Article

Solar Photovoltaic Cell Parameter Identification Based on Improved Honey Badger Algorithm

Wenjing Lei 1, Qing He 1,2,*, Liu Yang 3,* and Hongzan Jiao 4,5

1 College of Big Data & Information Engineering, Guizhou University, Guiyang 550025, China; lei_wenjing@163.com
2 Guizhou Provincial Key Laboratory of Public Big Data, Guizhou University, Guiyang 550025, China
3 College of Public Administration, Guizhou University, Guiyang 550025, China
4 Department of Urban Planning, School of Urban Design, Wuhan University, Wuhan 430072, China; jiaohongzan@whu.edu.cn
5 Engineering Research Center of Human Settlements and Environment of Hubei Province, Wuhan 430072, China
* Correspondence: qhe@gzu.edu.cn (Q.H.); lyang3@gzu.edu.cn (L.Y.)

Abstract: Photovoltaic technology, which converts the sun’s light energy directly into electricity, can be used to make photovoltaic cells. The use of photovoltaic cells is centered on the idea of a low-carbon economy and green environmental protection, which effectively addresses the pollution problem in smart cities. Accurate identification of photovoltaic cell parameters is critical for battery life cycle and energy utilization. To accurately identify the single diode model (SDM), dual diode model (DDM), and three diode model (TDM) parameters of solar photovoltaic cells, and an improved honey badger algorithm (IHBA) is proposed in this paper. In the early stages of iteration, the IHBA uses the spiral exploration mechanism to improve the population’s global exploration ability. Furthermore, a density update factor that varies according to the quasi-cosine law is introduced to speed up the algorithm’s convergence speed and prevent the algorithm from falling into the local optimal value. Simultaneously, the pinhole imaging strategy is utilized to disturb the present optimal position to improve the algorithm’s optimization accuracy. The experimental comparison results of 18 benchmark test functions, Wilcoxon rank sum statistical test, and 30 CEC2014 test functions reveal that an IHBA shows remarkable performance in convergence speed, optimization accuracy, and robustness. Finally, the IHBA is used to identify the parameters of three kinds of commercial silicon R.T.C French solar photovoltaic cell models with a 57 mm diameter. In comparison to other algorithms, the IHBA can minimize the root mean square error (RMSE) between the measured current and estimated current at the fastest speed, demonstrating the practicality and superiority of the IHBA in tackling this problem.

Keywords: solar photovoltaic cell; improved honey badger algorithm; diode model; parameter identification

1. Introduction

Renewable energy, such as solar, wind, and hydrogen energy, can be recycled in nature and is an inexhaustible energy source when compared to non-renewable energy. Non-renewable energy has a long history, and there is a lot of research showing that using non-renewable energy can boost economic growth [1,2]. However, there is a one-way positive association between the environmental emission index and the amount of non-renewable energy consumed. Additionally, the ash that remains after burning cannot be recycled, which is detrimental to the sustainable growth of the economy, society, and environment [3]. In order to achieve minimal carbon emissions, sustainable technology must be used [4]. Most importantly, there is a one-way link between renewable energy modes...
consumption and economic growth, while non-renewable energy consumption and economic growth are two-way links [5]. In the long run, renewable energy can meet two-thirds of all energy needs worldwide, effectively mitigating climate problems [6]. Promoting the use of renewable energy and energy transformation is now a global development trend that will have long-term effects on smart cities’ energy use, environmental pollution, and economic growth [7].

Solar photovoltaic technology will certainly aid in the rapid development of renewable energy [8]. Photovoltaic technology is a method of directly converting solar energy into electrical energy by utilizing the photovoltaic effect of the semiconductor interface. This method could be used to manufacture solar photovoltaic cells. The use of photovoltaic cells has the advantages of a natural pollution-free and low-carbon economy for the development of smart cities. However, photovoltaic cells will generate high investment, operation, and maintenance costs [9,10]. Therefore, to reduce various costs, in other words, to prolong the life cycle of photovoltaic cells and improve energy conversion efficiency, it is necessary to identify the parameters of photovoltaic cells. The single diode model (SDM) [11], double diode model (DDM) [12], and three diode model (TDM) [13] are three photovoltaic cell types that are widely studied in academia. The accuracy of the model, as well as the calculation time and complexity, increases as the number of diodes in the model grows. The parameters of the three photovoltaic cell models indicated above will be identified in this paper, with SDM, DDM, and TDM corresponding to five [14], seven [15], and nine [16] parameters to be identified, respectively. The mathematical equation derivation, the numerical or iterative approach, and the metaheuristic algorithm are now used to identify solar cell parameters [17]. When utilized to solve simple circuit models, the mathematical equation deduction approach [18] has a low complexity and operation cost. However, it is rarely used to solve models with complete parameters because the solving error is significant. When solving complex models, numerical [19] or iterative approaches [20,21] have strong robustness and realizability, but they have issues with difficult derivation and low solving accuracy. The metaheuristic algorithm is a type of iterative random operation that is widely utilized in parameter identification of photovoltaic cells because of its strong solving ability, fast calculation speed, and less reasoning process. The more common ones are the particle swarm optimization (PSO) [22], the grasshopper optimization algorithm (GOA) [23], the whale optimization algorithm (WOA) [24], the gray wolf optimization algorithm (GWO) [25], the slime mold algorithm (SMA) [26], etc.

The honey badger algorithm (HBA) is a new metaheuristic algorithm inspired by honey badger foraging behavior, proposed by Fatma A. Hashim et al. in 2021 [27]. The algorithm has a clear structure, is simple to implement, and is very stable. However, the HBA suffers from the same flaws as other metaheuristic algorithms, such as a lack of global exploration capability, sluggish convergence speed, low accuracy, and the ease with which it can fall into a local optimum. Scholars have improved the HBA, and the improved algorithm was used to proton exchange membrane fuel cells [28], feature selection [29], extreme learning machines [30], and other research areas. However, only a few researchers have used the improved HBA to identify solar photovoltaic cell parameters. As a result, this paper proposes an improved honey badger algorithm (IHBA), which is utilized to detect SDM, DDM, and TDM parameters of a 57 mm diameter commercial silicon R.T.C French solar photovoltaic cell at 33 °C and 1000 W/m² light intensity. Parameter identification is the process of calculating the model’s output current based on the model’s measured output voltage data. The smaller the discrepancy between the measured and calculated output current, the more precise algorithm will be in determining the model’s parameters. Compared with other algorithms, the IHBA can find the minimum root mean square error (RMSE) between the measured and calculated output current, indicating that the IHBA can identify the solar photovoltaic cell model parameters more accurately.

