{-64}	5.94×10^{-66}	3.49×10^{-65}	11	**0.377031**	2	0.587254	0.18
F3	1.46×10^{-69}	5.55×10^{-49}	1.11×10^{-50}	7.85×10^{-50}	$\mathbf{1.49 \times 10^{-149}}$	3.07×10^{-128}	1.20×10^{-129}	5.90×10^{-129}	30	1.523620	11	**1.433964**	0.37
F4	1.46×10^{-69}	5.55×10^{-49}	1.11×10^{-50}	7.85×10^{-50}	$\mathbf{1.54 \times 10^{-80}}$	7.08×10^{-67}	1.55×10^{-68}	1.00×10^{-67}	59	**0.370328**	23	0.612341	0.39
F5	1.97×10^{-04}	0.1103	0.0219	0.0315	$\mathbf{6.17 \times 10^{-08}}$	0.091609	0.015053	2.28×10^{-02}	12	**0.397905**	159	0.722258	13.25
F6	2.91×10^{-06}	7.19×10^{-04}	1.97×10^{-04}	1.61×10^{-04}	$\mathbf{1.94 \times 10^{-06}}$	1.14×10^{-02}	8.39×10^{-04}	1.77×10^{-03}	161	**0.357496**	185	0.627708	1.15
F7	3.79×10^{-06}	1.24×10^{-04}	4.57×10^{-04}	1.22×10^{-04}	$\mathbf{2.26 \times 10^{-06}}$	4.81×10^{-04}	1.37×10^{-04}	1.35×10^{-04}	176	**0.786597**	26	1.063293	0.15
	Multimodal Benchmark Functions with D=100												
F8	$-6.77 \times 10^{+04}$	$-2.08 \times 10^{+04}$	$-3.88 \times 10^{+04}$	$1.28 \times 10^{+04}$	$\mathbf{-1.87 \times 10^{+04}}$	$-5.80 \times 10^{+03}$	$-3.89 \times 10^{+03}$	$2.20 \times 10^{+03}$	177	**0.522011**	96	0.899167	0.54
F9	$\boldsymbol{0.00 \times 10^{+00}}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$\boldsymbol{0.00 \times 10^{+00}}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	21	0.541459	9	**0.528925**	0.43
F10	$\boldsymbol{8.88 \times 10^{-16}}$	8.88×10^{-16}	8.88×10^{-16}	$0.00 \times 10^{+00}$	$\boldsymbol{8.88 \times 10^{-16}}$	8.88×10^{-16}	8.88×10^{-16}	$0.00 \times 10^{+00}$	60	**0.579987**	24	0.647832	0.40
F11	$\boldsymbol{0.00 \times 10^{+00}}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$\boldsymbol{0.00 \times 10^{+00}}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	19	**0.567551**	7	0.679634	0.37
F12	$\mathbf{2.12 \times 10^{-09}}$	1.28×10^{-05}	1.89×10^{-06}	3.20×10^{-06}	3.31×10^{-08}	3.05×10^{-05}	4.56×10^{-06}	5.88×10^{-06}	198	1.771936	137	**1.669248**	0.69
F13	5.44×10^{-08}	4.67×10^{-04}	5.33×10^{-05}	8.09×10^{-05}	$\mathbf{2.05 \times 10^{-09}}$	7.29×10^{-04}	9.69×10^{-05}	1.59×10^{-04}	165	1.525974	151	**1.500534**	0.92

Table 8. Results obtained for unimodal and multimodal functions with D = 500.

	Unimodal Benchmark Functions with D = 500												
Fn	**AO**				**MAO**				**AO**		**MAO**		
	Best	**Avg**	**Worst**	**STD**	**Best**	**Avg**	**Worst**	**STD**	**nTI**	**Time(s)**	**nTI**	**Time(s)**	**AR**
F1	1.50×10^{-74}	8.20×10^{-48}	1.64×10^{-49}	1.16×10^{-48}	$\mathbf{9.80 \times 10^{-151}}$	1.25×10^{-132}	5.88×10^{-131}	8.32×10^{-132}	17	1.188284	7	**1.092645**	0.41
F2	2.69×10^{-36}	1.27×10^{-30}	7.35×10^{-32}	2.12×10^{-31}	$\mathbf{2.17 \times 10^{-77}}$	6.73×10^{-66}	3.17×10^{-64}	4.48×10^{-65}	15	1.264532	6	**1.253314**	0.40
F3	3.17×10^{-68}	5.83×10^{-44}	1.17×10^{-45}	8.25×10^{-45}	$\mathbf{8.20 \times 10^{-146}}$	8.36×10^{-126}	4.02×10^{-124}	5.69×10^{-125}	33	9.089672	9	**7.793591**	0.27
F4	7.47×10^{-37}	1.29×10^{-25}	2.60×10^{-27}	1.83×10^{-26}	$\mathbf{3.67 \times 10^{-77}}$	5.78×10^{-68}	2.83×10^{-66}	4.00×10^{-67}	60	**1.323380**	22	1.408070	0.37
F5	6.09×10^{-04}	4.674197	0.340405	0.772287	$\mathbf{4.44 \times 10^{-05}}$	1.407068	0.142883	2.35×10^{-01}	136	1.265221	7	**1.233198**	0.05
F6	1.03×10^{-05}	2.00×10^{-02}	1.76×10^{-03}	3.05×10^{-03}	$\mathbf{2.04 \times 10^{-06}}$	3.31×10^{-03}	0.025907	5.12×10^{-03}	20	1.377631	154	**1.166943**	7.70
F7	1.62×10^{-06}	4.57×10^{-04}	9.92×10^{-05}	9.26×10^{-05}	$\mathbf{1.08 \times 10^{-06}}$	1.43×10^{-04}	4.83×10^{-04}	1.16×10^{-04}	63	1.974413	59	**1.966353**	0.94
	Multimodal Benchmark Functions with D = 500												
F8	$-7.43 \times 10^{+04}$	$-2.14 \times 10^{+04}$	$-3.75 \times 10^{+04}$	$1.09 \times 10^{+04}$	$\mathbf{-3.45 \times 10^{+04}}$	$-1.56 \times 10^{+04}$	$-7.98 \times 10^{+03}$	$5.55 \times 10^{+03}$	198	1.430414	7	**1.418343**	0.04
F9	$\boldsymbol{0.00 \times 10^{+00}}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$\boldsymbol{0.00 \times 10^{+00}}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	25	**1.226005**	10	1.244486	0.40
F10	$\boldsymbol{8.88 \times 10^{-16}}$	8.88×10^{-16}	8.88×10^{-16}	$0.00 \times 10^{+00}$	$\boldsymbol{8.88 \times 10^{-16}}$	8.88×10^{-16}	8.88×10^{-16}	$0.00 \times 10^{+00}$	73	**1.217467**	24	1.266437	0.33
F11	$\boldsymbol{0.00 \times 10^{+00}}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$\boldsymbol{0.00 \times 10^{+00}}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	$0.00 \times 10^{+00}$	24	1.287624	7	**1.333875**	0.29
F12	2.07×10^{-09}	1.54×10^{-05}	1.49×10^{-06}	3.24×10^{-06}	$\mathbf{4.14 \times 10^{-10}}$	1.49×10^{-06}	$1.27 \times 10–^{05}$	2.58×10^{-6}	137	7.622310	17	**4.425322**	0.12
F13	4.12×10^{-06}	4.14×10^{-03}	9.53×10^{-04}	1.16×10^{-03}	$\mathbf{7.21{\times}10^{-07}}$	5.02×10^{-04}	4.02×10^{-03}	8.25×10^{-04}	14	4.602769	144	**4.367627**	10.28

For the fixed-dimensional CEC functions, the results obtained are given in Table 9. In each CEC benchmark function, the best, average, worst, and standard deviation of the computations after 50 independent runs are presented. The simulation time taken by each algorithm to obtain their respective optimal results is also given in each table as nTI. From the tables, the values in bold give the best results obtained by either the MAO or AO. The values in bold italics are used to represent a situation where both the MAO and AO obtained the same results for any of the benchmark functions.

Table 9. Results obtained for fixed-dimensional benchmark functions.

Fn	AO				MAO				AO		MAO		
	Best	**Avg**	**Worst**	**STD**	**Best**	**Avg**	**Worst**	**STD**	**nTI**	**Time(s)**	**nTI**	**Time(s)**	**AR**
F14	0.998004	1.411677	10.76318	1.467424	**0.998186**	7.301936	12.67051	1.996737	7	1.327123	56	**1.054049**	8.00
F15	$\mathbf{3.39 \times 10^{-04}}$	5.44×10^{-04}	1.13×10^{-03}	1.79×10^{-04}	3.45×10^{-04}	9.51×10^{-04}	1.68×10^{-03}	3.87×10^{-04}	6	**0.382481**	6	0.399001	1.00
F16	***−1.0316***	−1.0315	−1.03063	1.77×10^{-04}	***−1.0316***	−1.03083	−1.02881	6.33×10^{-04}	140	0.379773	139	**0.379187**	0.99
F17	0.397887	0.397935	0.398303	6.46×10^{-05}	**0.398349**	0.421134	0.479731	1.96×10^{-02}	185	**0.321453**	70	0.407058	0.38
F18	**3.00012**	3.038106	3.007461	0.00734	3.020373	3.969685	8.562764	1.156593	194	**0.306667**	88	0.376014	0.45
F19	−3.86274	−3.86073	−3.85411	0.002118	**−3.86195**	−3.7668	−3.637	0.055039	178	**0.333752**	75	0.393799	0.42
F20	−3.2984	−3.1204	−2.8735	0.0972	**−3.2097**	−2.7442	−2.0352	0.2411	200	**0.380213**	153	0.449120	0.76
F21	**−10.1532**	−10.1402	−9.9756	0.0263	−10.1338	−9.5941	−7.9488	0.4674	3	0.442247	187	**0.435953**	62.33
F22	**−10.4028**	−10.3868	−10.3251	0.0186	−10.4022	−9.7427	−7.4474	6275	5	**0.401554**	192	0.469871	38.40
F23	**−10.5358**	−10.5163	−10.4322	0.0257	−10.5303	−9.84755	−8.51132	0.512047	3	0.524773	197	**0.479695**	65.67

From the result tables, it can be observed that for the set of unimodal benchmark functions (F1–F7) the modified optimizer, MAO, demonstrated its superiority over the standard AO in all the functions when ten-dimensional variables were considered. For the set of multimodal functions (F8–F13), the MAO also showed superior performance by obtaining the best results in F8 and F13 and jointly obtaining the best results in F9–F11. The AO only performed better in this scenario in F12. When fifty-dimensional variables were considered, the MAO also showed superior performance over the AO in F1–F4 unimodal benchmark functions, whereas the AO marginally performed better in F5–F8 unimodal functions. Considering the same dimensional variables for the multimodal benchmark functions, the MAO obtained the best results in F12 and F13 whereas the AO obtained the best results only in F8. Both algorithms, however, obtained the same results in F9–F11. For the one-hundred-dimensional variables, the MAO showed very good performance in all the unimodal benchmark functions by performing better than the AO in all the seven benchmark functions. For the multimodal functions of these dimensional variables, the MAO also showed better performance over the AO by obtaining the best results in F8 and F13, where the AO obtained the best result only in F12. For this dimensional variable, both algorithms obtained the same results in F9–F11. Similarly, for 500-dimensional variables, the MAO also demonstrated its superiority over the AO by obtaining the best results in all the unimodal functions. For the multimodal functions, the MAO also obtained the best results in F8 and F11–F12, whereas both algorithms jointly obtained the same results in F9–F11. These results are a demonstration that irrespective of the size of the search space, the modified optimizer maintained a superior performance over the original AO on unimodal and multimodal functions.

When applied to the CEC benchmark functions of fixed dimension in the range of F14–F23, the modified optimizer, MAO, obtained the best results in F14, F17, and F19–F20. For these function groups, the original AO obtained the best result in the F15, F18, and F21–F23 benchmark functions. However, both the MAO and AO obtained the same results in the F16 benchmark function. This is also a demonstration that the MAO can effectively solve the benchmark functions of a fixed dimension in comparison with the standard AO.

