





Article

Modified Artificial Hummingbird Algorithm-Based Single-Sensor Global MPPT for Photovoltaic Systems

Hesham Alhumade ^{1,2,3} , Essam H. Houssein ^{4,*} , Hegazy Rezk ⁵ , Iqbal Ahmed Moujдин ^{2,6} 
and Saad Al-Shahrani ¹

¹ Chemical and Materials Engineering Department, Faculty of Engineering, King Abdulaziz University, Jeddah 21589, Saudi Arabia

² Center of Excellence in Desalination Technology, King Abdulaziz University, P.O. Box 80200, Jeddah 21589, Saudi Arabia

³ K. A. CARE Energy Research and Innovation Center, King Abdulaziz University, Jeddah 21589, Saudi Arabia

⁴ Faculty of Computers and Information, Minia University, Minia 61519, Egypt

⁵ Department of Electrical Engineering, College of Engineering in Wadi Alddawasir, Prince Sattam bin Abdulaziz University, Wadi Alddawasir 11991, Saudi Arabia

⁶ Department of Mechanical Engineering, King Abdulaziz University, P.O. Box 80200, Jeddah 21589, Saudi Arabia

* Correspondence: essam.halim@mu.edu.eg

Abstract: Recently, a swarm-based method called Artificial Hummingbird Algorithm (AHA) has been proposed for solving optimization problems. The AHA algorithm mimics the unique flight capabilities and intelligent foraging techniques of hummingbirds in their environment. In this paper, we propose a modified version of the AHA combined with genetic operators called mAHA. The experimental results show that the proposed mAHA improved the convergence speed and achieved better effective search results. Consequently, the proposed mAHA was used for the first time to find the global maximum power point (MPP). Low efficiency is a drawback of photovoltaic (PV) systems that explicitly use shading. Normally, the PV characteristic curve has an MPP when irradiance is uniform. Therefore, this MPP can be easily achieved with conventional tracking systems. With shadows, however, the conditions are completely different, and the PV characteristic has multiple MPPs (i.e., some local MPPs and a single global MPP). Traditional MPP tracking approaches cannot distinguish between local MPPs and global MPPs, and thus simply get stuck at the local MPP. Consequently, an optimized MPPT with a metaheuristic algorithm is required to determine the global MPP. Most MPPT techniques require more than one sensor, e.g., voltage, current, irradiance, and temperature sensors. This increases the cost of the control system. In the current research, a simple global MPPT method with only one sensor is proposed for PV systems considering the shadow conditions. Two shadow scenarios are considered to evaluate the superiority of the proposed mAHA. The obtained results show the superiority of the proposed single sensor based MPPT method for PV systems.

Keywords: modified artificial hummingbird algorithm; metaheuristics; maximum power point tracking; single sensor

MSC: 68T99; 68U99



Citation: Alhumade, H.; Houssein, E.H.; Rezk, H.; Moujдин, I.A.; Al-Shahrani, S. Modified Artificial Hummingbird Algorithm-Based Single-Sensor Global MPPT for Photovoltaic Systems. *Mathematics* **2023**, *11*, 979. <https://doi.org/10.3390/math11040979>

Academic Editor: Fuyuan Xiao

Received: 4 January 2023

Revised: 3 February 2023

Accepted: 10 February 2023

Published: 14 February 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Optimization is used to solve many problems by maximizing or minimizing objective functions. Various optimization methods are used to find the optimal solution. Metaheuristic algorithms (MHs) are optimization tools in which various methods are used to increase the effectiveness of search processes [1–4]. Although it is difficult to find an exact solution in most cases, algorithms can provide the optimal global solution [5].

In this context, MHs can be divided into two categories: stochastic and deterministic methods. If the gradient is available, deterministic algorithms could be an option.

Since these different techniques are based on established computations, it can be assumed that the optimization procedure is repeatable. Each time the algorithm is run with the same initial conditions, the optimization path and result are identical, making the process replicable. Stochastic methods, on the other hand, have random properties that generate different optimization pathways, leading to different optimized solutions even if the initial conditions are the same for each run. The main advantage of stochastic algorithms is that they do not require gradient information. However, accessing gradient information is becoming increasingly difficult as problems become more complex. Researchers have developed stochastic algorithms that do not require gradient information. The category of MHs was developed due to the advancement of stochastic techniques [6]. Some stable MHs, including genetic algorithm (GA) [7], particle swarm optimization (PSO) [8], bee colony optimization (BCO) [9], and differential evolution [10], have been used to solve various optimization problems. For example, MHs have excelled in several real-world applications, such as medicine [11,12], COVID-19 [13,14], feature selection [15,16], and image segmentation [17], as well as combinatorial problems [18–20].

Recently, an efficient swarm-based method called artificial hummingbird algorithm (AHA) has been proposed for solving optimization problems [21]. The source of inspiration of AHA is to mimic the unique flight capabilities and intelligent foraging behavior of hummingbirds in their environment. Although AHA has been used to solve various optimization problems, AHA still suffers from some drawbacks, such as (1) fast convergence and robust utilization of space, (2) fast convergence to the found optimal solution, (3) lack of self-adaptation of convergence rate. The aforementioned drawbacks cause the algorithm to fall victim to the problem of premature convergence, which leads to further problems, such as the algorithm getting stuck in regions with local optima, which prevents it from finding the nearest/best solutions.

In this paper, the genetic operators are combined with the original AHA algorithm to propose a modified version called mAHA. The role of genetic operators is to drive the solution from local to global to improve the position and enable trapping in local search, increasing the diversity of algorithms and avoiding local solutions in the comparison between a new and a previous solution. The mAHA was used to solve the IEEE CEC'20 test suite for optimization testing, and the results were compared with several stable methods, including SMA [22], HHO [23], GWO [24], WOA [25], and the original AHA [21]. The experimental results show that the proposed mAHA achieves (1) effective search results, (2) lower computational cost, (3) good convergence speed, (4) balanced exploration and exploitation behavior, and (5) avoidance of local optimum.

Electric power generation has been driven by new and renewable energy sources (RES), which have seen rapid growth in the energy industry in recent years. Their ability to provide clean, renewable, maintenance-free, ubiquitous, and low-cost energy has increased the popularity of renewable energy [26]. Moreover, renewables can improve the reliability and self-healing operation of utility grids through their reconfiguration into microgrids and nanogrids. The installed capacities of renewable energy sources are increasing rapidly worldwide. Photovoltaic (PV) capacity will increase by 179 TWh in 2021, a 22% growth from 2020. It accounts for 3.6% of global electricity generation [27]. PV electricity is the third largest energy source after hydropower and wind. It is expected that PV generation will be about 7400 TWh in 2030. Tracking the maximum power point (MPP) is very important for increasing the efficiency of the PV system. Under normal conditions, i.e., uniform irradiance distribution, the power-voltage characteristic has only a single MPP [28]. This MPP can be easily extracted using conventional MPPT techniques such as incremental resistance and hill climbing. The situation is quite different for shaded conditions. Especially for large PV systems, the probability of the presence of shade due to nearby trees or buildings is very high [29]. The shadow problem reduces the PV output power. It is difficult to avoid the shading. Therefore, a bypass diode is used to mitigate the shadow and protect the PV system. During shading, the bypass diode is on and provides an additional path to current draw. However, during normal operation, the bypass diode is off and has no effect [30].

In summary, the power-voltage curve with shadow has some local MPP and one global MPP. The conventional MPP tracking approaches cannot distinguish between local and global MPPs and therefore simply stop at the local MPP. Consequently, an optimized MPPT with a metaheuristic algorithm is required to determine the global MPP. The previous algorithms that have been used to mitigate the shadow effect are the Marine Predator algorithm [31], Search & Rescue algorithms [32], Group Teaching Optimization [33], and the Firefly algorithm [34].

The disadvantage of previous techniques is that more than one sensor is needed, e.g., voltage, current, irradiance, and temperature sensors. El-Helw et al. [35] proposed a global MPPT method using a neural network that requires six sensors: four irradiance sensors, voltage, and current. Moreover, the proposed method depends on the PV module characteristics. In the same direction, Nafeh [36] proposed an MPPT method using perturbation and observation, which requires five sensors: two voltage sensors, two current sensors, and one temperature sensor. Moreover, the shadow problem was not considered. Camilo et al. [37] proposed an MPPT method based on the Monod equation. This technique requires three sensors: current, temperature and radiation. Moreover, the shadow condition is not considered. To fill the defined research gap, a global MPPT method with one sensor for PV systems is proposed in this paper. For the first time, a mAHA is proposed to mitigate the shadow and determine the global MPP in PV systems.

The contributions of the paper can be outlined as follows.

1. In this paper, an effective method called mAHA is proposed.
2. The proposed mAHA introduces the mechanisms of genetic operators (crossover and mutation selection) to enhance AHA's performance in increasing the diversity of the population and avoiding local searches.
3. The proposed mAHA was adopted to address the ten global optimization tasks from the CEC'20 test suite and was compared with other optimization algorithms and the original AHA algorithm.
4. For the first time, a mAHA was used in the global MPPT optimization of PV systems with a sensor.
5. The superiority of the proposed MPPT technique has been demonstrated.

The remaining sections are organized as follows: The artificial hummingbird algorithm (AHA) is discussed in Section 2. Section 3 discusses the proposed mAHA algorithm. Section 4 presents the evaluation results, including the CEC'20 test suite. Section 5 present the maximum power point tracking (MPPT) application results. A brief discussion is introduced in Section 6. Finally, Section 7 presents the conclusions and future work.

2. Artificial Hummingbird Algorithm

In this section, we explain the mathematical model of AHA, which is biologically inspired by the smart behavior of hummingbirds and developed to solve various optimization problems [21]. The three basic phases of the AHA algorithm is explained below:

Food sources: The hummingbird frequently evaluates the characteristics of the sources, such as the nectar quality/quantity of each nectar replenishment rate, the flowers, and the last time the flowers were visited, to select a suitable source from a group of food sources. For simplicity, it is assumed that each food source in AHA has the same type and number of flowers—the solution vector represents the food source, and the function fitness value represents the nectar fill rate. Accordingly, the higher the nectar fill rate, the better the fitness value.

Hummingbirds: each hummingbird has a specific food source from which it can only feed, so the food source and the hummingbird are constantly in the same place. The hummingbird can register the location and nectar replenishment of a particular food source and pass this knowledge on to other hummingbirds throughout the population. In addition, each hummingbird could remember how long ago it last visited a particular food source.

Visit table: for different hummingbirds, the visit frequency for each food source is recorded in the visit table and indicates how long it has been since the same hummingbird

visited a particular food source. For a hummingbird, the food source with the highest visitation rate is prioritized for visitation. To obtain more nectar, among the food sources, a hummingbird should visit the one that has the highest nectar replenishment rate with the same highest visitation level. Moreover, the visitation table is updated at each iteration to find the target food source.

Moreover, the mathematical phases of the three foraging behaviors of hummingbirds including the following phases: (1) guided foraging, (2) territorial foraging, and (3) migrating foraging are discussed in the following subsections. In Algorithm 1, the generic AHA structure is provided [21].

Algorithm 1: Structure of AHA.