The rest of this paper is structured as follows: Section 2 introduces three kinds of solar photovoltaic cell models and the objective function to be solved; Section 3 describes
the improved honey badger algorithm IHBA; Section 4 analyzes the performance of the IHBA through experimental simulation; In Section 5, IHBA is used to identify photovoltaic cell model parameters to verify its practicality and superiority; Section 6 summarizes the paper.

2. Photovoltaic Cell Model and Objective Function

2.1. Single Diode Model (SDM)

The equivalent circuit diagram of the single diode model of the photovoltaic cell is shown in Figure 1, which is mainly composed of a photo-generated current $I_{ph}$, a parallel diode $D_1$, an equivalent parallel resistance $R_{sh}$, and an equivalent series resistance $R_s$.

![Figure 1. Equivalent circuit diagram of SDM.](image)

After being exposed to light, the photo-generated current source will give current to each branch. According to Kirchhoff’s current law, the output current $I$ of SDM can be expressed as:

$$I = I_{ph} - I_{d1} - I_{sh}$$  \hspace{1cm} (1)

In Formula (1), $I_{ph}$ is the current generated by the photo-generated current source, $I_{d1}$ is the current flowing through $D_1$, and $I_{sh}$ is the current flowing through $R_{sh}$. According to the Shockley equation and Kirchhoff’s voltage law, the expressions for $I_{d1}$ and $I_{sh}$ are derived as follows:

$$I_{d1} = I_{sd1}[\exp\left(\frac{q(V + R_sI)}{A_1kT}\right) - 1]$$  \hspace{1cm} (2)

$$I_{sh} = \frac{V + R_sI}{R_{sh}}$$  \hspace{1cm} (3)

where $I_{sd1}$ is the reverse saturation current of $D_1$, $q$ is the electron charge ($1.60217646 \times 10^{-19}$ C), $V$ and $I$ represent the measured output voltage and output current value of SDM respectively, $A_1$ is the ideality factor of $D_1$, and $k$ is the Boltzmann constant ($1.3806503 \times 10^{-23}$ J/K), $T$ is the absolute temperature of the photovoltaic cell.

Through Formulas (1)–(3), $I$ can be further expanded as:

$$I = I_{ph} - I_{sd1}[\exp\left(\frac{q(V + R_sI)}{A_1kT}\right) - 1] - \frac{V + R_sI}{R_{sh}}$$  \hspace{1cm} (4)

From Formula (4), it can be known that for SDM, the five parameters to be identified are: $I_{ph}, I_{sd1}, R_s, R_{sh}$, and $A_1$.

2.2. Double Diode Model (DDM)

DDM adds a parallel diode $D_2$ based on SDM to account for the compound loss in space charge, and the remaining components are the same as SDM. The equivalent circuit diagram of DDM is shown in Figure 2.
After being exposed to light, the photo-generated current source will give current to each branch. According to Kirchhoff’s current law, the output current $I$ of DDM can be expressed as:

$$I = I_{ph} - I_{d1} - I_{d2} - I_{sh}$$

(5)

In Formula (5), $I_{ph}$ is the current generated by the photo-generated current source, $I_{d1}$ and $I_{d2}$ are the currents that flow through $D_1$ and $D_2$, and $I_{sh}$ is the current flowing through $R_{sh}$. Through Formulas (2), (3) and (5), $I$ can be further expanded as:

$$I = I_{ph} - I_{sd1} [\exp(\frac{q(V + R_s I)}{A_1 kT}) - 1] - I_{sd2} [\exp(\frac{q(V + R_s I)}{A_2 kT}) - 1] - \frac{V + R_s I}{R_{sh}}$$

(6)

where $I_{sd1}$ and $I_{sd2}$ are the reverse saturation currents of $D_1$ and $D_2$, respectively, and $A_1$ and $A_2$ are the ideality factors of $D_1$ and $D_2$, respectively. From Formula (6), it can be known that for DDM, the seven parameters to be identified are: $I_{ph}$, $I_{sd1}$, $I_{sd2}$, $R_s$, $R_{sh}$, $A_1$, and $A_2$.

2.3. Three Diode Model (TDM)

TDM adds a parallel diode $D_3$ based on DDM to compensate for the loss of defect area, and the remaining components are the same as DDM. The equivalent circuit diagram of TDM is shown in Figure 3.

After being exposed to light, the photo-generated current source will give current to each branch. According to Kirchhoff’s current law, the output current $I$ of TDM can be expressed as:

$$I = I_{ph} - I_{d1} - I_{d2} - I_{d3} - I_{sh}$$

(7)

In Formula (7), $I_{ph}$ is the current generated by the photo-generated current source, $I_{d1}$, $I_{d2}$, and $I_{d3}$ are currents that flow through $D_1$, $D_2$, and $D_3$, and $I_{sh}$ is the current flowing through $R_{sh}$. Through Formulas (2), (3) and (7), $I$ can be further expanded as:
where \( I_{sd1}, I_{sd2}, \) and \( I_{sd3} \) are the reverse saturation currents of \( D_1, D_2, \) and \( D_3 \) respectively, \( A_1, A_2, \) and \( A_3 \) are the ideality factors of \( D_1, D_2, \) and \( D_3 \) respectively. From Formula (8), it can be known that for TDM, the nine parameters to be identified are: \( I_{ph}, I_{sd1}, I_{sd2}, I_{sd3}, R_s, R_{sh}, A_1, A_2, \) and \( A_3. \)