To examine the robustness and superiority of the results of any of the algorithms over the other, the Wilcoxon rank-sum nonparametric study was conducted using the 50-independent runs performed in this paper. This test was conducted as a benchmark where none of the algorithms obtained exactly the same results or exactly the same global solution. This is because the nonparametric test for two equal samples of the same size is not a real or complex number. Thus, this paper carried out the Wilcoxon rank-sum test at a significant probability level of 5% (p-value = 0.05) for the benchmark function results of 50 dimensions and fixed dimensions as shown in Table 10.

Table 10. Wilcoxon rank-sum test for D = 50.

Fn	p-Value	h-Value
F1	0.2009	0
F2	7.0661×10^{-18}	1
F3	1.3695×10^{-13}	1
F4	7.0661×10^{-18}	1
F5	5.1090×10^{-06}	1
F6	2.8118×10^{-08}	1
F7	0.8388	0
F8	1.9072×10^{-16}	1
F12	0.5147	0
F13	0.1152	0
F14	1.1485×10^{-12}	1
F15	1.5445×10^{-10}	1
F16	3.2423×10^{-04}	1
F17	1.0754×10^{-17}	1
F18	1.0703×10^{-16}	1
F19	7.4858×10^{-16}	1
F20	2.5415×10^{-16}	1
F21	4.2058×10^{-17}	1
F22	5.3319×10^{-16}	1
F23	3.5254×10^{-17}	1

From Table 10, when a p-value ≤ 0.05 and corresponded to a null hypothesis, where h-value = 1, then the difference in solution sets obtained by either of the algorithms over the other was significant. Similarly, if the p-value > 5 and the h-value = 0, the solution quality obtained by either of the algorithms over the other was not significant. From the table, it can be observed that for functions where none of the algorithms obtained the same results, the MAO dominate AO, especially for F2–F6, F14, F16–F17, and F20. Similarly, F8, F15, F21, F22, and F23 showed that the results obtained by the AO dominated MAO for these functions. However, for F1, F7, F12 and F13, results showed nonsignificant performance of any of the algorithms over the other by obtaining an h-value of 0.

3.1.2. Convergence Study

To effectively analyze the convergence of both the MAO and AO, and to avoid unnecessary repetition of convergence, the convergence characteristics of both algorithms on ten- and fifty-dimensional variables are considered in this paper. Since the benchmark functions considered in this paper are organized into three different categories, we present the convergence plots according to these categories. For example, Figure 3 shows the superimposed convergence plots of the MAO and AO on unimodal functions considering ten-dimensional variables, whereas the convergence plots for both algorithms on the multimodal functions for the same dimensional variables are shown in Figure 4. For fifty-dimensional variables, the convergence characteristics for both unimodal and multimodal functions are presented in Figures 5 and 6, respectively. Finally, the convergence for the fixed-dimensional variables is given in Figure 7.

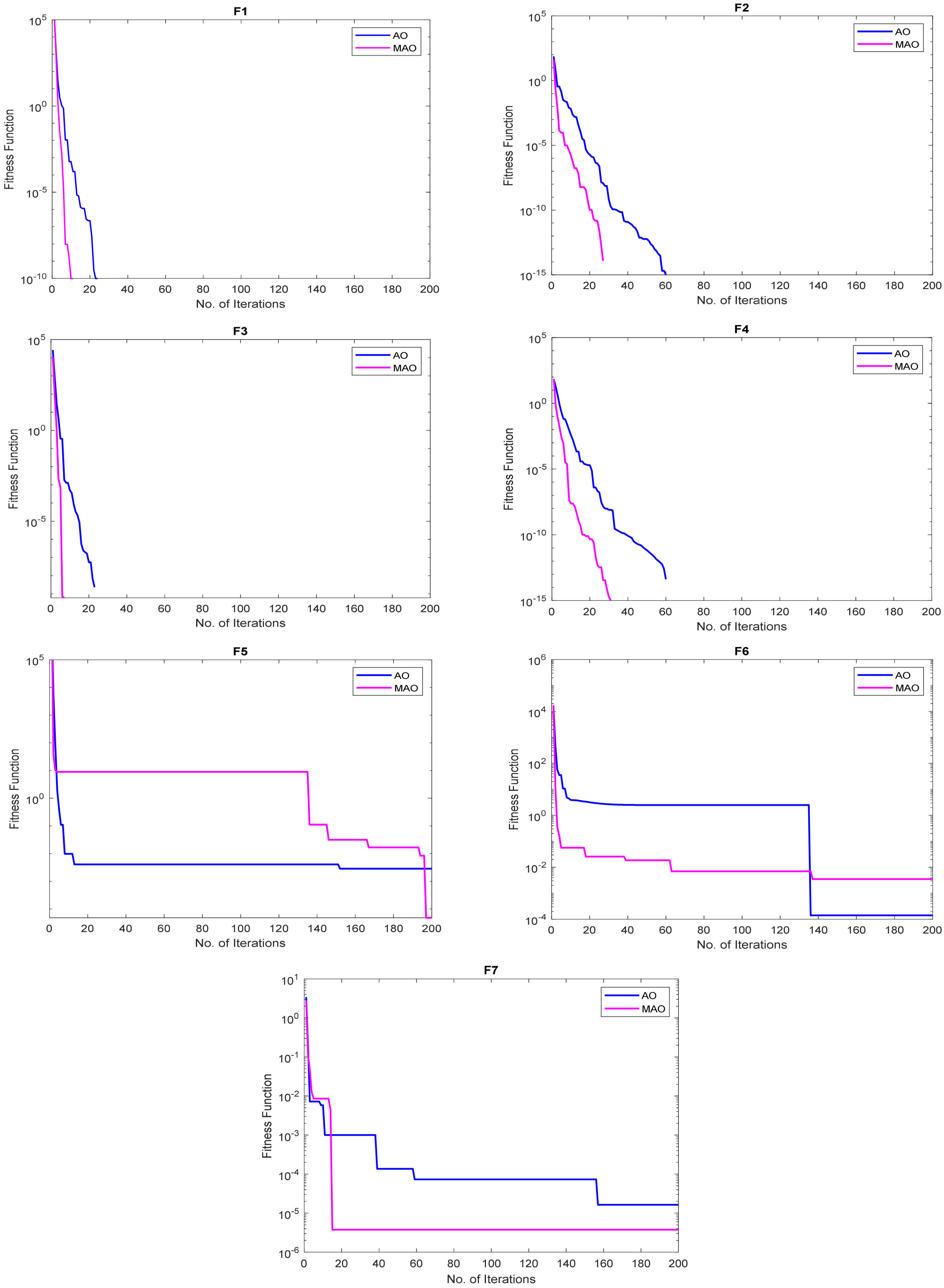


Figure 3. Convergence analysis of unimodal benchmark functions with D = 10.

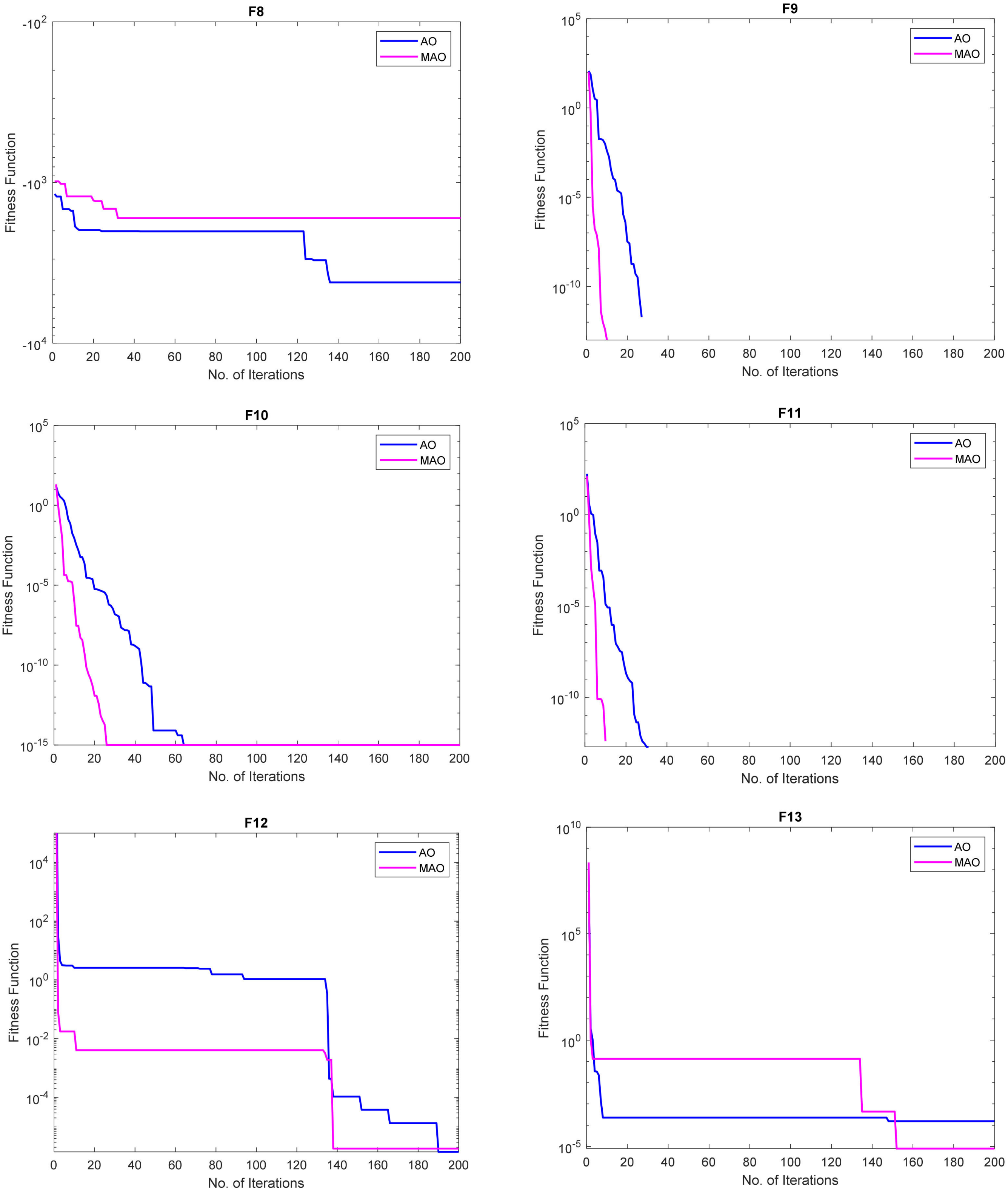


Figure 4. Convergence analysis of multimodal benchmark functions with D = 10.