Initialization
 While stop criterion is not satisfied
 Guided foraging
 Territorial foraging
 Migration foraging
 End

2.1. The Mathematical Model of AHA Algorithm

2.1.1. Initialization Phase

The n food sources construct the population of n hummingbirds, and is initialized randomly as defined in [38] by Equation (1):

$$x_i = Low + r \cdot (Up - Low) \quad i = 1, \dots, \tag{1}$$

where for d -Dim problem, the Low and Up represent the upper and lower boundaries. The variable r is a random vector within $[0, 1]$, and the i th food source position is represented by x_i which is the solution of a specific problem. Accordingly, the visit table is defined by Equation (2):

$$VT_{i,j} = \begin{cases} 0 & \text{if } i \neq j \\ \text{null} & \text{if } i = j \end{cases} \quad i = 1, \dots, n; j = 1, \dots, n \tag{2}$$

where for $i = j$, $VT_{i,j} = \text{null}$ refers to hummingbird is taking food from its particular food source; for $i \neq j$, $VT_{i,j} = 0$ refers to j th food source that has just been visited by the i th hummingbird in this current iteration.

2.1.2. Guided Foraging Phase

By incorporating a direction-switching vector into the AHA algorithm, three flight capabilities are used as follows: omnidirectional, diagonal, and axial flights are appropriately described during foraging. This vector is used to regulate the availability of one or more directions in a dimension space. Axial flight shows that hummingbirds can fly along any coordinate axis, while diagonal flight, which is determined by any two of three axes, allows hummingbirds to fly from one corner of the rectangle to the other.

To be specific, all three flight patterns can be adapted to d -Dim, where axial flying is formulated by Equation (3) as:

$$D^{(i)} = \begin{cases} 1 & \text{if } i = \text{randi}([1, d]) \\ 0 & \text{else} \end{cases} \quad i = 1, \dots, d \tag{3}$$

Moreover, diagonal flight is formulated by Equation (4) as:

$$D^{(i)} = \begin{cases} 1 & \text{if } i = P(j), j \in [1, k], P = \text{randperm}(k), k \in [2, \lceil r_1 \cdot (d - 2) \rceil + 1] \\ 0 & \text{else} \end{cases} \quad i = 1, \dots, d \tag{4}$$

Then, omnidirectional flight is formulated by Equation (5) as:

$$D^{(i)} = 1 \quad i = 1, \dots, d \tag{5}$$

where $\text{randi}([1, d])$ produces randomly an integer within 1 and d , a random integer permutation within 1 and k is generated by the $\text{randperm}(k)$, and r_1 is a random number within $(0, 1]$. Diagonal flight in a d -Dim is placed within a hyperrectangle that is bound by any 2 to $d-1$ coordinate axes.

Moreover, the mathematical modelling for the guided foraging behavior and candidate food source is defined as follows:

$$v_i(t + 1) = x_{i,\text{tar}}(t) + a \cdot D \cdot (x_i(t) - x_{i,\text{tar}}(t)) \tag{6}$$

$$a \sim N(0, 1) \tag{7}$$

where the i th food source position at time t is represented by $x_i(t)$. The target food source at i th hummingbird is represented by $x_{i,\text{tar}}(t)$, the variable a is a guided factor with mean equal 0 and standard deviation equal 1.

Accordingly, the position of i th food source is updated as follows:

$$x_i(t + 1) = \begin{cases} x_i(t) & f(x_i(t)) \leq f(v_i(t + 1)) \\ v_i(t + 1) & f(x_i(t)) > f(v_i(t + 1)) \end{cases} \tag{8}$$

where the function fitness value is indicated by $f(\cdot)$. The AHA's guided foraging method is presented in Algorithm 2 [21].

Algorithm 2: Guided foraging strategy of AHA.

```

For  $i$  th hummingbird from 1 to  $n$ 
  Perform Equation (6)
  If  $f(v_i(t + 1)) < f(x_i(t))$ 
     $x_i(t + 1) = v_i(t + 1)$ 
    For  $j$  th food source from 1 to  $n (j \neq \text{tar}, i)$ 
      Visit_table ( $i, j$ ) = Visit_table ( $i, j$ ) + 1
    End
    Visit_table ( $i, \text{tar}$ ) = 0
  For  $j$  th food source from 1 to  $n$ 
    Visit_table ( $j, i$ ) =  $\max_{l \in n \text{ and } l \neq j} (\text{Visit\_table} (j, l)) + 1$ 
  End
Else
  For  $j$  th food source from 1 to  $n (j \neq \text{tar}, i)$ 
    Visit_table ( $i, j$ ) = Visit_table ( $i, j$ ) + 1
  End
  Visit_table ( $i, \text{tar}$ ) = 0
End
End
    
```

2.1.3. Territorial Foraging Phase

The local search within territorial foraging phase and the candidate food source is defined by the mathematical formula:

$$v_i(t + 1) = x_i(t) + b \cdot D \cdot x_i(t) \tag{9}$$

$$b \sim N(0, 1) \tag{10}$$

where the territorial factor is a normal distribution $N(0, 1)$, represented by b with mean equal 0 and standard deviation equal 1. The territorial foraging phase of AHA algorithm is reported in Algorithm 3 [21].

Algorithm 3: Territorial foraging strategy of AHA.

```

For  $i$  th hummingbird from 1 to  $n$ 
  Perform Equation (9)
  If  $f(v_i(t+1)) < f(x_i(t))$ 
     $x_i(t+1) = v_i(t+1)$ 
    For  $j$  th food source from 1 to  $n(j \neq i)$ 
      Visit_table ( $i, j$ ) = Visit_table ( $i, j$ ) + 1
    End
    For  $j$  th food source from 1 to  $n$ 
      Visit_table ( $j, i$ ) =  $\max_{l \in n \text{ and } l \neq j}(\text{Visit\_table}(j, l)) + 1$ 
    End
  Else
    For  $j$  th food source from 1 to  $n(j \neq i)$ 
      Visit_table ( $i, j$ ) = Visit_table ( $i, j$ ) + 1
    End
  End
End
End
    
```

2.1.4. Migration Foraging Phase

Randomly, from the source, the obtained migration foraging with the lowest nectar-refilling rate to a new one is defined as:

$$x_{wor}(t+1) = Low + r \cdot (Up - Low) \tag{11}$$

where the food source is represented by x_{wor} in the population with the lowest nectar-refilling rate. Moreover, the migration foraging phase of AHA algorithm is reported in Algorithm 4 [21].

Algorithm 4: Migration foraging strategy of AHA.

```

If mod( $t, 2n$ ) == 0
  Perform Equation (11)
  For  $j$  th food source from 1 to  $n(j \neq wor)$ 
    Visit_table ( $wor, j$ ) = Visit_table ( $wor, j$ ) + 1
  End
  For  $j$  th food source from 1 to  $n$ 
    Visit_table ( $j, wor$ ) =  $\max_{l \in n \text{ and } l \neq j}(\text{Visit\_table}(j, l)) + 1$ 
  End
End
End
    
```

In AHA, the following formula is used to define the migration coefficient in relation to population size as follows:

$$M = 2n \tag{12}$$

3. The Proposed mAHA Algorithm

In this section, the AHA algorithm is combined with genetic operators with AHA algorithm to build a new version called mAHA. To be specific, the AHA includes three stages: (1) Search agents are controlled; (2) search agents are produced for the hybrid method; (3) the new population based on the previous step is updated.

In the update step, all AHA search agents control each position. Here, the genetic operators produce more populations to avoid local optimality and affect AHA positions.

Creating a new hybrid search agent requires creating a new hybrid individual from the current AHA as shown in Equation (13).

$$X_{\text{hybrid}} = \alpha \times X_i + (1 - \alpha) \times S_i \quad (13)$$

where α value between $[0, 1]$ indicates AHA weight, X_i is AHA position, and S_i represents AHA individual position based on the operators, so this Equation (13) indicates the effects of them. The search agent is updated using a greedy selection to select the optimal candidate solution. To improve the quality of solutions and performance of the proposed algorithm. The reasons that motivate us to present an alternative version of AHA called mAHA algorithm are described in the following subsections.

3.1. Shortcomings of the Original AHA

Although the original AHA algorithm exhibits rapid convergence during the exploration of the optimization problem space, where the solutions in each iteration move rapidly toward the optimal solution found at the current time, this solution may not be the best for the entire problem space. Moreover, the convergence of the algorithm is not self-adjusted to encourage the proposed solutions to explore the remaining regions in the problem space, but the convergence rate of AHA is not self-adjusted, which causes the algorithm to suffer from the problem of premature convergence. This leads to other problems, such as the algorithm getting stuck in regions with local optima and not being able to identify the closest/best solutions. For complex optimization problems, the algorithm AHA, like other metaheuristic optimization methods, is unable to adequately explore the entire search space in its current state [2].

Therefore, to address these drawbacks of AHA, this paper proposes an alternative version called modified artificial hummingbird algorithm (mAHA) by integrating the genetic operator's techniques. Genetic operators control the solution from local to global using Equations (15) and (20) to enhance the position and permit the trapping in local search, thereby increasing the variety of algorithms and avoiding local solutions compared between a new and previous one. For more diversity, genetic algorithm operators are used to integrate two mutant vectors, namely y_{Mut} and z_{Mut} , to generate a new child w_{Cross} as described in Equation (19). Offspring fitness value: y_{Mut} , z_{Mut} , and w_{Cross} based on the selection operator Equation (20) were compared to get the best prey p_x . To identify the best candidate solution, the search space was boosted by exploring new regions, so several positions are updated by Equations (19) and (20). The stop condition is the maximum iterations that permit for evaluating the performance of the mAHA algorithm. Upon completing the mAHA process, the best position is returned. Algorithm 5 demonstrates how the best solution is proposed by calculating the objective function using Equation (14) for many new populations. mAHA fitness ($fobj$) is indicated as defined as:

$$fobj = \alpha + \beta \quad (14)$$

where $\beta = \alpha$ and $fobj > T$, where T is the greatest iteration. However, neither the algorithm performance nor the optimization problem space is affected.

Additionally, plots of convergence curves in most testing functions show a slow divergence of the original AHA during exploitation at most function evaluations, demonstrating that the approach is trapping in local rather than global regions.

3.2. Architecture of the Proposed mAHA Algorithm

The main goal of this research is to propose an efficient alternative version of the original AHA to solve the problems mentioned in the previous section. However, it improves the algorithm AHA, by increasing the exploration time and maintaining diversity using the genetic operator strategy.

The mathematical modeling and steps of the mAHA are formulated in Algorithm 5. The algorithm starts with a random starting population of size $2N$. In the modified mAHA,

the solutions of the population are evolved first, followed by the individual phase from the original AHA. At each iteration, a linear population reduction is applied to manage the population diversity, and the same approach is repeated until the optimization is complete. The following subsections explain the genetic operators.

Genetic Operators

Several algorithms use evolutionary operators, especially the two basic algorithms namely, differential evolution and genetic algorithms. Some examples of such operators are mutation as Bit inversion, crossover as single-point crossover and selection.