### 2.4. Objective Function

The goal of photovoltaic cell parameter identification is to determine the optimal parameters by minimizing the difference between the measured and calculated output currents. The errors of measured output current and calculated output current of the three photovoltaic cell models can be derived by subtracting the right side from the left side of Equations (4), (6) and (8), respectively. As a result, the SDM, DDM, and TDM error functions \( f \) can be defined as follows:

\[
\begin{align*}
\chi_{SDM}(V, I, \chi) &= I - I_{ph} - I_{sd1}\left[\exp\left(\frac{q(V + R_sI)}{A_kT}\right) - 1\right] - I_{sd2}\left[\exp\left(\frac{q(V + R_sI)}{A_kT}\right) - 1\right] - I_{sd3}\left[\exp\left(\frac{q(V + R_sI)}{A_kT}\right) - 1\right] - \frac{V + R_sI}{R_{sh}} \\
\chi_{DDM}(V, I, \chi) &= I - I_{ph} - I_{sd1}\left[\exp\left(\frac{q(V + R_sI)}{A_kT}\right) - 1\right] - I_{sd2}\left[\exp\left(\frac{q(V + R_sI)}{A_kT}\right) - 1\right] - I_{sd3}\left[\exp\left(\frac{q(V + R_sI)}{A_kT}\right) - 1\right] - \frac{V + R_sI}{R_{sh}} \\
\chi_{TDM}(V, I, \chi) &= I - I_{ph} - I_{sd1}\left[\exp\left(\frac{q(V + R_sI)}{A_kT}\right) - 1\right] - I_{sd2}\left[\exp\left(\frac{q(V + R_sI)}{A_kT}\right) - 1\right] - I_{sd3}\left[\exp\left(\frac{q(V + R_sI)}{A_kT}\right) - 1\right] - \frac{V + R_sI}{R_{sh}}
\end{align*}
\]

where \( \chi \) is the solution vector to be solved, and its elements are the unknown parameters to be identified.

The objective function of this paper is the root mean square error (RMSE) between the measured output current and the calculated output current, and its mathematical expression is as follows:

\[
RMSE(\chi) = \sqrt{\frac{1}{N} \sum_{i=1}^{N} \left(f_i(V, I, \chi)\right)^2}
\]

where \( N \) is the number of experimental data sets. The smaller the RMSE, the closer the computed output current is to the measured output current and the higher the model parameter identification accuracy.

### 3. Improved Honey Badger Algorithm

#### 3.1. Honey Badger Algorithm

The honey badger algorithm (HBA) imitates the foraging behavior of the honey badger, and its bionic principle is as follows: In the global exploration stage, the honey badger uses the olfactory skills of mice to locate the location of the hive and selects the appropriate location to capture honey after reaching the vicinity of the hive. Honeyguide birds are expected to acquire honeycombs and are adept at locating them. In the local
development stage, honey badgers are guided by honeyguide birds to directly locate honeycombs and eventually obtain honey. The scent intensity $I_i$ of the prey felt by the honey badger at its current location will positively affect its movement speed. The formula of scent intensity is as follows:

$$I_i = r_i \times \frac{S}{4\pi d_i^2}$$

$$S = (x_i - x_{i+1})^2$$
$$d_i = x_{prey} - x_i$$

where $S$ is concentration intensity, determined by the honey badger’s present location $x_i$ and the next position $x_{i+1}$, $d_i$ is the distance between the prey position $x_{prey}$ and the current honey badger position $x_i$, and $r_i$ is a random value between $[0, 1]$.

The density factor $\alpha$ ensures a smooth transition of HBA from global exploration to local development, and its mathematical model is as follows:

$$\alpha = C \times \exp\left(\frac{-t}{t_{max}}\right)$$

where $t$ is the current iteration number, $t_{max}$ is the maximum iteration number, and $C$ is a constant greater than or equal to 1 (default is 2).

In the global exploration stage, the honey badger population follows the heart-shaped line’s movement trajectory, which has the following mathematical expression:

$$x_{new} = x_{prey} + F \times \beta \times I \times x_{prey} + F \times r_2 \times \alpha \times d_i \times \gamma$$

where $x_{new}$ is the honey badger’s new location, $x_{prey}$ is the prey’s position and the global best position found so far, $\beta \geq 1$ (default is 6) is the honey badger’s capacity to obtain honey, $I$ and $d_i$ are determined by Equation (13), $\alpha$ is determined by Equation (14), $r_2$, $r_3$, and $r_4$ are three separate random values between $[0, 1]$. Interference flag $F$ has a chance of changing the honey badger’s search direction, giving the individual extra chances to thoroughly examine the whole search field. Equation (16) shows the mathematical model of $F$, where $r_5$ is the random number between $[0, 1]$.

$$F = \begin{cases} 
1 & \text{if } r_5 \leq 0.5 \\
-1 & \text{else}
\end{cases}$$

The population migration trajectory may be modeled using the following formula during the local development stage:

$$x_{new} = x_{prey} + F \times r_6 \times \alpha \times d_i$$

where $x_{new}$ and $x_{prey}$ are the new location and prey location of honey badger respectively, $d_i$, $\alpha$, and $F$ are calculated by the Formulas (13), (14) and (16), and $r_6$ is a random value between $[0, 1]$.

### 3.2. Improved Honey Badger Algorithm
#### 3.2.1. Spiral Exploration Mechanism

The population explores the search space utilizing the motion trajectory of the heart-shaped line during the global exploration stage of the HBA. However, in the early stages of iteration, the traversal range of the heart-shaped line is limited, and the global exploration ability is insufficient. This paper provides the spiral search mechanism to prevent the
condition described above. The spiral’s trajectory will be used by the population to update its position. The following is the revised position update formula:

$$x_{new} = x_{prey} + F \times \beta \times I \times x_{prey} + F \times r_{j} \times \alpha \times d \times \gamma_{1}$$

$$\gamma_{1} = \cos(2\pi a) \times b \times \exp(a \times n)$$

(18)

where $a$ is the random number between $[-1, 1]$, which reflects the optimization trajectory of the helix. The search range of the population in space is represented by $b$. The spiral shape’s size is denoted by $n$. In this paper, $b = 1$ and $n = 1.2$ are chosen. $r_{j}$ is a random value between $[0, 1]$. After the reconstruction of the position updating formula of the population, honey badger can identify prey location in the search space with a spiral trajectory, thus improving the overall searchability of the population.