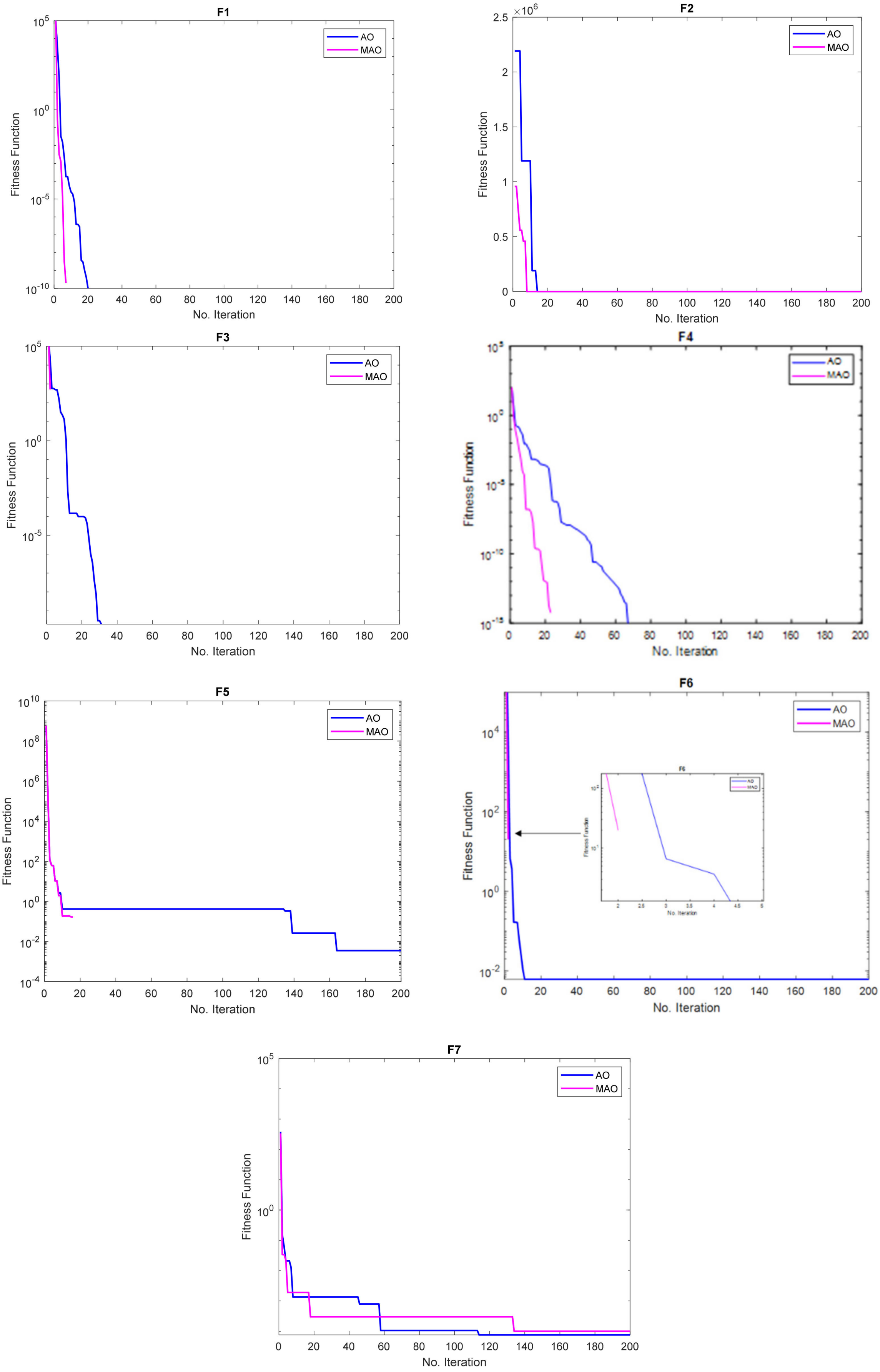


Figure 5. Convergence analysis of unimodal benchmark functions with D = 50.

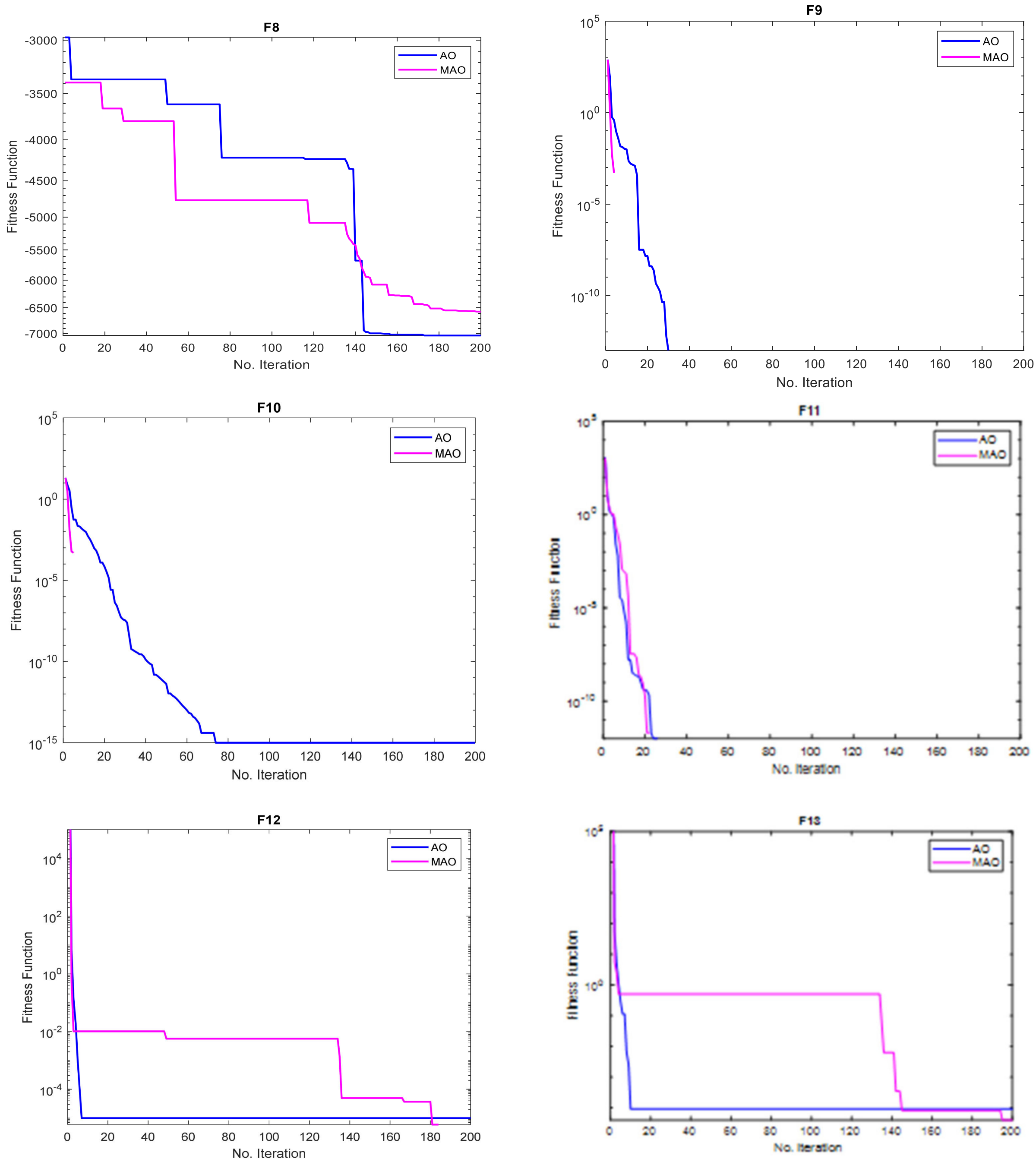


Figure 6. Convergence analysis of multimodal benchmark functions with D = 50.

Looking at the convergence plots given in Figures 3–7, it can be observed that the modified optimizer has faster convergence in almost all the benchmark functions and all the benchmark function classifications and dimensionalities, as considered in comparison with the original optimizer. For example, in Figure 3, the MAO converges faster in F1, F2, F3, F4, and F7, which constitute 71.43% of the total unimodal benchmark functions with 10 dimensions. In Figure 4, which shows the convergence study of multimodal functions with 10-dimensional variables, the MAO obtained the best convergence in F8, F9, F10, F11, and F12, constituting 83.33%, whereas the AO converged faster only in F13. When the number of dimensional variables was increased to 50 to examine robustness of the MAO on problems with a larger hyperspace, the MAO maintained its superior performance over the AO. In this context, the MAO obtained the fastest convergence in F1–F6, constituting 85.75% of the benchmark unimodal functions as given in Figure 5. However, for multimodal functions with 50-dimensional variables shown in Figure 6, the MAO obtained the fastest convergence in F9, F10, and F11 whereas the AO also obtained the best convergence in F8, F12, and F13.

This superior convergence of the MAO is further demonstrated by the acceleration rate computed in Tables 5–9. From these tables, it can be seen that nearly all the AR values computed for the dimension whose convergence was generated showed a value of less than 1 (i.e., AR < 1), except in a few cases where AR > 1 was observed, such as F5 and F13 in Table 5, F7, F8, F12, and F13 in Table 6, F5 and F6 in Table 5, and F6 and F13 in Table 8. For the fixed-dimensional benchmark functions, the MAO also had better convergence in most of the benchmark functions except in F14 and F21–F23. This superior convergence of the MAO over AO was also a clear demonstration of the effectiveness of the QOBL operator as an effective mechanism for population initialization and local search operation. Though the MAO showed superior performance considering solution accuracy and convergence speed, the original AO required less computational resources. The AO recorded a lesser simulation time to finish all the specified iterations, unlike the MAO. However, it was observed that as the dimension of the benchmark functions was increased up to D = 500, the MAO tended to obtain its optimized results in lesser computational time compared to the AO algorithm.