Mutation: The mutation operation is built using the results of AHA tasks, as the solution goal *Position*. A number between 0 and 1 is produced at random for each component. The target agent element position is considered when the value reaches the mutation rate (zeta). If this value is less than the mutation rate (zeta), the old vector is replaced with a component of the *y* or *z* vectors. The mutation operator is determined by applying the following formula: where j^{th} dimension is known by lb^j and ub^j, r_v have D components generated by randomly between (0, 1).

$$y_{Mut} = \begin{cases} position & \text{if } rand_1 \geq \zeta \\ y & \text{else} \end{cases} \text{ and } z_{Mut} = \begin{cases} position & \text{if } rand_2 \geq \zeta \\ z & \text{else} \end{cases} \tag{15}$$

$$\text{Where : } \begin{cases} \zeta = \frac{1}{T}; \\ y = |position - x_l^i| \end{cases} \tag{16}$$

$$y_{Mut} = \begin{cases} position & \text{if } \rho_1 \geq \zeta \\ y & \text{else} \end{cases} \text{ and } z_{Mut} = \begin{cases} position & \text{if } \rho_2 \geq \zeta \\ z & \text{else} \end{cases} \tag{17}$$

$$\text{Where : } \begin{cases} \zeta = \frac{t}{T}; \\ y = |(\text{position}) - x_l^i| \\ z = y - r_v \end{cases} \tag{18}$$

Crossover: The crossover is the combination of two individuals to produce more variety. To generate a new offspring w_{Cross} , a linear combination with random integers tau and tau' is used.

$$w_{Cross} = \tau * y_{Mut} + (1 - \tau') * z_{Mut} \text{ and } \tau \neq \tau' \tag{19}$$

Selection: The selection type used in AHA is a greedy selection, which is based on differential evolution. When functions of evolution (mutation and crossover) are accessed, the offspring are created. The child and parent's performance are then compared to determine which is the best. Finally, if the parent's performance is good, then they have a chance to stay in the population. The rule that defines greedy selection is as follows:

$$p_{x+1}^i = \begin{cases} y_{Mut} & \text{if } \text{fit}(y_{Mut}) < \text{fit}(P(x)^i) \\ z_{Mut} & \text{if } \text{fit}(z_{Mut}) < \text{fit}(P(x)^i) \\ w_{Cross} & \text{if } \text{fit}(w_{Cross}) < \text{fit}(P(x)^i) \end{cases} \tag{20}$$

where $p(x)$ is the current position and (p_{x+1}^i) is the next position.

Eventually, the pseudo code of the proposed mAHA algorithm is given in Algorithm 5.

Algorithm 5: The proposed mAHA.

- 1- Initialization phase based on Equations (1) and (2)
 - 2- While stop criterion is not satisfied
 - 3- Calculate the fitness of all individuals position \leftarrow the best search agent.
 - 4- Calculate guided foraging using Algorithm 2
 - 5- Calculate territorial foraging using Algorithm 3
 - 6- Calculate migration foraging using Algorithm 4
 - 7- $t = t + 1$
 - 8- End While
 - 9- Return the best criteria
-

4. Experimental Stage 1: Statistical Results for CEC'20 Test Suite

Several metaheuristic algorithms (MHs) are applied to solve different optimization problems such as adaptive gaining sharing knowledge [39], differential evolution through Bayesian [40], adapted evolutionary algorithm [41], and the hybrid gaining-sharing knowledge [42]. In this study, the IEEE CEC'20 test suite [43] is one of the proper test metrics, as it tests the optimizer performance over ten various search spaces, the performance of the proposed mAHA is assessed over the CEC'20 test suite on Dim 10 and Dim 20. The experimental results of the proposed mAHA on the CEC'20 test suite is compared to the results obtained from set of the state-of-the-art optimization algorithms, namely the slime mold algorithm (SMA) [22], Harris hawks optimization (HHO) [23], grey wolf optimizer (GWO) [24], whale optimization algorithm (WOA) [25], and the original AHA [21].

4.1. Statistical Results Analysis

Further, the statistical metrics includes the average, and the standard deviation (STD) methods are applied for the best-so-far solutions reached in each run. Tables 1 and 2 exhibit these statistical results for the proposed mAHA algorithm and the other competitors for each CEC'20 test suite with the Dim 10 and Dim 20 respectively. As the CEC'20 test suite are minimization problems, the best results are the lowest values. The introduced mAHA approach outperformed the other comparative algorithms in solving most of the CEC'20 test suite in terms of mean and STD. Consequently, the proposed mAHA algorithm achieved the first rank in the Friedman mean rank-sum test. Moreover, Table 3 reports the computation time that has been obtained by the proposed mAHA and the competitor algorithms for 30 runs with Dim 10 and Dim 20 on the CEC'20 test suite.

Table 1. The statistical results on the CEC’20 test suite that have been obtained by the proposed mAHA and the competitor algorithms for 30 runs with Dim 10.

Functions	SMA		GWO		WOA		HHO		AHA		mAHA	
	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
F1	9.34×10^3	3.93×10^3	3.80×10^7	1.08×10^8	2.75×10^6	5.04×10^6	6.14×10^5	2.58×10^5	5.64×10^1	$5.45E \times 10^1$	1.14×10^{-1}	5.49×10^1
F2	1.58×10^3	1.58×10^2	1.60×10^3	1.74×10^2	2.26×10^3	3.85×10^2	2.03×10^3	2.69×10^2	1.46×10^1	4.19×10^1	1.24×10^1	1.95×10^1
F3	7.24×10^2	4.89×10	7.32×10^2	1.27×10^1	7.66×10^2	1.47×10^1	7.89×10^2	1.93×10^1	5.61×10^1	1.56×10^1	1.03×10^1	1.18×10^1
F4	1.90×10^3	3.95×10^{-1}	1.90×10^3	2.46×10^1	1.91×10^3	2.77×10^1	1.91×10^3	2.75×10^1	3.77×10^1	4.77×10^{-1}	4.29×10^1	5.12×10^{-2}
F5	7.14×10^3	5.13×10^3	8.17×10^3	5.51×10^3	2.15×10^5	2.38×10^5	6.00×10^4	4.42×10^4	2.25×10^1	4.32×10^1	1.31×10^1	4.32×10^1
F6	1.60×10^3	3.08×10^{-1}	1.61×10^3	2.41×10^1	1.61×10^3	8.51×10^1	1.61×10^3	9.13×10^1	1.00×10^1	5.20×10^1	1.75×10^1	3.55×10^1
F7	2.52×10^3	3.31×10^2	8.00×10^3	4.41×10^3	5.66×10^4	5.03×10^4	2.03×10^4	2.66×10^4	1.38×10^1	4.83×10^1	1.19×10^1	3.06×10^1
F8	2.31×10^3	2.30×10^3	2.31×10^3	2.30×10^3	2.37×10^3	2.31×10^3	2.31×10^3	2.31×10^3	6.59×10^1	4.50×10^1	3.85×10^1	4.44×10^1
F9	2.76×10^3	6.12×10	2.74×10^3	1.15×10^1	2.76×10^3	5.75×10^1	2.81×10^3	1.24×10^2	1.38×10^1	3.27×10^1	5.62×10^1	2.86×10^1
F10	2.93×10^3	2.61×10^1	2.94×10^3	1.49×10^1	2.95×10^3	1.18×10^1	2.92×10^3	2.44×10^1	2.48×10^1	2.17×10^1	1.32×10^1	1.16×10^1
Friedman	6.2		2.3		4.7		6.8		3.1		1.2	
Rank	5		2		4		6		3		1	

Table 2. The statistical results on the CEC’20 test suite have been obtained by the proposed mAHA and the competitor algorithms for 30 runs with Dim 20.

Functions	SMA		GWO		WOA		HHO		AHA		mAHA	
	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD	Mean	STD
F1	9.54×10^3	3.96×10^3	3.83×10^7	1.09×10^8	2.78×10^6	5.09×10^6	6.18×10^5	2.65×10^5	5.74×10^1	5.55×10^1	1.24×10^{-1}	5.52×10^1
F2	1.70×10^3	1.61×10^2	1.65×10^3	1.77×10^2	2.29×10^3	3.89×10^2	2.08×10^3	2.78×10^2	1.56×10^1	4.29×10^1	1.29×10^1	2.05×10^1
F3	7.44×10^2	4.92×10^1	7.36×10^2	1.29×10^1	7.68×10^2	1.49×10^1	7.92×10^2	2.03×10^1	5.68×10^1	1.66×10^1	1.09×10^1	1.28×10^1
F4	2.10×10^3	3.98×10^{-1}	1.93×10^3	2.48×10^1	1.96×10^3	2.79×10^1	1.98×10^3	2.85×10^1	3.78×10^1	4.80×10^{-1}	4.33×10^1	5.52×10^{-2}
F5	7.34×10^3	5.18×10^3	8.19×10^3	5.56×10^3	2.18×10^5	2.40×10^5	6.09×10^4	4.47×10^4	2.28×10^1	4.42×10^1	1.38×10^1	4.42×10^1
F6	1.80×10^3	3.128×10^{-1}	1.66×10^3	2.48×10^1	1.66×10^3	8.58×10^1	1.69×10^3	9.19×10^1	1.04×10^1	5.33×10^1	1.79×10^1	3.65×10^1
F7	2.72×10^3	3.378×10^2	8.03×10^3	4.46×10^3	5.68×10^4	5.10×10^4	2.08×10^4	2.65×10^4	1.40×10^1	4.90×10^1	1.28×10^1	3.10×10^1
F8	2.51×10^3	2.34×10^3	2.36×10^3	2.36×10^3	2.39×10^3	2.30×10^3	2.39×10^3	2.37×10^3	6.62×10^1	4.62×10^1	3.99×10^1	4.45×10^1
F9	2.90×10^3	6.188×10^1	2.77×10^3	1.18×10^1	2.79×10^3	5.85×10^1	2.91×10^3	1.29×10^2	1.43×10^1	3.38×10^1	5.77×10^1	2.90×10^1
F10	3.13×10^3	2.68×10^1	2.97×10^3	1.52×10^1	3.09×10^3	1.20×10^1	2.99×10^3	2.49×10^1	2.52×10^1	2.29×10^1	1.44×10^1	1.30×10^1
Friedman	6.4		2.6		4.9		7.1		3.4		1.4	
Rank	5		2		4		6		3		1	

Table 3. The CPU time that has been obtained by the proposed mAHA and the competitor algorithms for 30 runs with Dim 10 and Dim 20 on the CEC'20 test suite.

Functions	SMA		GWO		WOA		HHO		AHA		mAHA	
	Dim 10	Dim 20	Dim 10	Dim 20	Dim 10	Dim 20	Dim 10	Dim 20	Dim 10	Dim 20	Dim 10	Dim 20
F1	6.24×10^1	6.63×10^1	6.40×10^1	6.77×10^1	6.20×10^1	6.34×10^1	6.24×10^1	6.48×10^1	6.25×10^1	6.50×10^1	6.35×10^1	6.98×10^1
F2	3.38×10^1	3.50×10^1	3.40×10^1	3.60×10^1	3.30×10^1	3.55×10^1	3.28×10^1	3.45×10^1	3.41×10^1	3.55×10^1	3.56×10^1	3.70×10^1
F3	3.98×10^1	4.10×10^1	4.08×10^1	4.20×10^1	4.02×10^1	4.30×10^1	4.03×10^1	4.20×10^1	4.11×10^1	4.40×10^1	4.20×10^1	4.50×10^1
F4	7.30×10^{-1}	7.40×10^{-1}	7.20×10^{-1}	7.50×10^{-1}	7.10×10^{-1}	7.40×10^{-1}	7.15×10^{-1}	7.40×10^{-1}	7.29×10^{-1}	7.40×10^{-1}	7.40×10^{-1}	7.60×10^{-1}
F5	4.24×10^1	4.43×10^1	4.34×10^1	4.54×10^1	4.20×10^1	4.42×10^1	4.20×10^1	4.44×10^1	4.34×10^1	4.54×10^1	4.40×10^1	4.64×10^1
F6	8.07×10^1	8.17×10^1	8.27×10^1	8.47×10^1	8.17×10^1	8.27×10^1	8.02×10^1	8.19×10^1	8.20×10^1	8.47×10^1	8.47×10^1	8.67×10^1
F7	4.34×10^1	4.54×10^1	4.37×10^1	4.57×10^1	4.30×10^1	4.44×10^1	4.30×10^1	4.54×10^1	4.37×10^1	4.50×10^1	4.50×10^1	4.74×10^1
F8	4.18×10^1	4.30×10^1	4.20×10^1	4.34×10^1	4.24×10^1	4.44×10^1	4.29×10^1	4.44×10^1	4.30×10^1	4.48×10^1	4.40×10^1	4.60×10^1
F9	8.40×10^1	8.60×10^1	8.43×10^1	8.67×10^1	8.47×10^1	8.70×10^1	8.35×10^1	8.57×10^1	8.45×10^1	8.70×10^1	8.55×10^1	8.77×10^1
F10	3.71×10^1	3.81×10^1	3.69×10^1	3.82×10^1	3.66×10^1	3.81×10^1	3.61×10^1	3.77×10^1	3.69×10^1	3.83×10^1	3.74×10^1	3.85×10^1

4.2. Convergence Performance Analysis

The performance of the proposed mAHA and other competitors evaluated on the CEC'20 test suite, the results are explained graphically with the convergence curves as shown in Figure 1. According to the convergence plots, the proposed mAHA reached a stable point over most of the test methods. From convergence plots the fast convergence refers to the optimal solution. Thus, the introduced mAHA method considers an applicable approach to tackle different optimization problems that need fast computing i.e., the online optimization problems.