3.2.2. Density Factor of Quasi-Cosine Law Variation

The density factor is an important element in the balancing algorithm’s exploration and development stages. It can be seen from Equation (14) that $\alpha$ decreases continuously in the shape of a concave function with the increase in the number of iterations, which conforms to the general law of the algorithm in global optimization. However, the change of $\alpha$ is gentle, which reduces the convergence speed of the algorithm. In addition, in the middle of the algorithm iteration, the population has the possibility of falling into the local optimal value, and a continuous decrease in the density factor may cause the population to fail to escape the local optimal value. Therefore, the density factor of the variation of the quasi-cosine law is proposed in this paper, and its mathematical model is as follows:

$$\alpha = \left[ 1.7 \times \omega \times \cos(2\pi t + 1.3 \times \cos(2\pi \times \omega)) \right]$$

$$\omega = \cos\left(\frac{t}{t_{max} \times 2} \times \pi\right) + 0.2$$

(19)

where $t$ is the current iteration number, and $t_{max}$ is the maximum iteration number. Figure 4 depicts the before and after improvement comparison curves of $\alpha$.

![Figure 4. Comparison curves of $\alpha$ before and after improvement.](image)

Figure 4 demonstrates that the improved $\alpha$ rapidly declines from the maximum value in the early iteration, guaranteeing that the population is searched over a wide range of space while simultaneously speeding up the algorithm’s convergence speed. In the middle of the iteration, when $\alpha$ drops to a small value, it is re-given an incremental step size,
allowing the population to find the global optimal solution again over a large range, which is beneficial for the algorithm to jump out of the local optimal value. In the later phases of the iteration, the value of $\alpha$ will once more decrease dramatically, enabling the population to quickly become close to the optimal value globally. The population will be able to conduct in-depth local searches when the value of $\alpha$ is small and fluctuates very little near the end of the iteration.

3.2.3. Pinhole Imaging Strategy

According to Equations (17) and (18), the process of the honey badger searching for prey is mainly guided by the current optimal individual. The algorithm will converge prematurely if the present optimum person is the local optimal individual. As a result, this paper presents a pinhole imaging strategy and applies it dimension by dimension to the current optimal individual, which not only eliminates reciprocal interference between individual dimensions but also improves the variety of the current optimal individual. The following is its one-dimensional mathematical model:

$$\frac{(a+b)/2 - x}{x' - (a+b)/2} = \frac{l}{l'}$$

(20)

As illustrated in Figure 5, in Formula (20), $a$ and $b$ are the projected image’s horizontal upper and lower bounds, respectively. $x$ and $x'$ are projections of projected point $q$ and the projected point $q^*$ on the boundary, respectively. $l$ and $l'$ are the vertical distances from $q$ and $q^*$ to the horizontal boundary, respectively.

![Figure 5. Pinhole imaging.](image)

Extending Formula (20) to multi-dimensional space and applying it to the current optimal individual dimension by dimension, the mathematical model becomes:

$$\frac{(a_{\text{prey}}^j + b_{\text{prey}}^j)/2 - x_{\text{prey}}^j}{x_{\text{prey}}^j - (a_{\text{prey}}^j + b_{\text{prey}}^j)/2} = \frac{l}{l'}$$

(21)

Set the scaling factor $l/l' = r$, the solution of pinhole imaging based on the current optimal individual is:

$$x_{\text{prey}}^j = \frac{(a_{\text{prey}}^j + b_{\text{prey}}^j) + (a_{\text{prey}}^j + b_{\text{prey}}^j) - x_{\text{prey}}^j}{2 \times r} - \frac{x_{\text{prey}}^j}{r}$$

(22)

where $a_{\text{prey}}^j$ and $b_{\text{prey}}^j$ denote the lower and higher borders of the current optimum individual’s $j$th dimension, respectively. $x_{\text{prey}}^j$ denote the value corresponding to the current optimal individual’s $j$th dimension. $r$ is assigned to a random value between $[-3, 3]$ in this paper. Taking a one-dimensional plane as an example, in Figure 5, the solution of the pinhole imaging can exist on both sides of the pinhole screen, which increases the diversity of the current optimal individuals.

Although using the pinhole imaging strategy to perturb the current optimal individual is advantageous for the algorithm to use a better current optimal individual in the next iteration to lead the population closer to the global optimal individual and improve the algorithm’s optimization accuracy, there is no guarantee that the current optimal
individual after disturbance is better than the individual before disturbance. As a result, a greedy approach is utilized to assess the fitness value of the current optimal individual before and after the disturbance, and the individual that is more valuable for a population position update is retained. Its mathematical model is as follows:

\[
    x_{\text{prey}} = \begin{cases} 
    x_{\text{prey}}, & f(x_{\text{prey}}) < f(x_{\text{prey}*}) \\
    x_{\text{prey}*}, & f(x_{\text{prey}}) \geq f(x_{\text{prey}*}) 
    \end{cases}
\]  

(23)

3.2.4. IHBA Implementation Steps

Step 1: Initialization parameters include the population size \( N \), the maximum number of iterations \( t_{\text{max}} \), variable \( \beta \), the space dimension, and the upper and lower search area bounds.

Step 2: Calculate the fitness value of each individual and save the initial optimal individual.

Step 3: Formula (13) is used to determine \( I \), while Formula (19) is used to update \( \alpha \).

Step 4: If the random number between (0, 1) is less than 0.5, update the population position according to Formula (18), otherwise, use Formula (17) to update the population position.

Step 5: Recalculate each individual’s fitness value and retain the current optimal individual \( x_{\text{prey}} \) and current optimal fitness value.