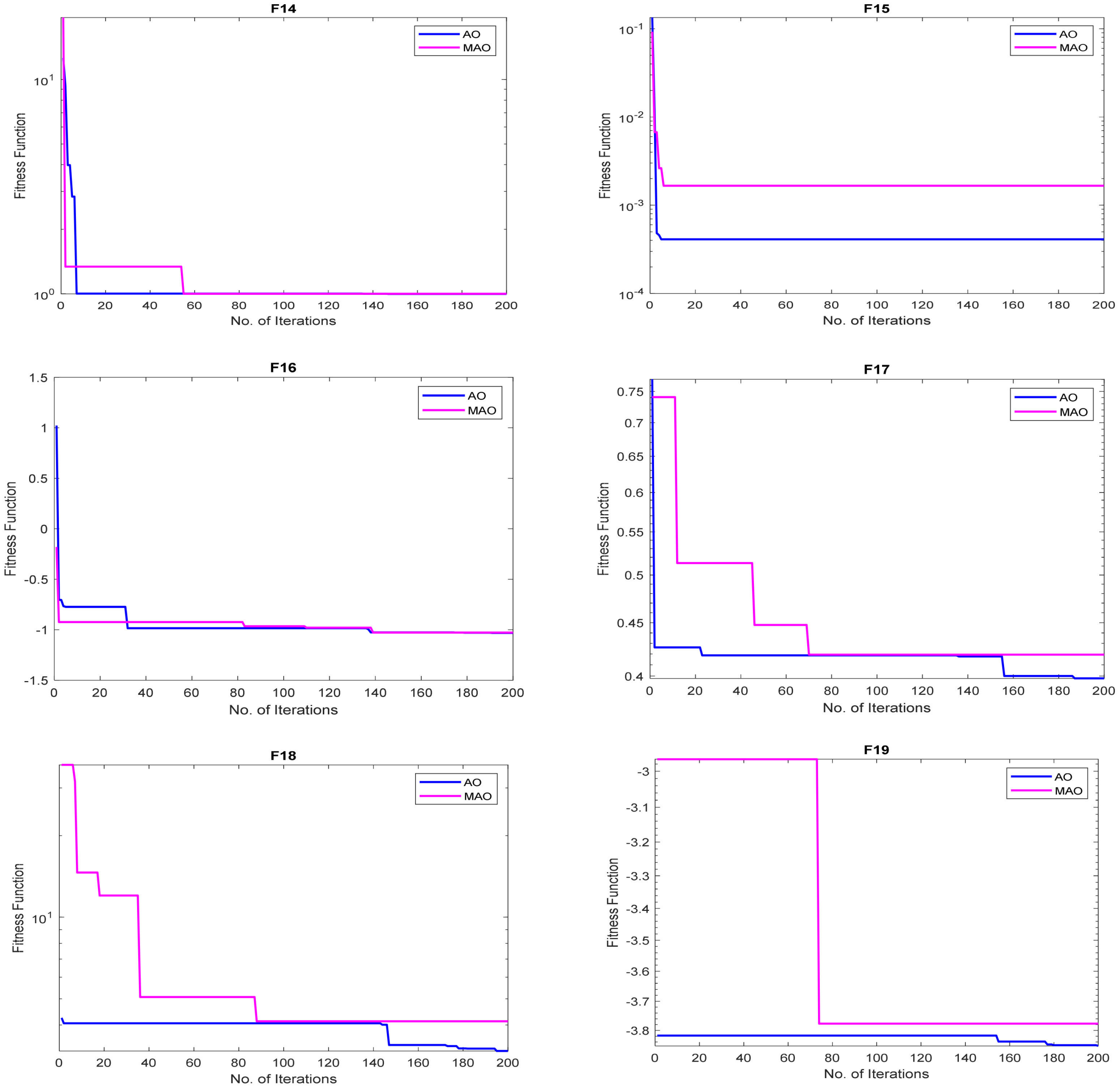


Figure 7. *Cont.*

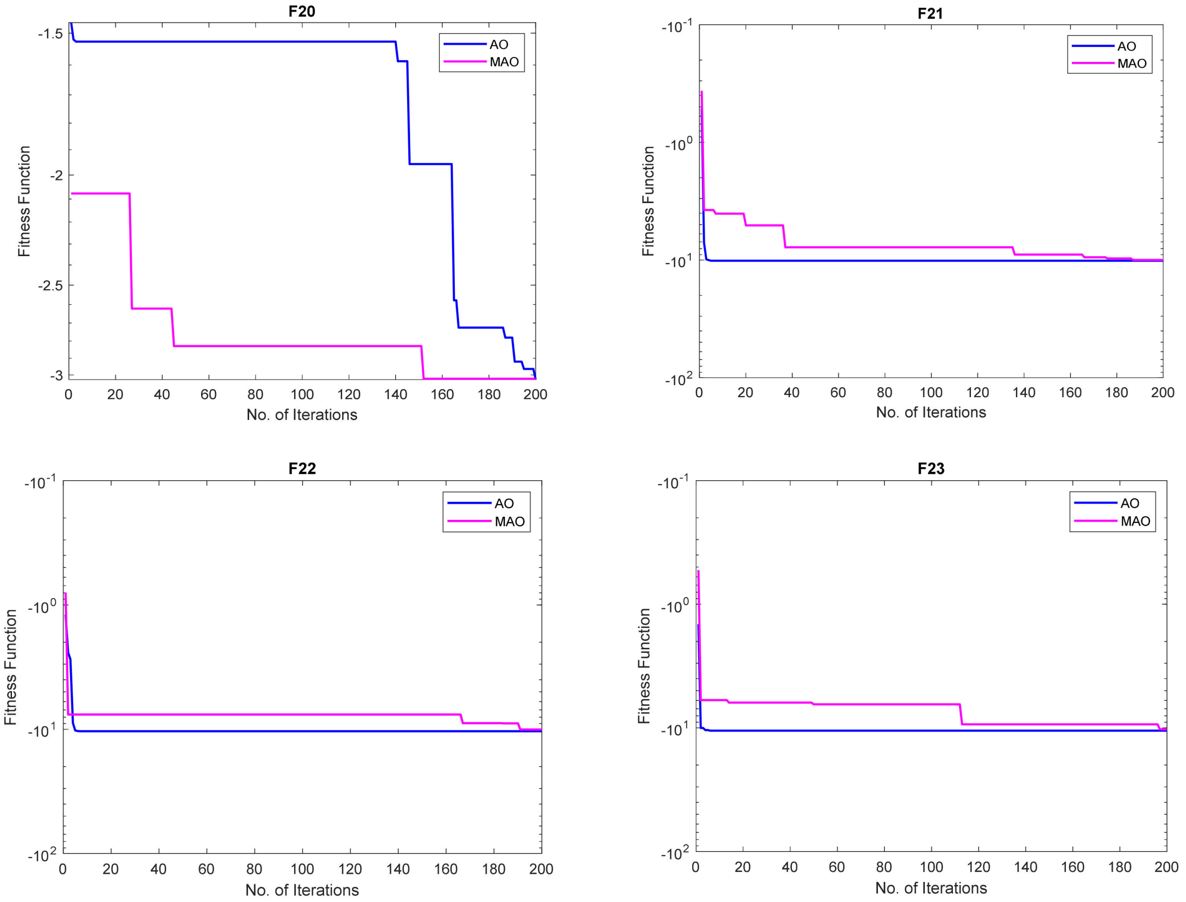


Figure 7. Convergence analysis of fixed-dimensional benchmark functions.

Note: an AR value of less than one is an indication that the MAO converged first before the AO, whereas an AR value of greater than one is an indication that the AO converged faster.

To evaluate the stability of the MAO and AO in solving the benchmark functions effectively, the plot of the objective function values for all the independent runs performed fifty times were generated. This plot of independent runs was generated for only the fifty-dimensional variables considered in this paper, as the interpretation and empirical meaning remains the same irrespective of the dimensional variables considered. The plots of independent runs performed for the unimodal and multimodal functions considering fifty-dimensional variables is shown in Figures 8 and 9, respectively, whereas the plots of independent runs performed for the fixed-dimensional variables is given in Figure 10.

As a measure of robustness, Figures 8–10 gives the distribution of the objective function value obtained by MAO and AO algorithms. From these figures, it can be observed that the MAO provided better quality solutions in F1–F4 and F8, indicating the superiority of the MAO in global searching capability of optimized solutions for these functions. The AO, in turn, obtained more quality solutions for F6 and F16–F23 over the MAO. Both algorithms obtained the same solution quality for F9–F11, whereas the quality of solution obtained for the remaining functions was not as good, even though the global solution for such functions was attained by the algorithms.

Lastly, the convergence curve for all the 50 independent trials juxtaposed on a single figure is presented to examine if the MAO maintained similar convergence characteristics irrespectively of the trials. This was implemented considering the number of dimensional variables as 50 only. In Figure 11, the convergence curve of the MAO for 50 independent trials of the unimodal functions considering 50-dimensional variables is presented. Similarly, the convergence curve of the MAO for the same number of trials and same dimensional variables generated for multimodal and fixed-dimensional variables are presented in Figures 12 and 13, respectively.

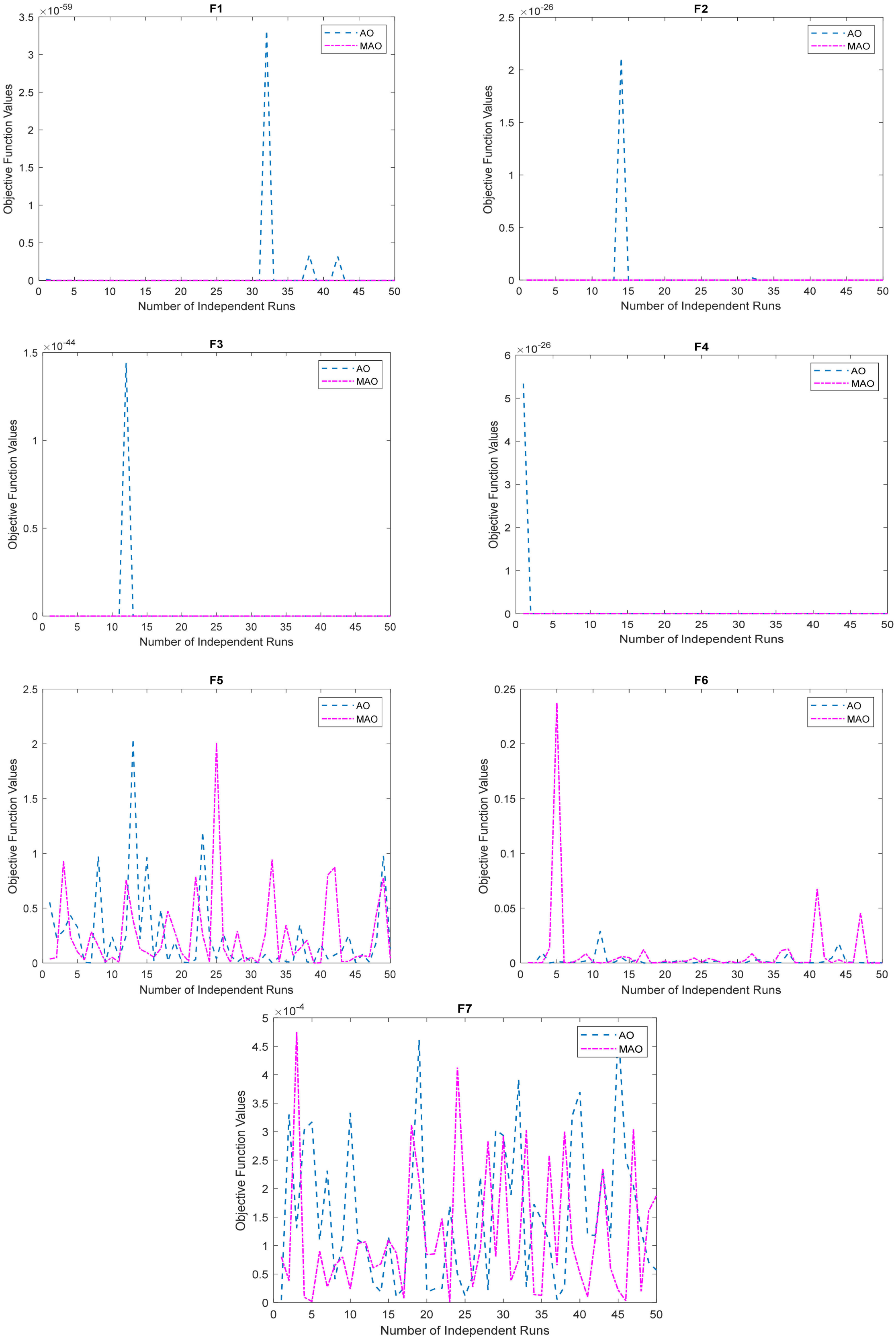


Figure 8. Independent runs of unimodal benchmark functions with D = 50.

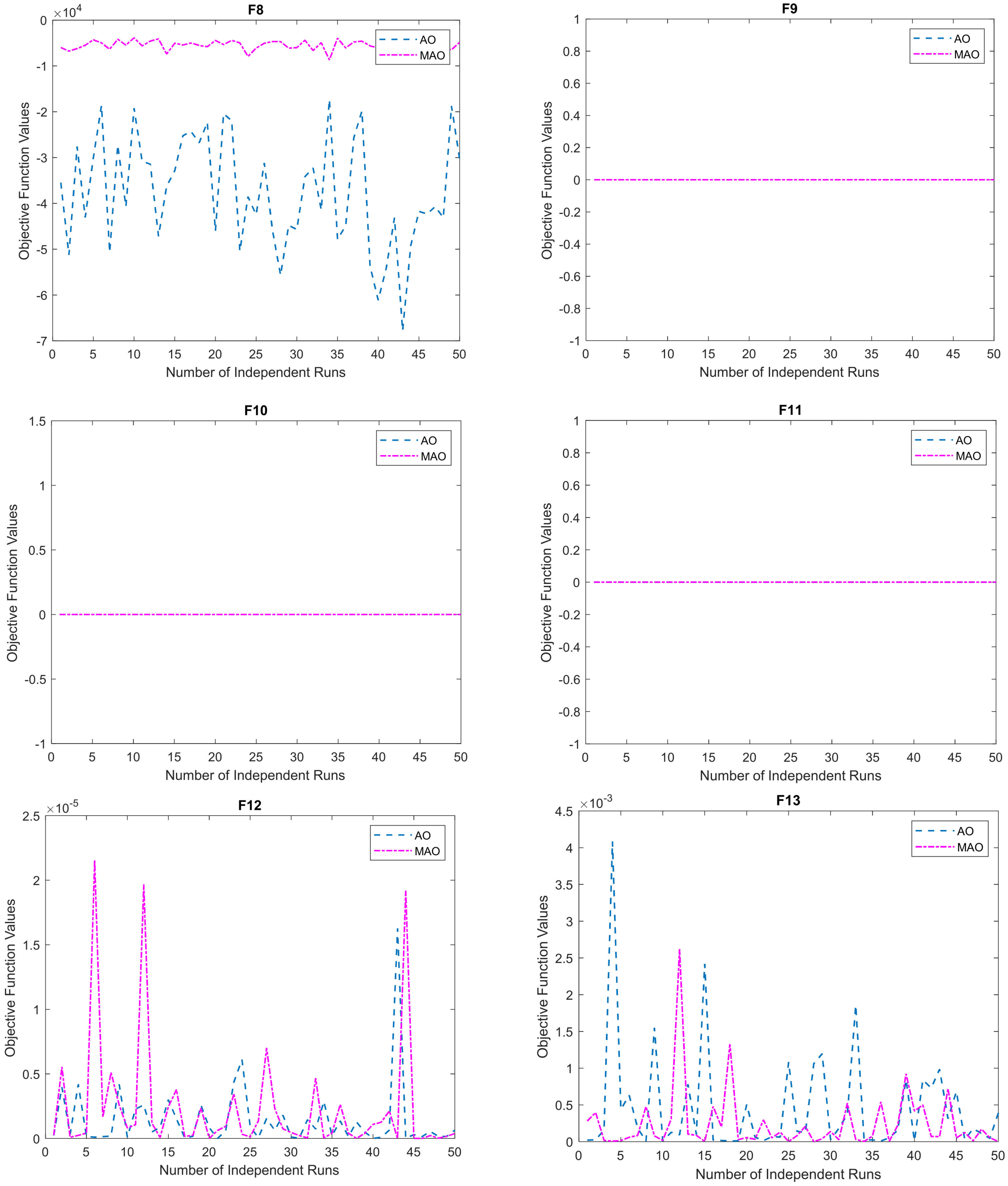


Figure 9. Independent runs of multimodal benchmark functions with D = 50.