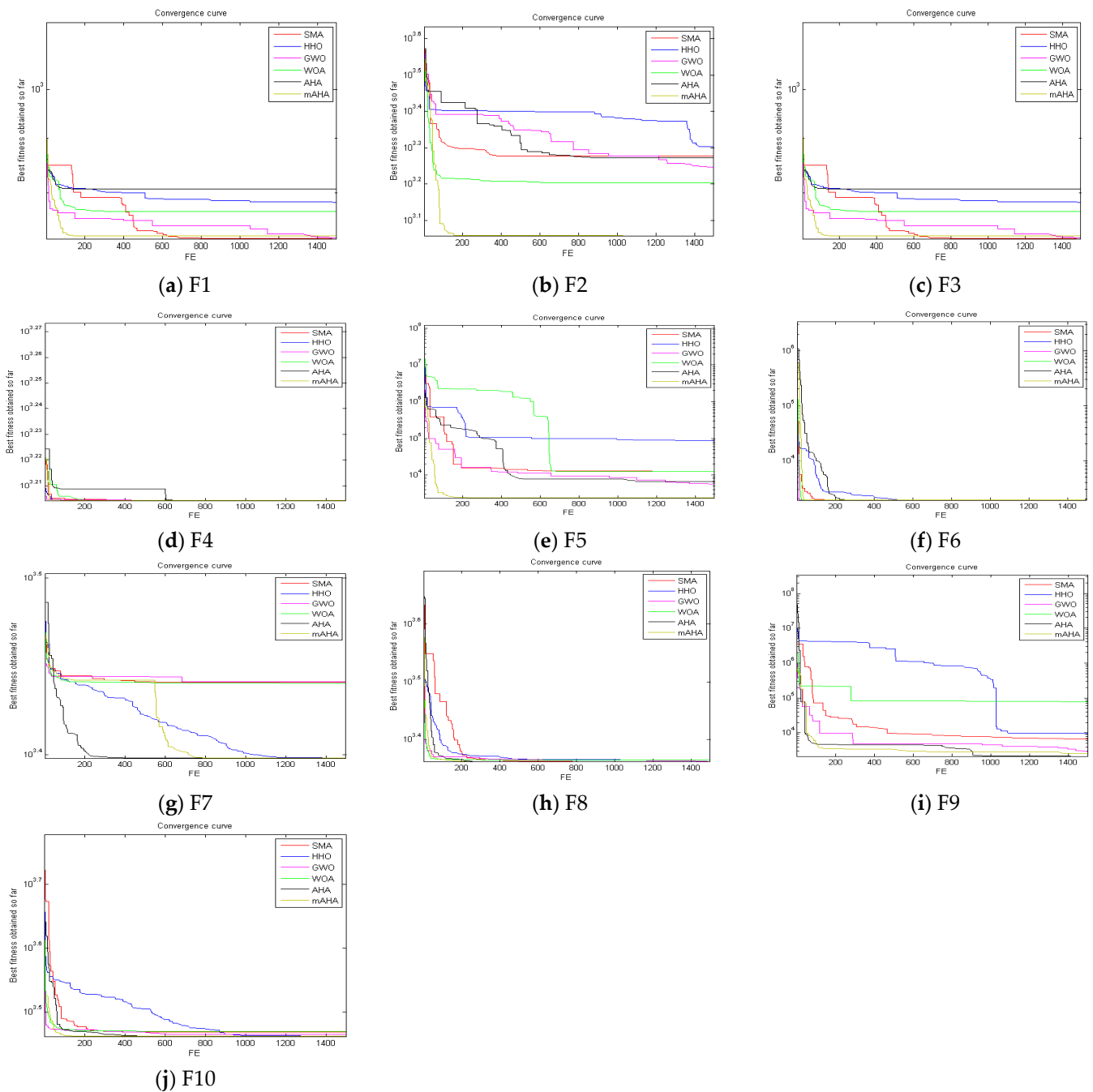


Figure 1. The convergence curves on the CEC'20 test suite have been obtained by the proposed mAHA and the competitor algorithms for 30 runs with Dim10.

4.3. Boxplot Behavior Analysis

Boxplots are employed to exhibit the data distribution. The distribution reflects the local minimum of test functions. Furthermore, boxplots consider an effective presenting method for data distributions in quartiles, where the obtained maximum and minimum data points are represented by the Boxplot whisker's edges. Furthermore, the higher the level of data agreement, the narrower the boxplot, Figure 2. Boxplot introduces the results for ten functions, Dim10. It is observed that the mAHA algorithm reaches the best results compared to the other algorithms.

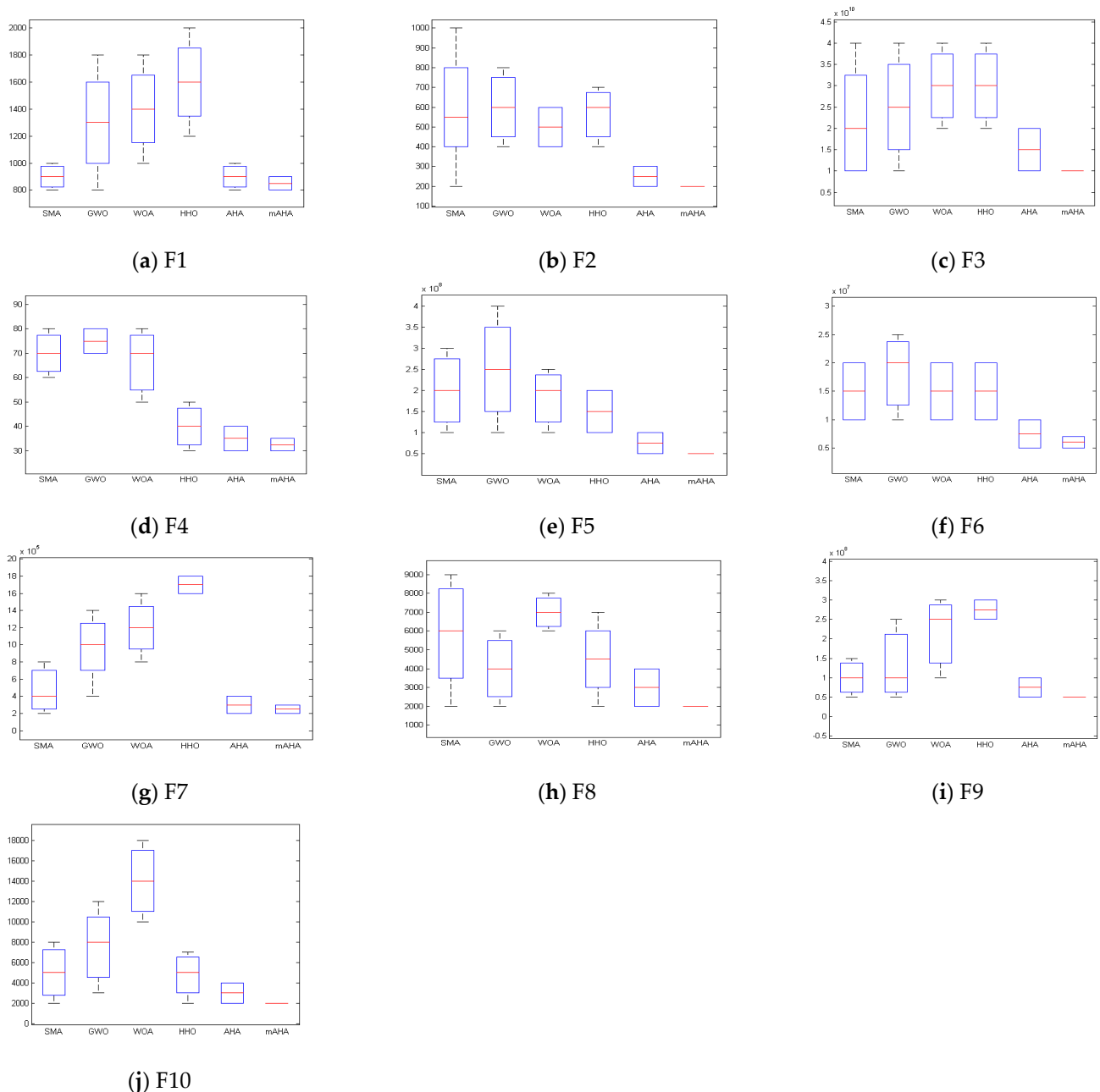


Figure 2. The box-plot curves on the CEC'20 test suite that have been obtained by the proposed mAHA and the competitor algorithms for 30 runs with Dim10.

4.4. Qualitative Metrics Analysis

The optimizer solutions behavior reflects a stable analysis about the algorithm performance and behavior through the search process. The qualitative analysis of the proposed

mAHA introduced in Figure 3, shows the agent’s behaviors, 3D views of the functions, average fitness history, search history, and convergence curves.