Step 6: The current optimal individual \( x_{\text{prey}} \) is perturbed dimension by dimension using the pinhole imaging strategy, and the greedy strategy is used to retain the current optimal individual that is more valuable for the population position update.

Step 7: Determine whether the maximum number of iterations has been reached and if so, return the value of the global optimal individual and the global optimal fitness value; otherwise, the algorithm returns to step 3 until the end of the iteration.

The flow chart of the IHBA is shown in Figure 6.
3.2.5. Time Complexity Analysis of IHBA

The time complexity characterizes the optimization efficiency of the algorithm. In the HBA, it is assumed that the maximum number of iterations of the population is $t_{\text{max}}$, the population size is $N$, and the number of decision variables is $D$. The time to initialize the relevant parameters of the algorithm is $O(u)$, and the time to solve the population fitness value is $O(f)$, then the time complexity of the HBA is:

$$O(u) + O(N \cdot (2f + t_{\text{max}} \cdot D)) = O(N \cdot t_{\text{max}} \cdot D) \quad (24)$$

In the IHBA, the population initialization is the same as the HBA, the time to update the density factor is $O(v)$, the execution time of the pinhole imaging strategy is $O(m)$, and the time for the population position update is $O(t_{\text{max}}ND)$, and the time required to update the current optimal individual by using pinhole imaging strategy is $O(t_{\text{max}}Nm)$, then the total time complexity of IHBA execution is:

$$O(u + t_{\text{max}} \cdot v + N \cdot (2f + t_{\text{max}} \cdot (D + m))) = O(N \cdot t_{\text{max}} \cdot D) \quad (25)$$

Comparing the time complexity of the HBA and the IHBA, it can be seen that the improved algorithm does not increase the time complexity of calculation.

4. Experimental Simulation and Result Analysis

4.1. Simulation Environment and Test Function

The simulation experiments were all run on an Intel Core i5-6500 CPU with a 3.20 GHz main frequency, Windows 10 (64-bit) as the operating system, and MATLAB R2016a (https://matlab.mathworks.com, accessed on 2 March 2022) as the programming software. The 18 benchmark functions necessary for the experiment are listed in Table 1, with F1~F6 being single-peak functions, F7~F10 being complicated multimodal functions, and F11~F18 being multimodal functions with fixed dimensions.

| Function | Name                                | Dim       | Domain               | Optimal Value |
|----------|-------------------------------------|-----------|----------------------|---------------|
| F1       | Sphere                              | 30/100/500| [−100, 100]          | 0             |
| F2       | Schwefel’ problem 2.22              | 30/100/500| [−10, 10]            | 0             |
| F3       | Schwefel’ problem 1.2               | 30/100/500| [−100, 100]          | 0             |
| F4       | Schwefel’ problem 2.21              | 30/100/500| [−100, 100]          | 0             |
| F5       | Step Function                       | 30/100/500| [−100, 100]          | 0             |
| F6       | Quartic Function                    | 30/100/500| [−1.28, 1.28]        | 0             |
| F7       | Generalized Rastrigin’s Function    | 30/100/500| [−5.12, 5.12]        | 0             |
| F8       | Ackley’s Function                   | 30/100/500| [−32, 32]            | 0             |
| F9       | Generalized Criedewank Function     | 30/100/500| [−600, 600]          | 0             |
| F10      | Generalized Penalized Function 1    | 30/100/500| [−50, 50]            | 0             |
| F11      | Shekel’s Foxholes Function          | 2         | [−65, 65]            | 1             |
| F12      | Kowelik’s Function                  | 4         | [−5, 5]              | 0.0003        |
| F13      | Six-Hump Camel-Back Function        | 2         | [−5, 5]              | −1.03         |
| F14      | Hatman’s Function1                  | 3         | [0, 1]               | −3.86         |
| F15      | Hatman’s Function2                  | 6         | [0, 1]               | −3.32         |
| F16      | Shekeль’s Family 1                  | 4         | [1, 10]              | −10           |
| F17      | Shekeль’s Family 2                  | 4         | [1, 10]              | −10           |
| F18      | Shekeль’s Family 3                  | 4         | [1, 10]              | −10           |

4.2. Optimization Comparison of Different Improvement Strategies

To verify the effectiveness of each improvement strategy, the honey badger algorithm (HBA), the honey badger algorithm that introduces spiral exploration mechanism.
(HBA1), the honey badger algorithm that introduces quasi-cosine law varied density factor (HBA2), the honey badger algorithm that introduces pinhole imaging strategy (HBA3) and the improved honey badger algorithm that fusions the three methods (IHBA) are compared using the benchmark functions shown in Table 1. To ensure the fairness of experimental findings and eliminate errors, each algorithm was independently run 30 times with a total number of iterations of 500, a population size of 30, and an algorithm dimension of 30. The independent operation outcomes’ best value, worst value, mean value, and standard deviation are chosen. The first three may be used to assess a search algorithm’s accuracy, while the latter can be used to evaluate a search algorithm’s stability. The experimental findings are shown in Table 2.