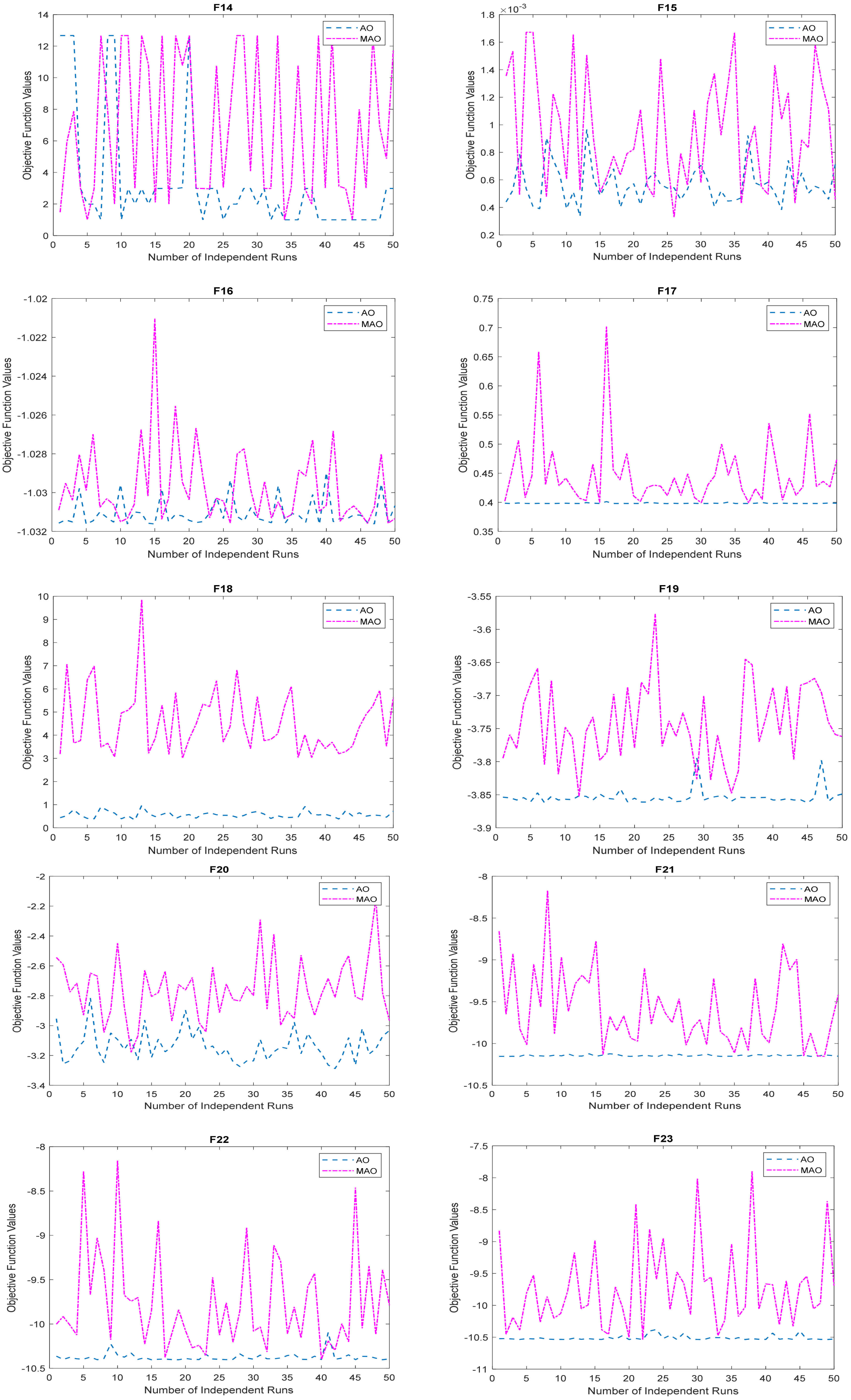


Figure 10. Independent runs of fixed-dimensional benchmark functions with D = 50.

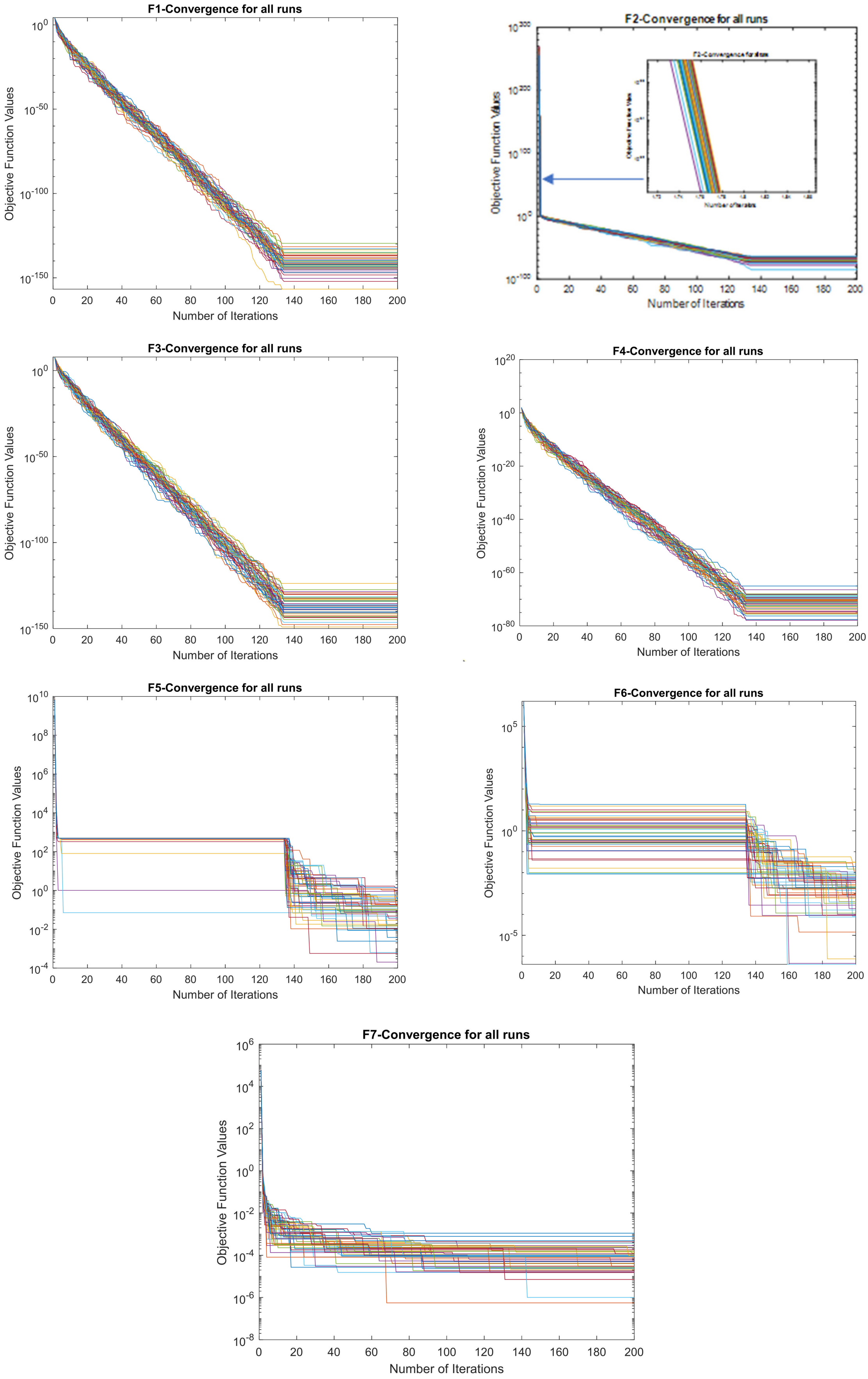


Figure 11. Convergence of MAO over 50 trials for unimodal functions of 50-dimensional variables.

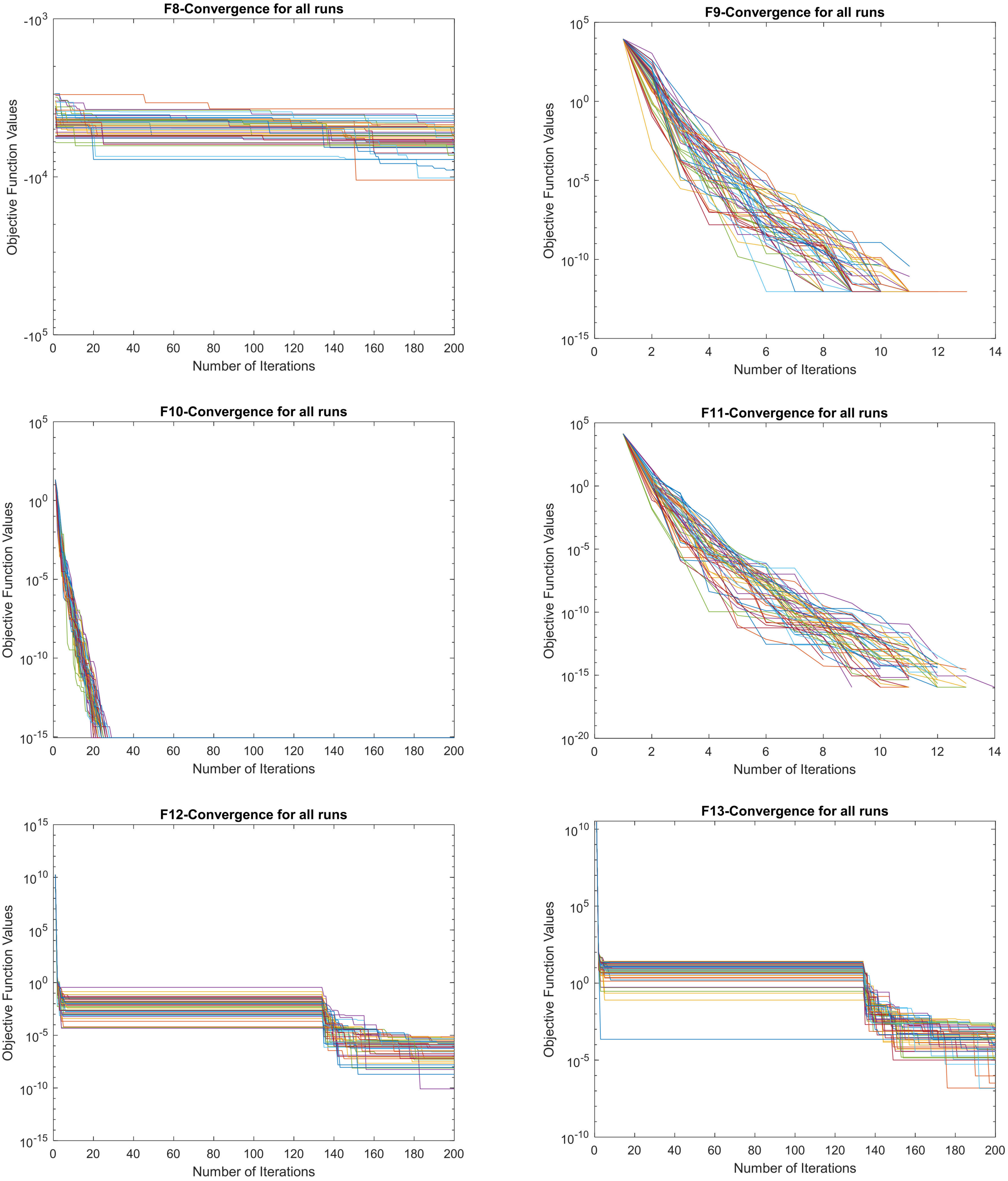


Figure 12. Convergence of MAO over 50 trials for multimodal functions of 50-dimensional variables.