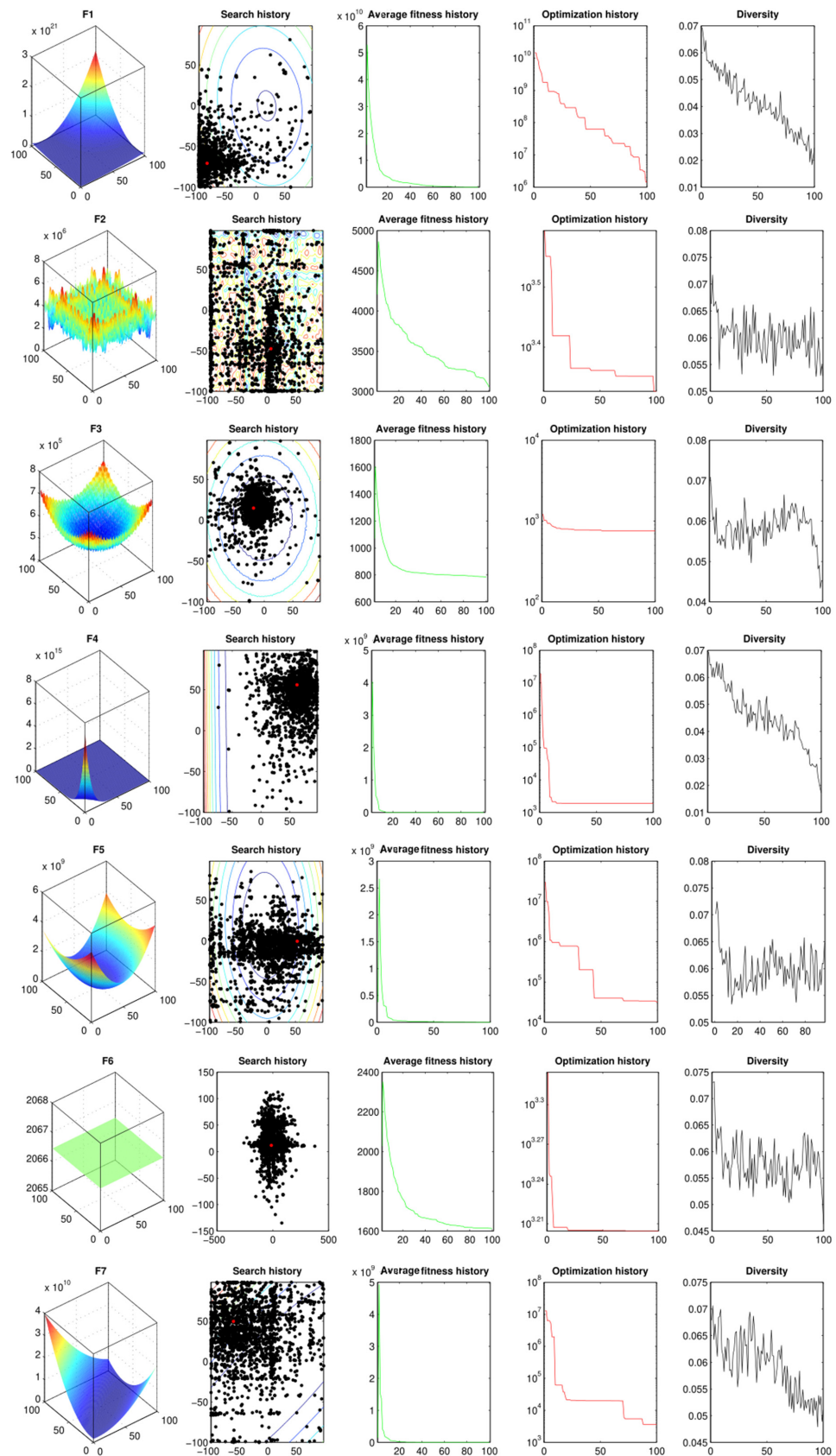


Figure 3. Cont.

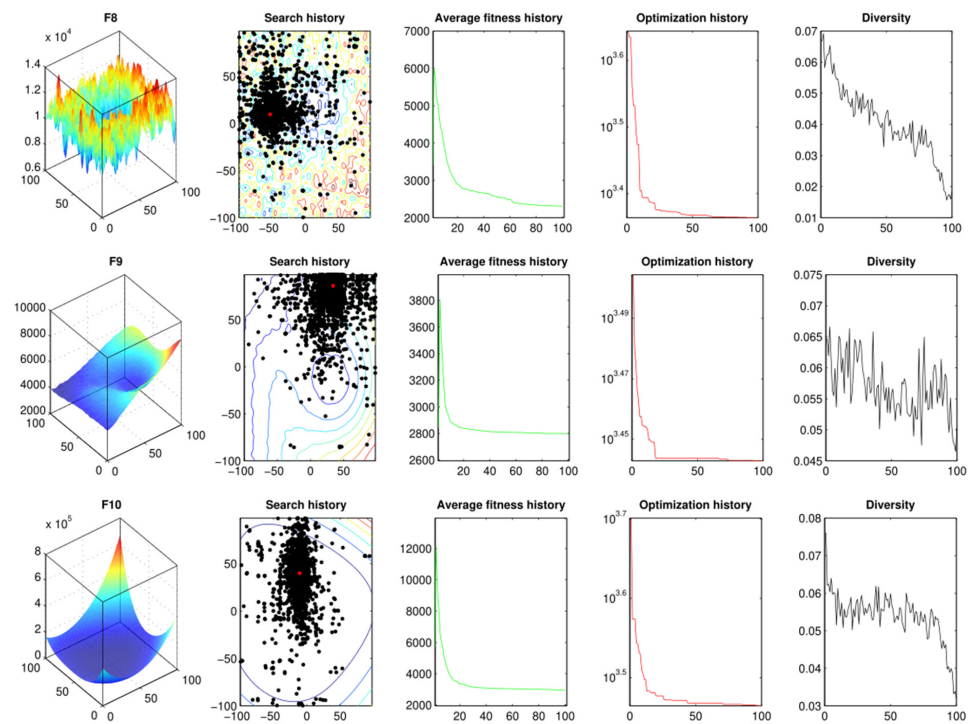


Figure 3. The qualitative metrics on the CEC’20 test suite have been obtained by the proposed mAHA.

The next points are worthwhile from the visualization curves such as convergence, boxplots, and qualitative analysis:

- (1) Search history: The second column in Figure 3 shows the agents’ search history from the beginning to the last iteration. Furthermore, the problem search space is formed on a contour line, it reflects the gradient from blue to red color lines indicating a higher fitness value. The introduced mAHA approach can reach the positions with the higher fitness values, according to the search history.
- (2) Average fitness history: The third column in Figure 3 demonstrates the average fitness value. From this figure, the agents’ overall behavior is represented by the fitness history as well as their contribution in the optimization process.
- (3) Accordingly, the performance of mAHA approach is assessed against the other competitors on CEC’20 test suite. The performance of the proposed mAHA is evaluated using both quantitative and qualitative indicators for mAHA. According to Tables 1–3, the proposed mAHA method has reached near/optimal results for convergence and the highest fitness value. The graphical boxplot and minimum convergence curve are shown in Figures 1 and 2 respectively. These graphical representations demonstrate the stable performance of the proposed mAHA algorithm as introduced in Figure 3, which indicate that the introduced method is dependable for a real situation and are drawn from the test metrics.

5. Stage 2: Maximum Power Point Tracking

The PV system is shown in Figure 4. It includes 36 photovoltaic panels forming 6 arrays, DC-DC converter, controller, and 480 V battery bank. The specification of the PV system is presented in Table 4.

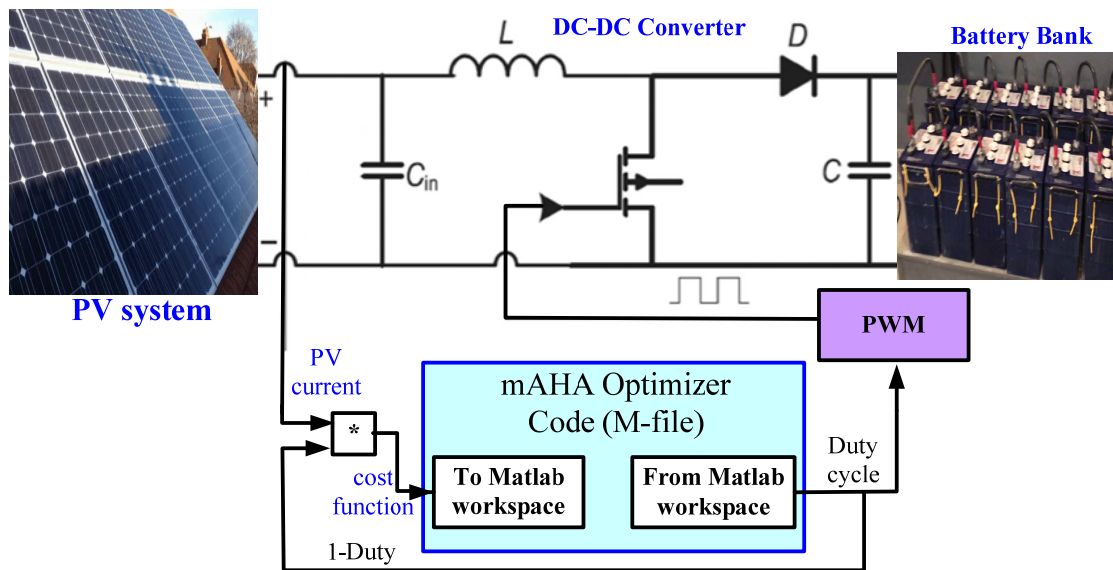


Figure 4. Schematic diagram for PV system with MPPT.

Table 4. Specifications of the PV system.

Item	Specification
Maximum power of the panel	200 W
PV current at MPP	7.61 A
PV voltage at MPP	26.3 V
No. of arrays	6
No. of series PV panels per array	3
No. of strings in array	2
Battery bank voltage	480 V

The duty cycle is the key to control the DC-DC converter to boost the PV power. It may be defined as follows:

$$V_{batt} = \frac{1}{1 - D} \times V_{PV} \tag{21}$$

where V_{pv} and V_{batt} represent the PV voltage and battery voltage respectively.

The PV voltage may be defined as follows:

$$V_{PV} = V_{batt}(1 - D) \tag{22}$$

where D is the duty cycle.

Referring to the above relation, with constant battery voltage, the PV voltage is proportional with $(1 - D)$. Therefore, as an alternative of utilizing voltage sensor, it can be using the value of $(1 - D)$ to replace the PV voltage. Consequently, the objective function required to be maximum can be defined as follows.

$$f(D) = I_{PV} \times (1 - D) \text{ subjected to } 0 \leq D \leq 1 \tag{23}$$

where I_{PV} denotes PV current, and the duty cycle (D) is selected to be the decision variable during the optimization process.

Two shadow scenarios have been considered to assess the suggested mAHA global MPPT technique. Figure 5 and Table 5 show the PV characteristics of the two scenarios. Altering shadow pattern is conducted to change the location of global MPP to assess the consistency of the suggested mAHA.

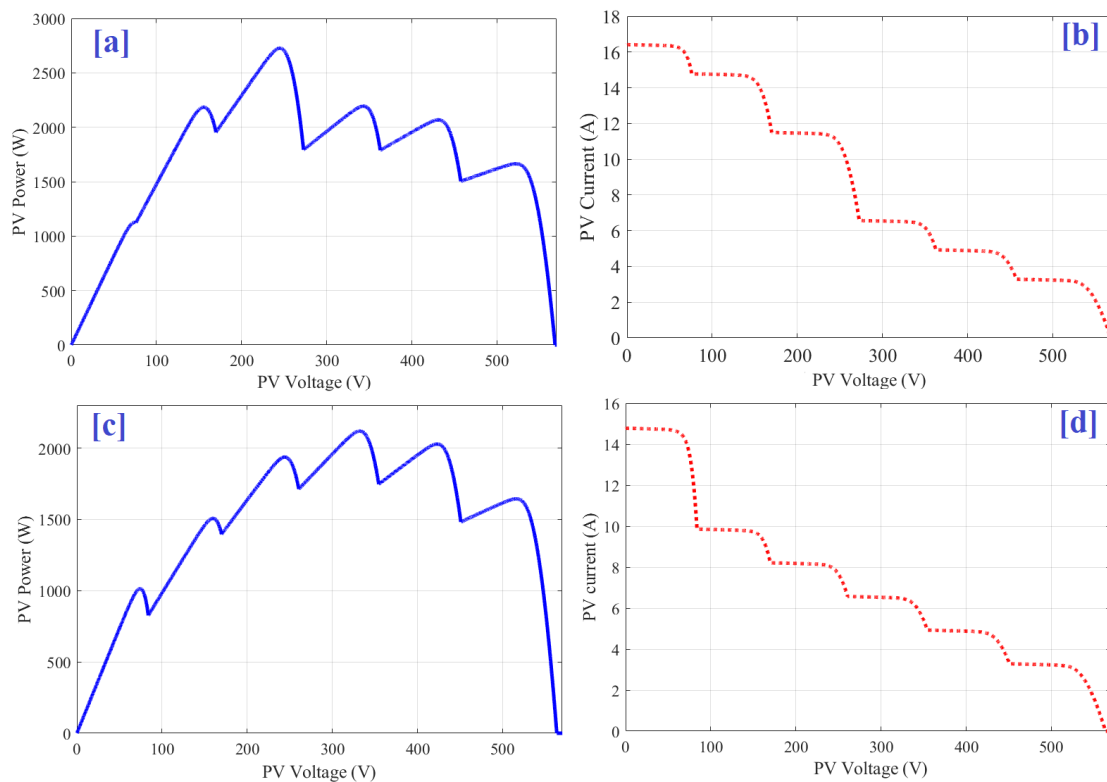


Figure 5. The details of shadow patterns. (a) *P-V* curve first scenario, (b) *I-V* curve first scenario, (c) *P-V* curve second scenario, and (d) *I-V* curve second scenario.