**Table 2.** Numerical optimization comparison of different improvement strategies.

| Function | Algorithm | Best | Worst | Mean | Std |
|----------|-----------|------|-------|------|-----|
| HBA      | 8.24 × 10^{-9} | 1.32 × 10^{-78} | 7.23 × 10^{-80} | 2.60 × 10^{-79} |
| HBA1     | 3.42 × 10^{-116} | 7.73 × 10^{-103} | 5.21 × 10^{-104} | 1.77 × 10^{-103} |
| HBA2     | 8.05 × 10^{-158} | 2.81 × 10^{-141} | 1.00 × 10^{-142} | 5.13 × 10^{-142} |
| HBA3     | 0.00 | 0.00 | 0.00 | 0.00 |
| IHBA     | 0.00 | 0.00 | 0.00 | 0.00 |
| F1       | HBA      | 8.92 × 10^{-46} | 8.76 × 10^{-43} | 1.66 × 10^{-43} | 1.66 × 10^{-43} |
|          | HBA1     | 7.12 × 10^{-9} | 4.02 × 10^{-53} | 5.24 × 10^{-54} | 5.24 × 10^{-54} |
|          | HBA2     | 2.22 × 10^{-79} | 3.56 × 10^{-73} | 2.94 × 10^{-74} | 2.94 × 10^{-74} |
|          | HBA3     | 3.02 × 10^{-295} | 9.06 × 10^{-274} | 3.05 × 10^{-275} | 3.05 × 10^{-275} |
|          | IHBA     | 0.00 | 1.61 × 10^{-319} | 5.37 × 10^{-321} | 5.37 × 10^{-321} |
| F2       | HBA      | 1.02 × 10^{-62} | 1.01 × 10^{-53} | 3.41 × 10^{-55} | 1.84 × 10^{-54} |
|          | HBA1     | 9.67 × 10^{-99} | 9.32 × 10^{-84} | 3.33 × 10^{-85} | 1.70 × 10^{-84} |
|          | HBA2     | 1.14 × 10^{-138} | 1.87 × 10^{-126} | 7.51 × 10^{-128} | 3.46 × 10^{-127} |
|          | HBA3     | 0.00 | 0.00 | 0.00 | 0.00 |
|          | IHBA     | 0.00 | 0.00 | 0.00 | 0.00 |
| F3       | HBA      | 3.91 × 10^{-33} | 2.73 × 10^{-31} | 4.39 × 10^{-32} | 5.83 × 10^{-32} |
|          | HBA1     | 2.04 × 10^{-55} | 1.32 × 10^{-45} | 9.14 × 10^{-47} | 3.24 × 10^{-46} |
|          | HBA2     | 8.04 × 10^{-71} | 3.71 × 10^{-66} | 2.73 × 10^{-67} | 7.13 × 10^{-67} |
|          | HBA3     | 2.16 × 10^{-269} | 8.49 × 10^{-254} | 3.41 × 10^{-255} | 0.00 |
|          | IHBA     | 0.00 | 2.07 × 10^{-321} | 6.90 × 10^{-323} | 0.00 |
| F4       | HBA      | 0.00 | 3.89 × 10^{-7} | 1.44 × 10^{-8} | 6.64 × 10^{-8} |
|          | HBA1     | 0.00 | 2.24 × 10^{-8} | 2.02 × 10^{-8} | 7.14 × 10^{-8} |
|          | HBA2     | 0.00 | 2.24 × 10^{-8} | 3.12 × 10^{-9} | 6.04 × 10^{-9} |
|          | HBA3     | 0.00 | 1.77 × 10^{-8} | 2.03 × 10^{-9} | 3.83 × 10^{-9} |
|          | IHBA     | 0.00 | 3.65 × 10^{-8} | 2.53 × 10^{-9} | 7.70 × 10^{-9} |
| F5       | HBA      | 8.56 × 10^{-5} | 2.28 × 10^{-3} | 8.38 × 10^{-4} | 5.95 × 10^{-4} |
|          | HBA1     | 4.66 × 10^{-5} | 1.93 × 10^{-3} | 5.21 × 10^{-4} | 5.95 × 10^{-4} |
|          | HBA2     | 3.47 × 10^{-5} | 1.99 × 10^{-3} | 5.91 × 10^{-4} | 5.57 × 10^{-4} |
|          | HBA3     | 3.88 × 10^{-6} | 2.55 × 10^{-4} | 6.71 × 10^{-5} | 6.12 × 10^{-5} |
|          | IHBA     | 4.55 × 10^{-6} | 3.30 × 10^{-4} | 8.60 × 10^{-5} | 7.35 × 10^{-5} |
| F6       | HBA      | 0.00 | 0.00 | 0.00 | 0.00 |
|          | HBA1     | 0.00 | 0.00 | 0.00 | 0.00 |
|          | HBA2     | 0.00 | 0.00 | 0.00 | 0.00 |
|          | HBA3     | 0.00 | 0.00 | 0.00 | 0.00 |
|          | IHBA     | 0.00 | 0.00 | 0.00 | 0.00 |
| F7       | HBA      | 8.88 × 10^{-16} | 2.00 × 10 | 1.46 × 10 | 8.98 |
|          | HBA1     | 8.88 × 10^{-16} | 1.55 × 10 | 5.15 × 10^{-3} | 2.82 |
|          | HBA2     | 8.88 × 10^{-16} | 2.77 × 10^{-10} | 9.24 × 10^{-12} | 5.06 × 10^{-11} |
|   | HBA3  | IHBA  | HBA1  | IHBA  | HBA2  | IHBA  |
|---|-------|-------|-------|-------|-------|-------|
| F9 | 8.88 × 10^{-16} | 8.88 × 10^{-16} | 8.88 × 10^{-16} | 0.00  | 0.00  | 0.00  |
|   | 8.88 × 10^{-16} | 8.88 × 10^{-16} | 8.88 × 10^{-16} | 0.00  | 0.00  | 0.00  |
|   | 8.88 × 10^{-16} | 8.88 × 10^{-16} | 8.88 × 10^{-16} | 0.00  | 0.00  | 0.00  |

| HBA | 1.28 × 10^{-10} | 1.04 × 10^{-1} | 3.46 × 10^{-3} | 1.89 × 10^{-2} |
| IHBA | 4.10 × 10^{-11} | 1.04 × 10^{-1} | 3.46 × 10^{-3} | 1.89 × 10^{-2} |
| HBA2 | 2.13 × 10^{-10} | 1.04 × 10^{-1} | 3.47 × 10^{-3} | 1.89 × 10^{-2} |
| HBA3 | 1.82 × 10^{-10} | 6.57 × 10^{-3} | 2.19 × 10^{-4} | 1.20 × 10^{-3} |
| IHBA | 7.00 × 10^{-11} | 6.01 × 10^{-8} | 7.68 × 10^{-9} | 1.26 × 10^{-8} |