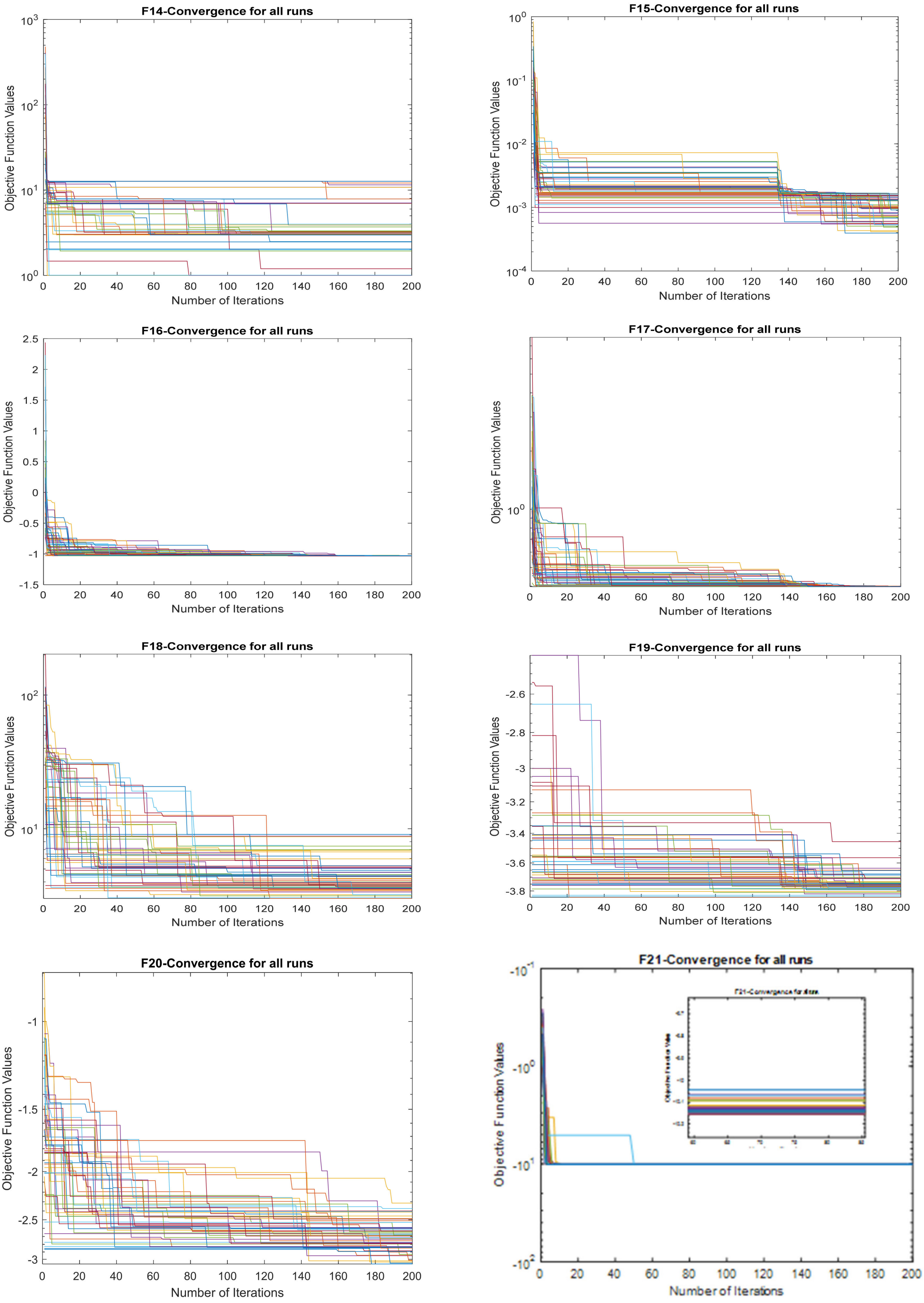


Figure 13. *Cont*.

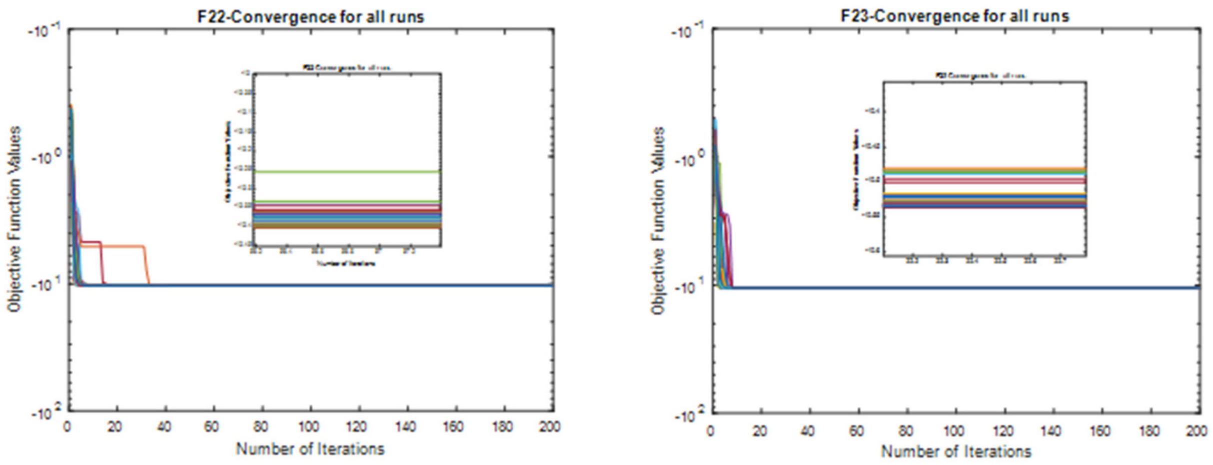


Figure 13. Convergence of MAO over 50 trials for fixed-dimensional variables.

From Figures 11–13, it can be observed that the convergence characteristics of the MAO had very similar behaviors in all the unimodal functions (F1–F7) irrespective of the number of trials. For multimodal functions (F8–F13), the MAO also obtained very similar convergence characteristics in all the functions. In the case of fixed-dimensional variables, the convergence of the MAO showed similar behaviors for F16 and F21–F23. However, F14, F15, and F18–F20 indicated irregular behavior of the MAO in terms of convergence. This demonstrated the stability and robustness of the algorithm in solving various degrees of optimization problems of different complexity and modality.

3.2. ORPD Test Problems

The described algorithm was employed and validated on the IEEE 33-bus network to ensure its usefulness. The starting node voltage was assumed to be 1 pu, and all loading buses were considered viable installation options.

Figure 14 clarifies a single-line schematic of this system [9]. Real and reactive power consumption for the 33-bus test system was 3715 kW and 2300 KVAR, respectively. The test system's initial power loss was 202 kW, and its minimum voltage was 0.9042 pu.

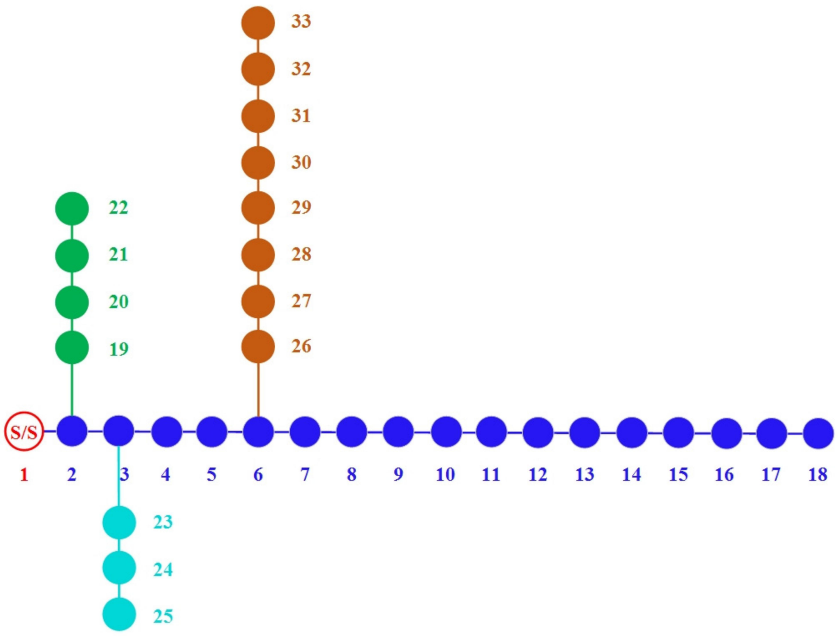


Figure 14. Standard IEEE 33-bus test system.

In this section, the effectiveness of the MAO algorithm is verified. The main DG forms that are used are given in Table 11. The main parameters of algorithm and operative restrictions are listed in Table 12.

Table 11. Various DG types.

DG Type	*P*	*Q*	Example
I	+	0	Fuel cells and photovoltaic devices
II	0	+	Capacitors
III	+	+	Synchronous generators and wind turbines
Note:	(+) produces		(−) absorbs

Table 12. The algorithm parameters and operative restrictions.

Parameters	Value
Number of population	30
Maximum iteration numbers	100
α	0.2
δ	0.13
Base MVA	100 MVA
Base kV	12.66 kV
Node system voltage constraints	0.95 pu $\leq V_i \leq$ 1.05 pu
DG's power generation constraints	0 MW $\leq P_{DG} \leq$ 3 MW

Scope of Work

The proposed work includes two scenarios with different cases.

A. Scenario I: using one DG in different types for supplying various PQ capacities to optimize a certain objective function as below:

Case 1: optimal DG allocation based on minimization of power losses.
Case 2: optimal DG allocation based on minimization of voltage deviation.
Case 3: optimal DG allocation based on loss minimization and voltage deviation reduction.

B. Scenario II: using a number of DGs for supplying various PQ capacities to optimize a certain objective function as below:

Case 4: optimal DG units for minimization of power losses.
Case 5: optimal DG units for minimization of voltage deviation.
Case 6: optimal DG units for loss minimization and voltage deviation reduction.

In the first scenario, a single DG is used with different types at a time to minimize a candidate's multi-objective function. Three cases are implemented, as in case 1, the power loss is minimized, whereas in case 2, it is based on voltage deviation minimization, and minimization of both real power loss and voltage deviation are considered in case 3. The optimal locations and sizes were determined via the AO and MAO and compared with conventional PSO and TIA. The set of optimal values attained by the AO and MAO are given in Tables 13–15.

Table 13. Case 1: optimal DG allocation based on loss minimization.

DG Type	AO					MAO				
	Location	Size		Ploss	VD	Location	Size		Ploss	VD
		KW	KVAR				KW	KVAR		
Base case	-	-	-	201.90	1.6961	-	-	-	201.90	1.6961
DG I	6	2588.6	-	102.79	0.7929	6	2608.1	-	102.79	0.786
DG II	30	-	1311.9	141.84	1.2610	30	-	1310.3	141.84	1.2616
DG III	6	2465	1847.6	61.537	0.4791	6	2515.9	1745.3	61.323	0.4833

Table 14. Case 2: optimal DG allocation based on voltage deviation.

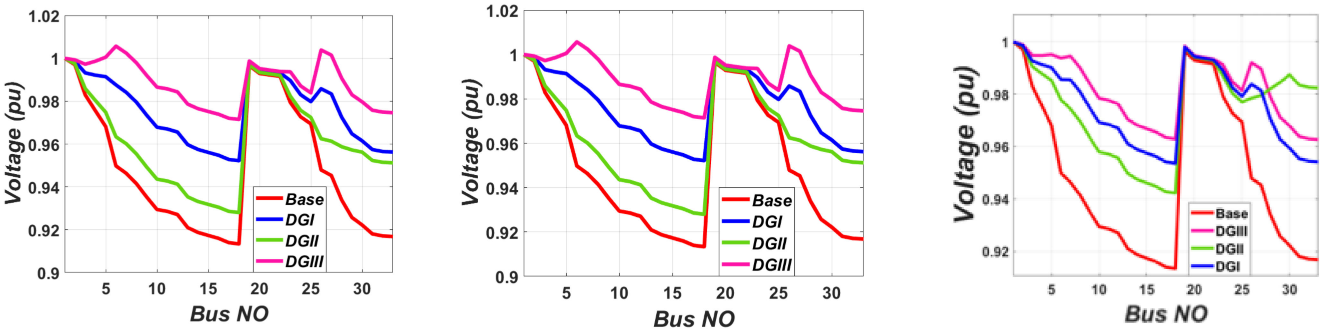
DG Type	AO					MAO				
	Location	DG Size KW KVAR		Ploss	VD	Location	Size KW KVAR		Ploss	VD
Base case	-	-	-	201.90	1.6961	-	-	-	201.90	1.6961
DG I	9	3000	-	149.55	0.3717	9	3000	-	149.55	0.3717
DG II	10	-	2869.2	266.89	0.5292	10	-	2894.19	269.5	0.529
DG III	26	3000	3000	91.685	0.2537	26	3000	3000	91.685	0.2537

Table 15. Case 3: optimal DG allocation based on loss minimization and voltage deviation.