Table 5. Specification of shadow patterns and data at MPP.

Irradiance Intensity (W/m ²) Six Series-Connected PV Arrays							Data at MPP			
Pattern	Array 1	Array 2	Array 3	Array 4	Array 5	Array 6	Current (A)	Voltage (V)	Power (W)	Duty
1st scenario	1000	900	700	400	300	200	11.13	245	2725.9	0.4896
2nd scenario	900	600	500	400	300	200	6.38	332.28	2119.4	0.3077

To allow a fair comparison, both the number of populations (5) and iterations (20) were kept constants for all methods considered. During the optimization process, the product of PV current and (1 – D) was used as the objective function to be maximized. The decision variable is the duty cycle of DC-DC. To prove the consistency of the proposed mAHA, the studied algorithms were executed 30 times. The statistical analysis of the considered algorithms is shown in Table 6. The details of the 30 runs for both shadow scenarios are shown in Table 7.

Table 6. Statistical assessments of considered algorithms for both shadow scenarios.

	SMA	HHO	GWO	WOA	AHA	mAHA
1st Scenario						
Best cost function	5.678486	5.678486	5.678486	5.678486	5.678486	5.678486
Maximum PV power (W)	2725.673	2725.673	2725.673	2725.673	2725.673	2725.673
Worst	4.307119	4.555636	4.307109	4.307124	4.30633	5.659904
Mean	5.632443	5.603838	5.412551	5.63269	5.30468	5.676989
Average PV power (W)	2067.417	2186.705	2067.413	2067.419	2067.039	2716.754
STD	0.246108	0.277208	0.482421	0.246152	0.516302	0.004308

Table 6. Cont.

	SMA	HHO	GWO	WOA	AHA	mAHA
Median	5.678466	5.678475	5.67825	5.67848	5.659086	5.678435
Variance	0.060569	0.076844	0.23273	0.060591	0.266568	1.86×10^{-5}
Average time (s)	69.39628	169.6144	70.38237	70.86129	74.29785	73.42117
2nd Scenario						
Best cost function	4.415491	4.415491	4.415491	4.415491	4.415491	4.415491
Maximum PV power (W)	2027.256	2027.74	2027.735	2027.742	2017.267	2117.477
Worst	4.223449	4.224458	4.224447	4.224463	4.202639	4.41141
Mean	4.363836	4.402683	4.383512	4.40273	4.37282	4.415039
Average PV power (W)	2094.641	2113.288	2104.086	2113.31	2098.954	2119.219
STD	0.08421	0.047632	0.071131	0.047644	0.076916	0.000996
Median	4.415443	4.415484	4.415365	4.415483	4.41405	4.415467
Variance	0.007091	0.002269	0.00506	0.00227	0.005916	9.92×10^{-7}
Average time (s)	69.3757	159.2509	69.37294	69.5756	73.37224	73.80433

Table 7. Cost function values during optimization process.

	SMA	HHO	GWO	WOA	AHA	mAHA	SMA	HHO	GWO	WOA	AHA	mAHA
	1st Shadow Scenario						2nd Shadow Scenario					
1	5.6785	5.6761	5.671	5.6784	5.6543	5.6785	4.4154	4.4155	4.2245	4.4155	4.239	4.4138
2	4.3071	5.6785	5.6785	5.6785	5.6785	5.6783	4.4154	4.4155	4.415	4.4155	4.4155	4.4155
3	5.6767	5.6674	5.6784	4.3071	5.6785	5.6785	4.4154	4.4153	4.4155	4.4155	4.4155	4.4155
4	5.6785	5.6785	5.6785	5.6785	5.6781	5.6785	4.4155	4.4154	4.4155	4.2245	4.4132	4.4151
5	5.6782	5.6784	4.3071	5.6785	4.5697	5.6771	4.2244	4.4155	4.4154	4.4155	4.2242	4.4155
6	5.6783	5.6778	5.6718	5.6785	5.6765	5.6777	4.4155	4.4155	4.4155	4.4154	4.4155	4.4155
7	5.6781	5.6785	5.6784	5.6785	5.6783	5.6783	4.4155	4.4155	4.4154	4.4155	4.4154	4.4155
8	5.6731	5.6785	5.6771	5.6785	5.6783	5.6784	4.4155	4.4155	4.4153	4.4154	4.4152	4.4155
9	5.6785	5.6785	5.6785	5.6785	4.4733	5.6785	4.4155	4.4155	4.2245	4.2245	4.4154	4.4154
10	5.6785	5.6785	4.5696	5.6785	5.678	5.6784	4.4155	4.4155	4.4155	4.4155	4.4155	4.4155
11	5.6782	5.6785	4.5697	5.6785	5.6784	5.6785	4.4155	4.2245	4.4144	4.4155	4.4155	4.4143
12	5.6785	5.6785	5.6782	5.6785	5.6785	5.6784	4.4153	4.4155	4.4154	4.4155	4.4137	4.4155
13	5.6785	5.6784	5.6785	5.6785	4.5647	5.6783	4.4153	4.4155	4.4154	4.4155	4.4118	4.4155
14	5.6785	5.6785	4.5697	5.6784	4.5697	5.6599	4.4155	4.2245	4.415	4.4155	4.4154	4.4155
15	5.6785	4.5556	5.6764	5.6785	5.6776	5.6785	4.2245	4.4155	4.4152	4.4154	4.4154	4.4155
16	5.6785	5.6785	5.6784	5.6785	5.6578	5.6749	4.2244	4.4155	4.4151	4.4155	4.4144	4.4114
17	5.6785	5.6785	5.6783	5.6785	5.6604	5.6784	4.2244	4.4146	4.4155	4.4155	4.4154	4.4155
18	5.6784	5.6785	5.6783	5.6785	4.5697	5.6784	4.4155	4.4155	4.4155	4.4155	4.4119	4.4154
19	5.6785	5.6785	5.6759	5.6785	4.3063	5.6785	4.3963	4.4152	4.4155	4.4155	4.4153	4.4154
20	5.6784	5.6785	5.6784	5.6766	5.3566	5.6785	4.4152	4.4154	4.4152	4.4155	4.3922	4.4119
21	5.6785	5.6784	5.6785	5.6785	5.6146	5.6767	4.2243	4.4155	4.2244	4.4155	4.2095	4.4155
22	5.6785	5.678	5.6783	5.6782	5.5155	5.6785	4.4155	4.4154	4.4152	4.4155	4.4084	4.4155

Table 7. Cont.

	SMA	HHO	GWO	WOA	AHA	mAHA	SMA	HHO	GWO	WOA	AHA	mAHA
	1st Shadow Scenario						2nd Shadow Scenario					
23	5.6783	5.6783	4.6416	5.6785	4.5661	5.6785	4.2234	4.4153	4.2245	4.4154	4.4155	4.4155
24	5.6785	5.6784	5.6778	5.6784	5.6785	5.6785	4.4155	4.4155	4.4155	4.4154	4.2245	4.4154
25	5.6785	4.5778	5.6783	5.6785	5.6722	5.6784	4.4155	4.4155	4.4155	4.4153	4.406	4.4155
26	5.678	5.6785	5.6784	5.6785	5.6785	5.6784	4.4155	4.4153	4.4155	4.4155	4.2026	4.4146
27	5.6785	5.6785	5.6785	5.6785	5.4584	5.6624	4.2245	4.4155	4.4155	4.4155	4.3614	4.4155
28	5.6778	5.6782	4.5696	5.6785	5.6741	5.6782	4.2244	4.4155	4.2245	4.4155	4.4113	4.4155
29	5.6785	5.6781	5.6751	5.6785	4.5697	5.6785	4.4155	4.4152	4.4155	4.4155	4.4154	4.4152
30	5.6784	5.6785	4.5697	5.6784	4.5498	5.677	4.4155	4.4155	4.4154	4.4155	4.2245	4.4146

The two shadow scenarios are shown in Table 7, where the proposed mAHA has the best performance compared to other algorithms. For the first scenario, the average PV power values varied between 2716.754 W and 2067.039 W. mAHA achieves the maximum PV power of 2716.754 W, which is matched by HHO (2186.705 W). The minimum PV power of 2067.039 W is achieved by AHA. Thus, the proposed mAHA increases the PV power by 31.3% compared to the original AHA. The values of STD vary from 0.004308 to 0.516302. The minimum STD of 0.004308 is achieved by mAHA, followed by SMA (0.246108). The worst value of STD of 0.516302 is achieved by AHA. For the second shadow scenario, the average PV power values vary from 2119.219 W to 2094.641 W. mAHA achieves the maximum PV power of 2119.219 W, followed by WOA (2113.31 W). The minimum PV power of 2094.641 W is achieved by SMA. The values of STD vary from 0.000996 to 0.08421. The minimum STD of 0.000996 is achieved by mAHA, followed by HHO (0.047632). The worst value of STD of 0.08421 is obtained by SMA.

The mean cost function variation during optimization process for first and second shadow scenarios are presented in Figures 6 and 7 respectively.

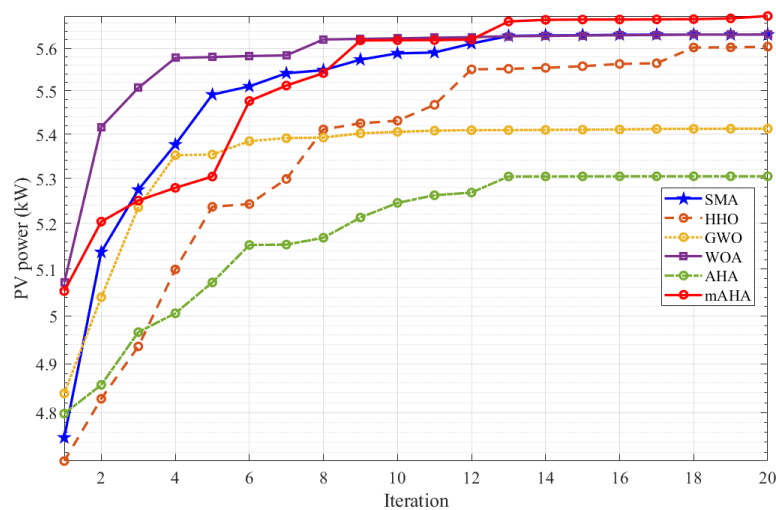


Figure 6. Mean cost function variation during optimization process for first shadow scenario.