| F10 | 9.98 × 10^{-1} | 1.08 × 10 | 2.18 | 2.10 |
| IHBA | 9.98 × 10^{-1} | 3.97 | 1.46 | 9.64 × 10^{-1} |
| HBA2 | 9.98 × 10^{-1} | 1.08 × 10 | 2.50 | 3.09 |
| HBA3 | 9.98 × 10^{-1} | 1.08 × 10 | 1.88 | 2.49 |
| IHBA | 9.98 × 10^{-1} | 5.93 | 1.82 | 1.42 |

| F11 | 0.00 | 2.26 × 10^{-2} | 2.38 × 10^{-3} | 6.62 × 10^{-3} |
| IHBA | 0.00 | 2.26 × 10^{-2} | 1.78 × 10^{-3} | 5.67 × 10^{-3} |
| HBA2 | 0.00 | 2.26 × 10^{-2} | 2.21 × 10^{-3} | 6.41 × 10^{-3} |
| HBA3 | 0.00 | 2.26 × 10^{-2} | 1.59 × 10^{-3} | 5.42 × 10^{-3} |
| IHBA | 0.00 | 2.04 × 10^{-2} | 8.75 × 10^{-4} | 3.70 × 10^{-3} |

| F12 | -1.03 | 0.00 | -3.44 × 10^{-12} | 4.95 × 10^{-12} |
| IHBA | -1.03 | 0.00 | -3.44 × 10^{-12} | 4.95 × 10^{-12} |
| HBA2 | -1.03 | 0.00 | -3.44 × 10^{-12} | 4.95 × 10^{-12} |
| HBA3 | -1.03 | 0.00 | -3.44 × 10^{-12} | 4.95 × 10^{-12} |
| IHBA | -1.03 | 0.00 | -3.44 × 10^{-12} | 4.95 × 10^{-12} |

| F13 | -3.86 | -3.65 | -3.86 | -3.07 × 10^{-4} |
| IHBA | 3.86 | -3.75 | 3.86 | -3.27 × 10^{-4} |
| HBA2 | -3.86 | -3.75 | -3.86 | -3.27 × 10^{-4} |
| HBA3 | -3.86 | -3.75 | -3.86 | -3.27 × 10^{-4} |
| IHBA | -3.86 | -3.75 | -3.86 | -3.27 × 10^{-4} |

| F14 | -3.20 | -3.20 | -3.20 | 1.56 × 10^{-2} |
| IHBA | -3.20 | -3.20 | -3.20 | 1.56 × 10^{-2} |
| HBA2 | -3.20 | -3.20 | -3.20 | 1.56 × 10^{-2} |
| HBA3 | -3.32 | -3.30 | -3.32 | 1.58 × 10^{-2} |
| IHBA | -3.32 | -3.32 | -3.32 | 1.58 × 10^{-2} |

| F15 | -1.02 × 10 | -9.36 | -9.74 | 4.87 |
| IHBA | -1.02 × 10 | -9.36 | -9.74 | 4.87 |
| HBA2 | -1.02 × 10 | -1.02 × 10 | -1.02 × 10 | 4.71 |
| HBA3 | -1.02 × 10 | -1.02 × 10 | -1.02 × 10 | 4.71 |
| IHBA | -1.02 × 10 | -1.02 × 10 | -1.02 × 10 | 4.71 |

| F16 | -9.24 | -8.21 | -8.68 | 4.99 |
| IHBA | -9.79 | -9.24 | -9.38 | 4.16 |
| HBA2 | -9.79 | -9.24 | -9.38 | 4.16 |
| HBA3 | -9.79 | -9.24 | -9.38 | 4.16 |
| IHBA | -1.04 × 10 | -1.04 × 10 | -1.04 × 10 | 3.89 |

| F17 | -9.13 | -9.04 | -9.10 | 4.46 × 10^{-2} |
| IHBA | -9.93 | -9.64 | -9.93 | 5.05 × 10^{-2} |
| HBA2 | -9.93 | -9.64 | -9.93 | 5.05 × 10^{-2} |
| HBA3 | -1.05 × 10 | -1.00 × 10 | -1.03 × 10 | 5.05 × 10^{-2} |
Table 2 shows that, in terms of best value, the IHBA finds the optimal value for functions F1–F5, F7, F9, F11–F12, and F14–F15. For functions F6 and F8, there are many local extreme values in their function shapes, causing each algorithm to fail to find the optimal value. The IHBA, on the other hand, came out on top in terms of optimization accuracy and stability among these algorithms. The optimization accuracy of each modified algorithm is enhanced when compared to the original algorithm for functions F13 and F16–F18. In terms of worst value, mean value, and standard deviation, the IHBA has the highest searching accuracy and the best searching stability among comparison algorithms.

To be more explicit, the HBA1 does not considerably increase the solution accuracy of other functions except F10, but it does minimize the instability of algorithm optimization to some level and aids in following improvement strategies. The accuracy and stability of the HBA2 have increased greatly, thanks to the fact that the density factor of quasi-cosine law variation might cause the algorithm to escape from a local optimum in the middle stage. The HBA3 shows the greatest improvement impact, indicating that employing the pinhole imaging strategy can obtain the population nearest the optimal value. The IHBA combines the best of the three improvement strategies to achieve excellent optimization accuracy and stability. In order to further verify the optimization effect of different improvement strategies, the convergence curve will be used for comparative analysis.