DG Type	AO					MAO				
	Location	Size KW KVAR		Ploss	VD	Location	Size KW KVAR		Ploss	VD
Base case	-	-	-	201.90	1.6961	-	-	-	201.90	1.6961
DG I	7	3000	-	107.26	0.595	7	3000	-	107.26	0.595
DG II	30	-	1876.6	115.26	0.805	30	-	1832.2	114.55	0.825
DG III	7	2430.4	1685.0	103.41	0.801	7	2479.7	130.20	103.37	0.783

Without DG installation, the power losses and the total voltage deviation were 202 kW and 1.7 pu, respectively. Table 13 illustrates that when case 1 took place by employing the proposed three DG types, the power losses were reduced by 49.1%, 29.78%, and 69.64%, respectively. Moreover, the total voltage deviation (TVD) decreased by 53%, 25.6%, and 71.5%. Similarly, from case 2, it was found that the power losses were decreased by 26%, increased by 32%, and decreased by 54.6%, whereas the TVD decreased by 78%, 68.8%, and 85%, as shown in Table 14. Finally, case 3 indicated, as shown in Table 15, that the power losses were reduced by 46.9%, 43.29%, and 48.8%. Additionally, the TVD decreased by 64.9%, 52.5%, and 53.8%.

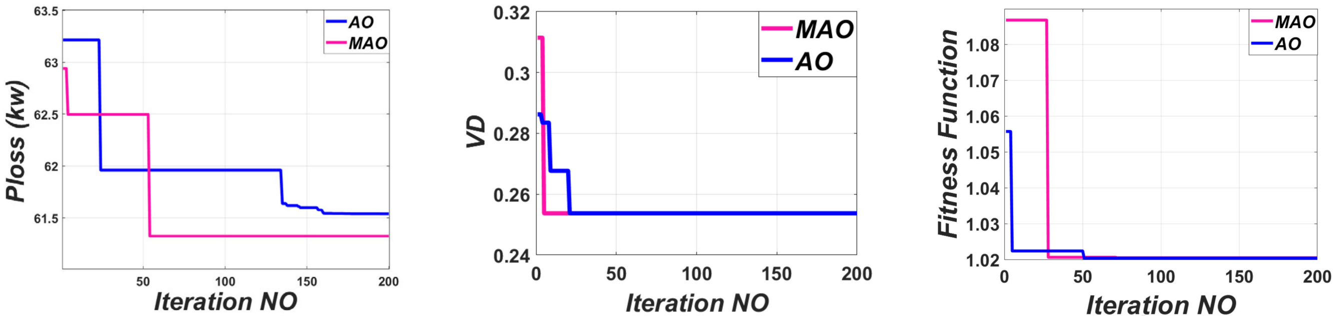
The voltage profiles before and after enhancement are shown in Figure 15. The results emphasize that if case 1 occurred, the DG of type 2 failed to maintain the system voltage within the minimum permissible limits as illustrated in Figure 15a. Noticeably, the performance of the DG of type 3 can strongly outperform that of the other types. Under case 2, the results described in Figure 15b illustrate that the DGs of types 1 and 3 succeeded in keeping the system voltage within the permissible limits, but the remaining type failed to preserve these limits. Moreover, a DG of type 3 showed a substantial impact on the system enhancement level. When considering case 3, the results confirm that the DGs of types 1 and 3 showed a high impact on improving the system performance as clarified in Figure 15c. Noticeably, the DG of type 2 failed to maintain the system voltage within the standard limits.



(**a**) Case 1 (**b**) Case 2 (**c**) Case 3

Figure 15. Voltage profile for IEEE 33-bus test system.

The simulation results proved that considering only a single objective improves this variable at the other's expense. Moreover, it was found that under the power loss minimization criterion, the network minimum TVD had a slight improvement and vice versa. The outcomes showed that the voltage deviation improved and power loss was reduced by applying the DG units of three types based on the three different OFs, considering both objectives are needed to obtain the best performance. The convergence curves of the AO and MAO algorithms for optimizing type 3 are presented in Figure 16. It can be of concern that the MAO converged to the steady value within a period of less than 50 iterations whereas the AO took more than 100 iterations and fell into local optimal solutions, as shown in Figure 16a. The MAO had the best performance along with lower computation time compared with the AO algorithm without trapping into a local optimal solution.



(**a**) Case 1 (**b**) Case 2 (**c**) Case 3

Figure 16. Convergence curve of the IEEE 33-bus system.

Table 16 displays the outcomes of the suggested approach for determining the best size, site, and power factor of DG units in a 33-bus system. According to this table, the suggested algorithm obtained the lowest power loss when compared to the PSO [35] and TIA [35].

Table 16. The compared results of different algorithms in the IEEE 33-bus test system.

DG Type	Comparison		MAO	AO	PSO	TIA
DG I	Location		6	6	6	6
	Size	KW	2608.1	2588.6	2588	2590
		KVAR	0	0	0	0
	Ploss		102.79	102.79	111	111
	VD		0.786	0.7929	-	-
DG II	Location		30	30	30	30
	Size	KW	0	0	0	0
		KVAR	1310.3	1311.9	1257	1260
	Ploss		141.84	141.84	151	151
	VD		1.2616	1.2610	-	-
DG III	Location		6	6	6	6
	Size	KW	2515.9	2465	2700	2560
		KVAR	1745.3	1847.6	1307.7	1920
	Ploss		61.323	61.537	71	68
	VD		0.4833	0.4791	-	-

In the second scenario, simultaneous optimal placement of multi-DGs was implemented for supplying various PQ capacities in the distribution networks, considering power loss lessening and voltage deviation enhancement as an objective function. Table 17 explains the results obtained for choosing the third type of DG, considering power loss reduction only, whereas Tables 18 and 19 present the ODGA, considering voltage deviation reduction only and power loss and voltage deviation minifying together, respectively. Figure 17 shows the voltage profiles for the IEEE 33-bus system in various cases. From the aforementioned tables, it was deduced that power loss was 202 kW and the min voltage obtained was 0.9038 pu at bus 18 without DG. If a single DG was installed at bus 6, the power loss could be reduced to 69.94% and the min voltage would be 0.975 pu at bus 18, whereas placing two DGs simultaneously yielded an 81.2% reduction in power. Figure 17a depicts the voltage profile at various buses for varying numbers of DGs in the network.

Table 17. Case 4: optimal DG III allocation based on loss minimization.

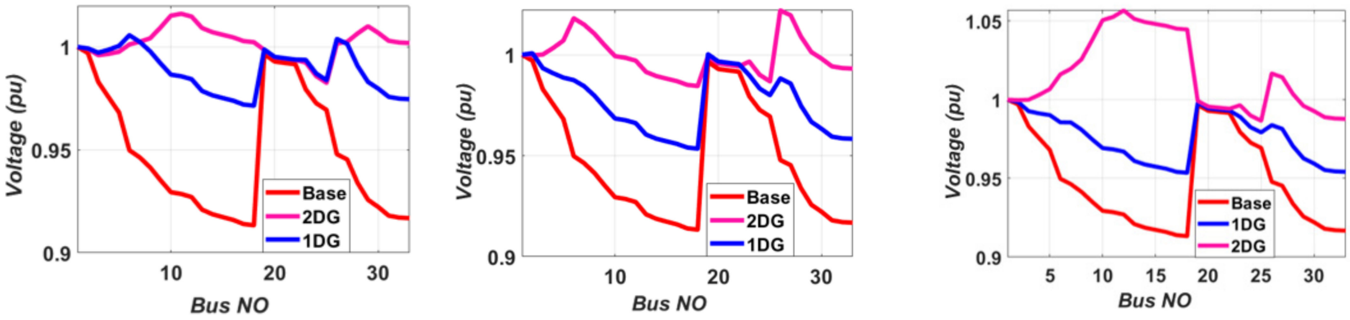
DG Units	Location	DG Size		Ploss	VD
		KW	KVAR		
1 DG	6	2515.9	1745.3	61.323	0.4833
2 DG	11	1117.1	851.1	38.1	0.1985
	29	1213.9	1145.2		

Table 18. Case 5: optimal DG III allocation based on voltage deviation.

DG Units	Location	DG Size		Ploss	VD
		KW	KVAR		
1 DG	26	3000	3000	91.685	0.2537
2 DG	23	1536.82	1032.2	109.2637	0.2442
	26	609.009	270.12		

Table 19. Case 6: optimal DG III allocation based on loss minimization and voltage deviation.

DG Units	Location	DG Size		Ploss	VD
		KW	KVAR		
1 DG	7	2430.4	1685.0	103.41	0.801
2 DG	12	2030.7	1218.2	107.7715	0.6781
	26	1391.0	971.04		



(**a**) Case 4 (**b**) Case 5 (**c**) Case 6

Figure 17. Voltage profile for IEEE 33-bus test system including multiple DGs.

4. Conclusions

In this study, a novel alternate metaheuristic approach, named the modified Aquila optimizer, was created. The performance of the developed MAO was first evaluated using 23 benchmark functions to establish its superiority over the standard Aquila optimizer. Thereafter, both the MAO and AO were used to design a multi-DG placement with the objectives of cutting losses and developing voltage profiles. Proper DG selection considerably diminishes the power loss in the system. The third type of DG that injects real and reactive power reduced the power losses by more than the first and second types of DG for the same number of DG placements. The suggested approach is efficient in terms of loss reduction and voltage enhancement. It also enhances the voltages at the nodes at the tail end.

The study performed in this work was worthwhile, but some significant issues should be emphasized. The basic principles were optimal siting and scaling of DGs in the IEEE 33-bus system; however, other parallel types of study have been conducted to enhance the results. The impacts of the number and type of DG were addressed in this paper. The DG providing varied PQ capabilities was also covered, including cases of P only, Q only, and P and Q together. Three distinct algorithms were used to compare and assess all of the findings (MAO, PSO, and TIA). The following are the visible final conclusions of the research:

(1) The MAO gives the overall best results for the CEC benchmark functions considering solution accuracy and convergence time.
(2) The MAO was tested on the IEEE 33-bus test system to detect its notability.
(3) The MAO gives the optimal result for the ODGA problem in a very short time.
(4) Placement of two DGs is the best case, demonstrating that higher DG incorporation reduces overall power losses and reinforces the voltage profile in the system.
(5) DG delivering both real (P) and reactive (Q) power is the best-suited option for the system, indicating that DG produces the greatest outcomes while operating in an ideal power factor environment.
(6) The proposed algorithm is preferred for obtaining the control variables (DG locations and sizes) to find the best optimal values of the fitness function.
(7) The traditional system is restructured by optimally merging the DG in ideal locations with optimal dimensions. This makes the grid work smarter, which falls within the purview of a smart grid environment.

According to the previous discussion that illustrated the developed MAO's superiority, it can open a wide range of future works. This includes applying the MAO to various applications such as PV parameter estimation, PI parameter tuning, management applications such as energy management and load management, conventional and smart grid applications, industry and engineering applications, other benchmark test functions, and smart home applications. It can also be extended to real-world applications dependent on binary-, discrete-, and multiple-objective optimization. Moreover, it can be mixed with other stochastic algorithms for enhancing their results of optimization problems, escaping the local minima problem, and obtaining a global solution.

Author Contributions: Conceptualization, M.H.A. and S.K.; Methodology, A.T.S. and H.B.S.; Formal analysis, M.H. and M.S.; Validation, M.H.A. and S.K.; Visualization, A.T.S. and H.B.S.; Resources, M.H. and M.S.; Writing—original draft, M.H.A., S.K. and A.T.S.; Writing—review & editing, H.B.S., M.H. and M.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: All authors declare no conflict of interest.