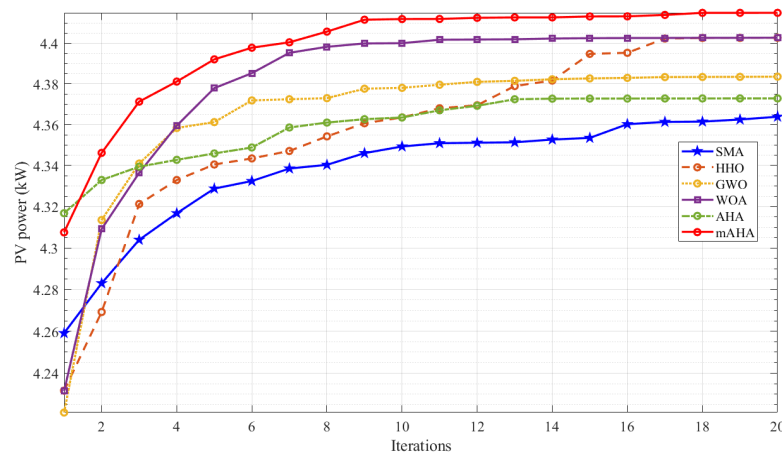


Figure 7. Mean cost function variation during optimization process for second shadow scenario.

Table 8 shows the results of the analysis of variance (ANOVA), and Figure 8 shows the corresponding ranking. If the value of F is greater than the p value, the null hypothesis is true. The data collected show that the p-value is much lower than the F-value, indicating a significant difference between the results. As shown in Figure 8, the mAHA can outperform the other commonly used methods. The mAHA has the smallest range of variance and the largest mean fitness (maximization problem), indicating its resilience and accuracy.

Table 8. ANOVA results for first shadow scenario.

Source	df	SS	MS	F	p-Value
Columns	5	3.3452	0.6690	5.56	8.8039×10^{-5}
Error	174	20.9198	0.1202		
Total	179	24.2651			

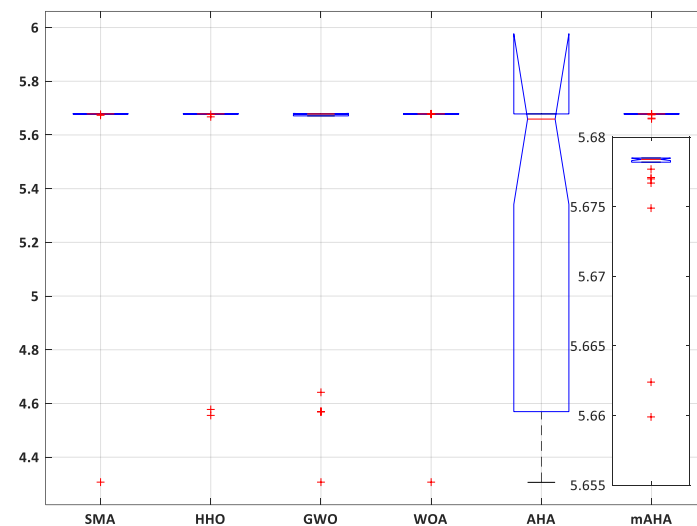


Figure 8. ANOVA ranking for first shadow scenario.

A Tukey Honestly Significant Difference (Tukey HSD) post hoc analysis was performed to support the ANOVA results. The findings are shown in Figure 9. The mAHA has the greatest mean fitness. After the mAHA, the WAO and SMA provided good results.

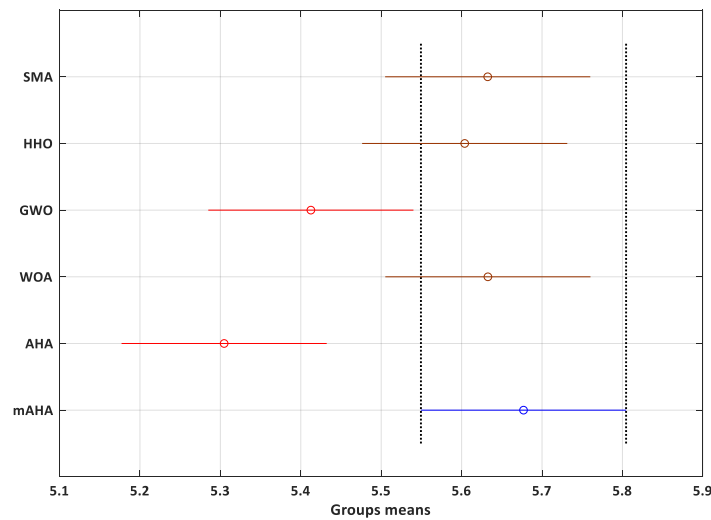


Figure 9. Tukey test for first shadow scenario.

Table 9 shows the ANOVA test results for the second case, and Figure 10 shows the corresponding ranking. The obtained results show that the p -value is smaller than the F value, indicating a significant difference between the outcomes. As shown in Figure 10, the mAHA may outperform the other commonly used method. The mAHA has the smallest variance range and the greatest mean fitness (maximization problem), indicating its resilience and accuracy.

Table 9. ANOVA results for the second shadow scenario.

Source	df	SS	MS	F	p -Value
Columns	5	0.0592	0.0118	3.04	0.0118
Error	174	0.6782	0.0039		
Total	179	0.7374			

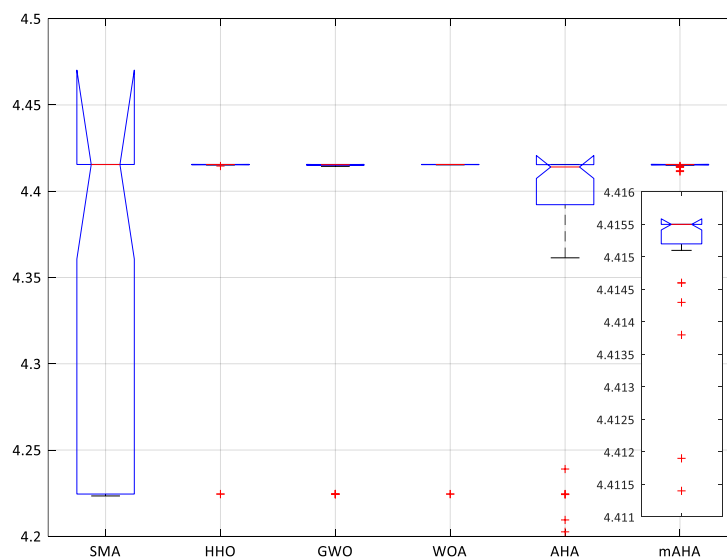


Figure 10. ANOVA ranking for the second shadow scenario.

To support the ANOVA results for this case, a Tukey Honestly Significant Difference (Tukey HSD) post hoc analysis was performed. The findings are shown in Figure 11. Like the previous case, the mAHA has the greatest mean fitness. After the mAHA, the WAO and HHO provided good results.

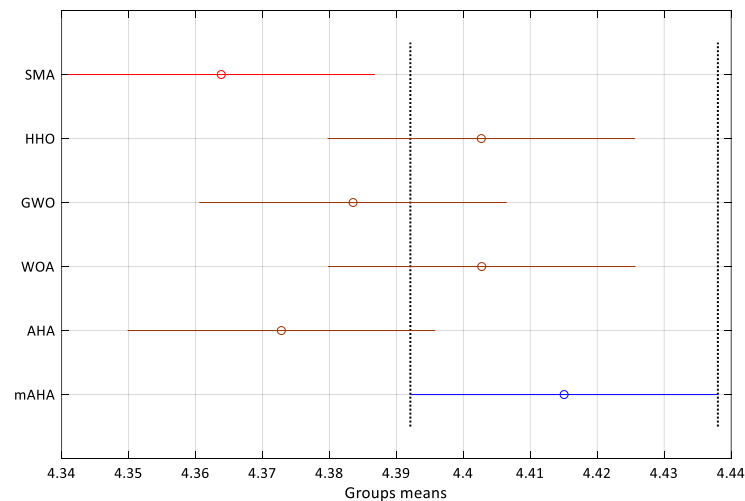


Figure 11. Tukey test for the second shadow scenario.

6. Discussion

The aim of this study is to propose an effective optimization method for solving the optimization problem and finding the global maximum power point (MPP). In many occurrences, the proposed mAHA method achieved better or similar results.

The proposed mAHA method provides the following merits:

1. mAHA is well able to resolve global optimization issues based on the CEC'22 test suite. mAHA generates optimization solutions with better fitness values than the other competitor algorithms as shown in Tables 1–3.
2. The AHA algorithm is combined with genetic operators to enhance the convergence ability of the mAHA method is achieved as demonstrated in Figures 1 and 2.
3. The proposed mAHA was used for the first time to find the global maximum power point (MPP), includes 36 photovoltaic panels forming 6 arrays, DC-DC converter, controller, and 480 V battery bank.
4. Table 6 shows the results of the analysis of variance (ANOVA), and Figure 8 shows the corresponding ranking. As shown in Figures 8–11, the mAHA can outperform the other commonly used methods. The mAHA has the smallest range of variance and the largest mean fitness (maximization problem), indicating its resilience and accuracy.
5. The obtained results show the superiority of the proposed single sensor-based MPPT method for PV systems. The scalability analysis demonstrated the robustness and flexibility of the proposed mAHA method.

Along with advantages, the proposed mAHA also has some limitations, which are detailed below:

1. Despite eminent applications, AHA is still attributed for its slow convergence and stagnancy issues when employed on high-dimensional problems.
2. The obtained solutions generated by mAHA may change each time it is run because it is an optimization strategy based on randomization. As a result, there is no assurance that the features subset chosen in one run will be present in another.
3. The performance of the proposed mAHA method on complex and high dimensional problems may be worse according to the several mutations.

7. Conclusions

A modified version of the artificial hummingbird algorithm (AHA) has been proposed in this paper. It introduces the mechanisms of genetic operators (crossover and mutation selection) to enhance AHA's performance in increasing the diversity of the population and avoiding local searches. After demonstration the superiority of mAHA, it has been used for first time to mitigate the shadow condition and extracting the global maximum power

point (MPP) for photovoltaic (PV) system. The proposed MPP tracking (MPPT) technique needs only a single current sensor. Therefore, the cost of the controller will be reduced. Two shadow scenarios are used to evaluate the proposed single-sensor MPPT technique. For the first scenario, the average PV power values fluctuated between 2716.754 W and 2067.039 W. mAHA achieves the maximum PV power of 2716.754 W flowed by HHO (2186.705 W). The minimum PV power of 2067.039 W is obtained by AHA. Therefore, the proposed mAHA increased the PV power by 31.3% compared to the original AHA. For the second shadow scenario, the average PV power values fluctuated between 2119.219 W and 2094.641 W. mAHA achieves the maximum PV power of 2119.219 W flowed by WOA (2113.31W). The minimum PV power of 2094.641 W is obtained by SMA. The STD values varied between 0.000996 and 0.08421. According to the excellent efficiency obtained, the proposed mAHA can be applied in several real-world applications, such as object tracking, calculating solar cell parameters, electrical applications, hyperparameter optimization, and image segmentation.

Author Contributions: Conceptualization, E.H.H., H.R. and I.A.M.; Methodology, E.H.H. and H.R.; Software, E.H.H. and H.R.; Validation, I.A.M. and S.A.-S.; Formal analysis, H.A., E.H.H., H.R. and S.A.-S.; Investigation, H.R. and S.A.-S.; Resources, H.A., I.A.M. and S.A.-S.; Data curation, H.A. and I.A.M.; Writing—original draft, H.A., E.H.H., H.R., I.A.M. and S.A.-S.; Writing—review & editing, H.A., E.H.H., H.R., I.A.M. and S.A.-S.; Visualization, E.H.H. and H.R.; Supervision, E.H.H. and H.R.; Project administration, H.A.; Funding acquisition, H.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research work was funded by the Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia through the project number (IFPRC-78-135-2020) and King Abdulaziz University, DSR, Jeddah, Saudi Arabia.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors extend their appreciation to the Deputyship for Research & Innovation, Ministry of Education in Saudi Arabia for funding this research work through the project number (IFPRC-78-135-2020) and King Abdulaziz University, DSR, Jeddah, Saudi Arabia.

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

References

1. Houssein, E.H.; Mahdy, M.A.; Shebl, D.; Manzoor, A.; Sarkar, R.; Mohamed, W.M. An efficient slime mould algorithm for solving multi-objective optimization problems. *Expert Syst. Appl.* **2022**, *187*, 115870. [[CrossRef](#)]
2. Singh, N.; Houssein, E.H.; Singh, S.; Dhiman, G. Hssahho: A novel hybrid salp swarm-harris hawks optimization algorithm for complex engineering problems. *J. Ambient. Intell. Humaniz. Comput.* **2022**, *97*, 1–37.
3. Houssein, E.H.; Gad, A.G.; Wazery, Y.M.; Suganthan, P.N. Task scheduling in cloud computing based on meta-heuristics: Review, taxonomy, open challenges, and future trends. *Swarm Evol. Comput.* **2021**, *62*, 100841. [[CrossRef](#)]
4. Houssein, E.H.; Mahdy, M.A.; Blondin, M.J.; Shebl, D.; Mohamed, W.M. Hybrid slime mould algorithm with adaptive guided differential evolution algorithm for combinatorial and global optimization problems. *Expert Syst. Appl.* **2021**, *174*, 114689. [[CrossRef](#)]
5. Houssein, E.H.; Mahdy, M.A.; Fathy, A.; Rezk, H. A modified Marine Predator Algorithm based on opposition based learning for tracking the global MPP of shaded PV system. *Expert Syst. Appl.* **2021**, *183*, 115253. [[CrossRef](#)]
6. Lu, S.; Wang, C.; Fan, Y.; Lin, B. Robustness of building energy optimization with uncertainties using deterministic and stochastic methods: Analysis of two forms. *Build. Environ.* **2021**, *205*, 108185. [[CrossRef](#)]
7. Forrest, S. Genetic algorithms. *ACM Comput. Surv.* **1996**, *28*, 77–80. [[CrossRef](#)]
8. Eberhart, R.; Kennedy, J. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; Volume 4, pp. 1942–1948.
9. Teodorović, D. Bee colony optimization (BCO). In *Innovations in Swarm Intelligence*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 39–60.
10. Storn, R.; Kenneth, P. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **1997**, *11*, 341–359. [[CrossRef](#)]

11. Houssein, E.H.; Abdelkareem, D.A.; Emam, M.M.; Hameed, M.A.; Younan, M. An efficient image segmentation method for skin cancer imaging using improved golden jackal optimization algorithm. *Comput. Biol. Med.* **2022**, *149*, 106075. [[CrossRef](#)]
12. Houssein, E.H.; Emam, M.M.; Ali, A.A.; Suganthan, P.N. Deep and machine learning techniques for medical imaging-based breast cancer: A comprehensive review. *Expert Syst. Appl.* **2021**, *167*, 114161. [[CrossRef](#)]
13. Houssein, E.H.; Emam, M.M.; Ali, A.A. Improved manta ray foraging optimization for multi-level thresholding using COVID-19 CT images. *Neural Comput. Appl.* **2021**, *33*, 16899–16919. [[CrossRef](#)] [[PubMed](#)]
14. Houssein, E.H.; Abohashima, Z.; Elhoseny, M.; Mohamed, W.M. Hybrid quantum-classical convolutional neural network model for COVID-19 prediction using chest X-ray images. *J. Comput. Des. Eng.* **2022**, *9*, 343–363. [[CrossRef](#)]
15. Houssein, E.H.; Oliva, D.; Çelik, E.; Emam, M.M.; Ghoniem, R.M. Boosted sooty tern optimization algorithm for global optimization and feature selection. *Expert Syst. Appl.* **2023**, *213*, 119015. [[CrossRef](#)]
16. Houssein, E.H.; Hosney, M.E.; Mohamed, W.M.; Ali, A.A.; Younis, E.M. Fuzzy-based hunger games search algorithm for global optimization and feature selection using medical data. *Neural Comput. Appl.* **2022**, 1–25. [[CrossRef](#)] [[PubMed](#)]
17. Emam, M.M.; Houssein, E.H.; Ghoniem, R.M. A modified reptile search algorithm for global optimization and image segmentation: Case study brain MRI images. *Comput. Biol. Med.* **2023**, *152*, 106404. [[CrossRef](#)] [[PubMed](#)]
18. Hashim, F.A.; Hussain, K.; Houssein, E.H.; Mabrouk, M.S.; Al-Atabany, W. Archimedes optimization algorithm: A new metaheuristic algorithm for solving optimization problems. *Appl. Intell.* **2021**, *51*, 1531–1551. [[CrossRef](#)]
19. Houssein, E.H.; Çelik, E.; Mahdy, M.A.; Ghoniem, R.M. Self-adaptive Equilibrium Optimizer for solving global, combinatorial, engineering, and Multi-Objective problems. *Expert Syst. Appl.* **2022**, *195*, 116552. [[CrossRef](#)]
20. Houssein, E.H.; Saad, M.R.; Hashim, F.A.; Shaban, H.; Hassaballah, M. Lévy flight distribution: A new metaheuristic algorithm for solving engineering optimization problems. *Eng. Appl. Artif. Intell.* **2020**, *94*, 103731. [[CrossRef](#)]
21. Zhao, W.; Wang, L.; Mirjalili, S. Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications. *Comput. Methods Appl. Mech. Eng.* **2022**, *388*, 114194. [[CrossRef](#)]
22. Li, S.; Chen, H.; Wang, M.; Heidari, A.A.; Mirjalili, S. Slime mould algorithm: A new method for stochastic optimization. *Future Gener. Comput. Syst.* **2020**, *111*, 300–323. [[CrossRef](#)]
23. Heidari, A.A.; Mirjalili, S.; Faris, H.; Aljarah, I.; Mafarja, M.; Chen, H. Harris hawks optimization: Algorithm and applications. *Future Gener. Comput. Syst.* **2019**, *97*, 849–872. [[CrossRef](#)]
24. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey wolf optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [[CrossRef](#)]
25. Mirjalili, S.; Andrew, L. The whale optimization algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [[CrossRef](#)]
26. Krupnov, Y.A.; Krasilnikova, V.G.; Kiselev, V.; Yashchenko, A.V. The Contribution of Sustainable and Clean Energy to the Strengthening of Energy Security. *Front. Environ. Sci.* **2022**, *10*, 2505. [[CrossRef](#)]
27. Zsiborács, H.; Pintér, G.; Vincze, A.; Baranyai, N.H.; Mayer, M.J. The reliability of photovoltaic power generation scheduling in seventeen European countries. *Energy Convers. Manag.* **2022**, *260*, 115641. [[CrossRef](#)]
28. Karmouni, H.; Chouiekh, M.; Motahhir, S.; Qjidaa, H.; Jamil, M.O.; Sayyouri, M. A fast and accurate sine-cosine MPPT algorithm under partial shading with implementation using arduino board. *Clean. Eng. Technol.* **2022**, *9*, 100535. [[CrossRef](#)]
29. Bhukya, L.; Kedika, N.R.; Salkuti, S.R. Enhanced Maximum Power Point Techniques for Solar Photovoltaic System under Uniform Insolation and Partial Shading Conditions: A Review. *Algorithms* **2022**, *15*, 365. [[CrossRef](#)]
30. Boghdady, T.A.; Kotb, Y.E.; Aljumah, A.; Sayed, M.M. Comparative Study of Optimal PV Array Configurations and MPPT under Partial Shading with Fast Dynamical Change of Hybrid Load. *Sustainability* **2022**, *14*, 2937. [[CrossRef](#)]
31. Zafar, M.H.; Khan, N.M.; Mirza, A.F.; Mansoor, M. Bio-inspired optimization algorithms based maximum power point tracking technique for photovoltaic systems under partial shading and complex partial shading conditions. *J. Clean. Prod.* **2021**, *309*, 127279. [[CrossRef](#)]
32. Zafar, M.H.; Khan, N.M.; Mirza, A.F.; Mansoor, M.; Akhtar, N.; Qadir, M.U.; Khan, N.A.; Moosavi, S.K.R. A novel meta-heuristic optimization algorithm based MPPT control technique for PV systems under complex partial shading condition. *Sustain. Energy Technol. Assess.* **2021**, *47*, 101367.
33. Zafar, M.H.; Al-shahrani, T.; Khan, N.M.; Mirza, A.F.; Mansoor, M.; Qadir, M.U.; Khan, M.I.; Naqvi, R.A. Group teaching optimization algorithm based MPPT control of PV systems under partial shading and complex partial shading. *Electronics* **2020**, *9*, 1962. [[CrossRef](#)]
34. Huang, Y.-P.; Chen, X.; Ye, C.-E. A hybrid maximum power point tracking approach for photovoltaic systems under partial shading conditions using a modified genetic algorithm and the firefly algorithm. *Int. J. Photoenergy* **2018**, 7598653. [[CrossRef](#)]
35. El-Helw, H.M.; Magdy, A.; Marei, M.I. A Hybrid Maximum Power Point Tracking Technique for Partially Shaded Photovoltaic Arrays. *IEEE Access* **2017**, *5*, 11900–11908. [[CrossRef](#)]
36. El-Shafy, K.A.; Nafeh, A. An effective and safe charging algorithm for lead-acid batteries in PV systems. *Int. J. Energy Res.* **2011**, *35*, 733–740.
37. Camilo, J.C.; Guedes, T.; Fernandes, D.A.; Melo, J.; Costa, F.; Filho, A.J.S. A maximum power point tracking for photovoltaic systems based on Monod equation. *Renew. Energy* **2019**, *130*, 428–438. [[CrossRef](#)]
38. Zhou, Y.; Wang, Y.; Wang, K.; Kang, L.; Peng, F.; Wang, L.; Pang, J. Hybrid genetic algorithm method for efficient and robust evaluation of remaining useful life of supercapacitors. *Appl. Energy* **2020**, *260*, 114169. [[CrossRef](#)]
39. Mohamed, A.W.; Hadi, A.A.; Mohamed, A.K.; Awad, N.H. Evaluating the performance of adaptive gaining sharing knowledge based algorithm on CEC 2020 benchmark problems. In Proceedings of the 2020 IEEE Congress on Evolutionary Computation (CEC), Glasgow, UK, 19–24 July 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 1–8.

40. Biswas; Subhodip; Saha, D.; De, S.; Cobb, A.D.; Das, S.; Jalaian, B.A. Improving differential evolution through Bayesian hyperparameter optimization. In Proceedings of the 2021 IEEE Congress on Evolutionary Computation (CEC), Kraków, Poland, 28 June–1 July 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 832–840.
41. Saha, D.; Sallam, K.M.; De, S.; Mohamed, A.W. Framework of Ensemble Parameter Adapted Evolutionary Algorithm for Solving Constrained Optimization Problems. *Preprints* **2022**. [[CrossRef](#)]
42. Mohamed, A.W.; Hadi, A.A.; Agrawal, P.; Sallam, K.M.; Mohamed, A.K. Gaining-sharing knowledge based algorithm with adaptive parameters hybrid with imode algorithm for solving cec 2021 benchmark problems. In Proceedings of the 2021 IEEE Congress on Evolutionary Computation (CEC), Kraków, Poland, 28 June–1 July 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 841–848.
43. Houssein, E.H.; Rezk, H.; Fathy, A.; Mahdy, M.A.; Nassef, A.M. A modified adaptive guided differential evolution algorithm applied to engineering applications. *Eng. Appl. Artif. Intell.* **2022**, *113*, 104920. [[CrossRef](#)]

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