Nomenclature

ABC	artificial bee colony
AO	Aquila optimizer
ALO	ant lion optimization
AVDI	aggregate voltage deviation index
BS	backtracking search
CB	capacitor bank
CEC	competition on evolutionary computation
DG	distributed generation
DNs	distribution networks
DS	distribution system

FLC	fuzzy logic control
GA	genetic algorithm
GSA	gravitational search algorithm
IDSA	improved differential search algorithm
IRRO	improved raven roosting optimization
IWHO	improved wild horse optimization
IWO	invasive weed optimization
LSF	loss sensitivity factor
LLI	line loading index
MAO	modified Aquila optimizer
MODGA	multi-objective optimal distributed generation allocation
MO	multi-objective
MOF	multi-objective function
MOPI	multi-objective performance index
MOSB	multi-objective shuffled bat
MOCDE	MO opposition-based chaotic differential evolution
MOPSO	MO particle swarm optimization
MOHSA	MO harmony search algorithm
OBL	oppositional-based learning
ODGA	optimal distributed generation allocation
ODGP	optimal distributed generation problem
ORESP	optimal renewable energy source placement
PLRI	power loss reduction index
PQ	power quality
PSO	particle swarm optimization
QOCSOS	quasi-oppositional chaotic symbiotic organisms search
QOBL	quasi-oppositional-based learning
RES	renewable energy source
SCA	sine-cosine algorithm
SFLA	shuffled frog-leaping algorithm
TSA	tunicate swarm algorithm
TIA	trader-inspired algorithm
UPSO	unified particle swarm optimization
TVD	total voltage deviation
VDI	voltage deviation index
VPI	voltage profile index
VSI	voltage stability index
VSM	voltage stability margin
WOA	whale optimization algorithm
Symbols	
B	mutual susceptance
D	dimension size
G	mutual conductance
$G1$	prey tracking motion pattern
$G2$	flight slope
G_K	line conductance
$Levy(D)$	levy flight distribution
lb	lower bound
n	total system buses
N	population size
P	real power
P_D	real power demand
P_G	real power generated
PL	real power loss
Q	reactive power
Q_D	reactive power demand
Q_G	reactive power generated
QF	quality function

T	maximum iteration
t	current iteration
ub	upper bound
V_i	voltage at bus i
V_i^{max}	maximum voltage at bus i
V_i^{min}	minimum voltage at bus i
x^0	opposite number
x^{q0}	quasi-opposite number
α	AO adjustment parameter 1
δ	AO adjustment parameter 2

References

1. Mirjalili, S.; Lewis, A. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [CrossRef]
2. Abualigah, L.; Yousri, D.; Elaziz, M.A.; Ewees, A.A.; Al-Qaness, M.A.; Gandomi, A.H. Aquila Optimizer: A novel meta-heuristic optimization algorithm. *Comput. Ind. Eng.* **2021**, *157*, 107250. [CrossRef]
3. Wang, S.; Jia, H.; Abualigah, L.; Liu, Q.; Zheng, R. An Improved Hybrid Aquila Optimizer and Harris Hawks Algorithm for Solving Industrial Engineering Optimization Problems. *Processes* **2021**, *9*, 1551. [CrossRef]
4. Salawudeen, A.T.; Mu'Azu, M.B.; Sha'Aban, Y.A.; Adedokun, A.E. A Novel Smell Agent Optimization (SAO): An extensive CEC study and engineering application. *Knowl.-Based Syst.* **2021**, *232*, 107486. [CrossRef]
5. Alam, S.; Al-Ismail, F.S.; Salem, A.; Abido, M.A. High-Level Penetration of Renewable Energy Sources Into Grid Utility: Challenges and Solutions. *IEEE Access* **2020**, *8*, 190277–190299. [CrossRef]
6. Duong, M.Q.; Pham, T.D.; Nguyen, T.T.; Doan, A.T.; Tran, H.V. Determination of optimal location and sizing of solar photovoltaic distribution generation units in radial distribution systems. *Energies* **2019**, *12*, 174. [CrossRef]
7. Naderi, E.; Narimani, H.; Pourakbari-Kasmaei, M.; Cerna, F.V.; Marzband, M.; Lehtonen, M. State-of-the-Art of Optimal Active and Reactive Power Flow: A Comprehensive Review from Various Standpoints. *Processes* **2021**, *9*, 1319. [CrossRef]
8. Palanisamy, R.; Muthusamy, S.K. Optimal Siting and Sizing of Multiple Distributed Generation Units in Radial Distribution System Using Ant Lion Optimization Algorithm. *J. Electr. Eng. Technol.* **2021**, *16*, 79–89. [CrossRef]
9. Ali, M.H.; Mehanna, M.; Othman, E. Optimal planning of RDGs in electrical distribution networks using hybrid SAPSO algorithm. *Int. J. Electr. Comput. Eng. (IJECE)* **2020**, *10*, 6153–6163. [CrossRef]
10. Ali, M.H.; Kamel, S.; Hassan, M.H.; Tostado-Véliz, M.; Zawbaa, H.M. An improved wild horse optimization algorithm for reliability based optimal DG planning of radial distribution networks. *Energy Rep.* **2021**, *8*, 582–604. [CrossRef]
11. Agajie, T.F.; Salau, A.O.; Hailu, E.A.; Sood, M.; Jain, S. Optimal sizing and siting of distributed generators for minimization of power losses and voltage deviation. In Proceedings of the 2019 5th International Conference on Signal Processing, Computing and Control (ISPCC), Solan, India, 10–12 October 2019; pp. 292–297.
12. Awad, A.; Abdel-Mawgoud, H.; Kamel, S.; Ibrahim, A.; Jurado, F. Developing a Hybrid Optimization Algorithm for Optimal Allocation of Renewable DGs in Distribution Network. *Clean Technol.* **2021**, *3*, 23. [CrossRef]
13. Ali, Z.M.; Diaaeldin, I.M.; HEAbdel Aleem, S.; El-Rafei, A.; Abdelaziz, A.Y.; Jurado, F. Scenario-based network reconfiguration and renewable energy resources integration in large-scale distribution systems considering parameters uncertainty. *Mathematics* **2021**, *9*, 26. [CrossRef]
14. Tan, Z.; Zeng, M.; Sun, L. Optimal Placement and Sizing of Distributed Generators Based on Swarm Moth Flame Optimization. *Front. Energy Res.* **2021**, *9*, 148. [CrossRef]
15. Haider, W.; Hassan, S.J.U.; Mehdi, A.; Hussain, A.; Adjayeng, G.O.M.; Kim, C.-H. Voltage Profile Enhancement and Loss Minimization Using Optimal Placement and Sizing of Distributed Generation in Reconfigured Network. *Machines* **2021**, *9*, 20. [CrossRef]
16. Jordehi, A.R. Optimisation of electric distribution systems: A review. *Renew. Sustain. Energy Rev.* **2015**, *51*, 1088–1100. [CrossRef]
17. Venkatesan, C.; Kannadasan, R.; Alsharif, M.; Kim, M.-K.; Nebhen, J. A Novel Multiobjective Hybrid Technique for Siting and Sizing of Distributed Generation and Capacitor Banks in Radial Distribution Systems. *Sustainability* **2021**, *13*, 3308. [CrossRef]
18. Prakash, P.; Khatod, D.K. Optimal sizing and siting techniques for distributed generation in distribution systems: A review. *Renew. Sustain. Energy Rev.* **2016**, *57*, 111–130. [CrossRef]
19. Hussain, A.; Shah, S.D.A.; Arif, S.M. Heuristic optimisation-based sizing and siting of DGs for enhancing resiliency of autonomous microgrid networks. *IET Smart Grid* **2019**, *2*, 269–282. [CrossRef]
20. Ramamoorthy, A.; Ramachandran, R. Optimal Siting and Sizing of Multiple DG Units for the Enhancement of Voltage Profile and Loss Minimization in Transmission Systems Using Nature Inspired Algorithms. *Sci. World J.* **2016**, *2016*, 1086579. [CrossRef] [PubMed]
21. Prakash, D.; Lakshminarayana, C. Multiple DG placements in radial distribution system for multi objectives using Whale Optimization Algorithm. *Alex. Eng. J.* **2018**, *57*, 2797–2806. [CrossRef]
22. Sa'Ed, J.A.; Amer, M.; Bodair, A.; Baransi, A.; Favuzza, S.; Zizzo, G. A Simplified Analytical Approach for Optimal Planning of Distributed Generation in Electrical Distribution Networks. *Appl. Sci.* **2019**, *9*, 5446. [CrossRef]
23. Sharma, S.; Bhattacharjee, S.; Bhattacharya, A. Quasi-Oppositional Swine Influenza Model Based Optimization with Quarantine for optimal allocation of DG in radial distribution network. *Int. J. Electr. Power Energy Syst.* **2016**, *74*, 348–373. [CrossRef]

24. El-Fergany, A. Multi-objective Allocation of Multi-type Distributed Generators along Distribution Networks Using Backtracking Search Algorithm and Fuzzy Expert Rules. *Electr. Power Compon. Syst.* **2015**, *44*, 252–267. [CrossRef]
25. Prabha, D.R.; Jayabarathi, T. Optimal placement and sizing of multiple distributed generating units in distribution networks by invasive weed optimization algorithm. *Ain Shams Eng. J.* **2016**, *7*, 683–694. [CrossRef]
26. Mehta, P.; Bhatt, P.; Pandya, V. Optimal selection of distributed generating units and its placement for voltage stability enhancement and energy loss minimization. *Ain Shams Eng. J.* **2018**, *9*, 187–201. [CrossRef]
27. Gkaidatzis, P.A.; Bouhouras, A.S.; Sgouras, K.I.; Doukas, D.I.; Christoforidis, G.C.; Labridis, D.P. Efficient RES Penetration under Optimal Distributed Generation Placement Approach. *Energies* **2019**, *12*, 1250. [CrossRef]
28. Yammani, C.; Maheswarapu, S.; Matam, S.K. A Multi-objective Shuffled Bat algorithm for optimal placement and sizing of multi distributed generations with different load models. *Int. J. Electr. Power Energy Syst.* **2016**, *79*, 120–131. [CrossRef]
29. Injeti, S.K. A Pareto optimal approach for allocation of distributed generators in radial distribution systems using improved differential search algorithm. *J. Electr. Syst. Inf. Technol.* **2018**, *5*, 908–927. [CrossRef]
30. Tang, D.; Zhao, J.; Yang, J.; Liu, Z.; Cai, Y. An Evolutionary Frog Leaping Algorithm for Global Optimization Problems and Applications. *Comput. Intell. Neurosci.* **2021**, *2021*, 8928182. [CrossRef]
31. Shandilya, S.; Izonin, I.; Shandilya, S.K.; Singh, K.K. Mathematical modelling of bio-inspired frog leap optimization algorithm for transmission expansion planning. *Math. Biosci. Eng.* **2022**, *19*, 7232–7247. [CrossRef]
32. Kumar, S.; Mandal, K.; Chakraborty, N. Optimal DG placement by multi-objective opposition based chaotic differential evolution for techno-economic analysis. *Appl. Soft Comput.* **2019**, *78*, 70–83. [CrossRef]
33. Nagaballi, S.; Kale, V.S. Pareto optimality and game theory approach for optimal deployment of DG in radial distribution system to improve techno-economic benefits. *Appl. Soft Comput.* **2020**, *92*, 106234. [CrossRef]
34. Truong, K.H.; Nallagownden, P.; Elamvazuthi, I.; Vo, D.N. A Quasi-Oppositional-Chaotic Symbiotic Organisms Search algorithm for optimal allocation of DG in radial distribution networks. *Appl. Soft Comput.* **2020**, *88*, 106067. [CrossRef]
35. Mohamed, M.A.; El-Sehiemy, R.A.; El-Dabah, M.A. Optimal operation of electrical distribution networks including distributed generation units using trader optimization algorithm. In Proceedings of the CIRED 2021 Conference, online, 20–23 September 2021.
36. Basetti, V.; Rangarajan, S.S.; Kumar Shiva, C.; Verma, S.; Collins, R.E.; Senjyu, T. A Quasi-Oppositional Heap-Based Optimization Technique for Power Flow Analysis by Considering Large Scale Photovoltaic Generator. *Energies* **2021**, *14*, 5382. [CrossRef]