4.3. Comparison of Average Convergence Curves of Different Improvement Strategies

The convergence curve can display the algorithm’s optimization speed and accuracy in a more accessible way. The HBA, HBA1, HBA2, HBA3, and IHBA are run using the benchmark functions shown in Table 1. The total number of iterations is 500, and each algorithm runs 30 times separately, the algorithm dimension is 30, and the population number is 30. Figure 7 depicts different improvement strategies’ average convergence curve.
Figure 7. Different improvement strategies average convergence curves: (a) F1; (b) F2; (c) F3; (d) F4; (e) F5; (f) F6; (g) F7; (h) F8; (i) F9; (j) F10; (k) F11; (l) F12; (m) F13; (n) F14; (o) F15; (p) F16; (q) F17; (r) F18.
It can be seen from Figure 7 that for functions F1, F3~F4, and F7~F10, the convergence curve of the IHBA is always below the comparison algorithm, which shows that it has superiority in both convergence speed and optimization accuracy. When compared to the HBA3, the IHBA has no evident advantages in the early stages of iteration for functions F2 and F5, but it can continually optimize to the optimal value and swiftly converge in the latter stages to discover the theoretical optimal value. For function F6, the IHBA can jump out of local extrema multiple times to improve the convergence accuracy. For the function F11, the convergence accuracy of the other improved algorithms was greatly improved, except that the convergence accuracy of the HBA1 is not significantly improved. For functions F12, and F16~F18, the optimization accuracy of each algorithm is improved compared with the original algorithm, although the final optimization accuracy of each improved algorithm is not much different, the convergence speed of the IHBA is the fastest among the comparison algorithms. Although the convergence accuracy of each approach is not significantly different for functions F13~F14, the IHBA has the fastest convergence speed. For function F15, the IHBA has the highest optimization accuracy. From the overall improvement effect, each improved algorithm has improved the solution speed and accuracy when compared to the original algorithm, with the IHBA performing better.

4.4. Comparison with Other New and Improved Algorithms

On the benchmark test function provided in Table 1, an optimization comparison was conducted between the IHBA and the recently proposed joint search mechanism of the whale optimization algorithm (JSWOA) [31], the modified equilibrium optimizer (M-EO) [32], the chaotic mechanism whale optimization algorithm based on quasi-opposition (OBCWOA) [33], and the reinforced variant whale optimization algorithm (RDWOA) [34] to demonstrate the competitiveness of the IHBA. The spatial dimensions of the other functions are set at 30/100/500 correspondingly, except the multi-peak function F11~F18, which has fixed dimensions. Each algorithm was run 30 times in total, with a total of 500 iterations and a population size of 30. Tables 3 and 4 contain the mean and standard deviation of the experimental data. The value "-" in the table indicates that the original literature does not have a matching value.

| Function | Algorithm   | 30dim | 100dim | 500dim |
|----------|-------------|-------|--------|--------|
|          | Mean | Std | Mean | Std | Mean | Std |
| F1       | JSWOA | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
|          | m-EO  | 0.00 | 0.00 | 1.53 × 10⁻⁰⁴ | 0.00 | 0.00 |
| OBCWOA  | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| RDWOA   | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| IHBA     | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| F2       | JSWOA | 3.93 × 10⁻⁶⁷ | 0.00 | 3.09 × 10⁻⁶⁶ | 1.579 × 10⁻⁶⁰ | 1.36 × 10⁻⁶⁰ | 5.90 × 10⁻⁶⁰ |
|          | m-EO  | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| OBCWOA  | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| RDWOA   | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| IHBA     | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| F3       | JSWOA | 2.71 × 10⁻³⁰⁶ | 0.00 | 8.50 × 10⁻²⁹⁷ | 0.00 | 4.63 × 10⁻²⁹³ | 0.00 |
|          | m-EO  | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| OBCWOA  | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| RDWOA   | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| IHBA     | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| F4       | JSWOA | 2.31 × 10⁻¹⁵⁹ | 1.17 × 10⁻¹⁵⁸ | 2.83 × 10⁻¹⁵⁷ | 9.04 × 10⁻¹⁵⁷ | 1.48 × 10⁻¹⁵⁴ | 7.46 × 10⁻¹⁵⁴ |
|          | m-EO  | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| OBCWOA  | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| Function | Dim | Algorithm | Mean   | Std    | Function | Dim | Algorithm | Mean   | Std    |
|----------|-----|-----------|--------|--------|----------|-----|-----------|--------|--------|
| F5       |     | OBCWOA    | 3.87 x 10^{-1} | 2.10 x 10^{-1} | F12    | 4   | OBCWOA    | 3.82 x 10^{-4} | 2.05 x 10^{-4} |
|          |     | RDWOA     | -      | -      |          |     | RDWOA     | 3.08 x 10^{-4} | 1.24 x 10^{-7}  |
|          |     | IHBA      | 9.95 x 10^{-4} | 3.36 x 10^{-8} | F16    |     |          | -      | -      |
|          |     |           |        |        |          |     |           |        |        |
| F6       |     | JSWOA     | 7.63 x 10^{-5} | 7.12 x 10^{-5} | F7     |     | OBCWOA    | 8.88 x 10^{-16} | 8.88 x 10^{-16} |
|          |     | m-EO      | 2.47 x 10^{-4} | 2.23 x 10^{-4} |       |     | RDWOA     | 8.88 x 10^{-16} | 8.88 x 10^{-16} |
|          |     | RDWOA     | 1.30 x 10^{-5} | 1.23 x 10^{-5} | IHBA   |     | 8.88 x 10^{-16} | 8.88 x 10^{-16} | 8.88 x 10^{-16} |
|          |     | IHBA      | 6.24 x 10^{-5} | 4.48 x 10^{-5} |       |     |          | -      | -      |
|          |     |           |        |        |          |     |           |        |        |
| F7       |     | JSWOA     | 8.88 x 10^{-16} | 8.88 x 10^{-16} | F8     |     | JSWOA     | 4.40 x 10^{-2} | 1.52 x 10^{-2}  |
|          |     | m-EO      | 8.88 x 10^{-16} | 8.88 x 10^{-16} |       |     | m-EO      | 6.25 x 10^{-4} | 3.92 x 10^{-6}  |
|          |     | OBCWOA    | 8.88 x 10^{-16} | 8.88 x 10^{-16} | IHBA   |     | 8.88 x 10^{-16} | 8.88 x 10^{-16} | 8.88 x 10^{-16} |
|          |     | RDWOA     | 8.88 x 10^{-16} | 8.88 x 10^{-16} |       |     |          | -      | -      |
|          |     | IHBA      | 8.88 x 10^{-16} | 8.88 x 10^{-16} |       |     |          | -      | -      |
|          |     |           |        |        |          |     |           |        |        |
| F9       |     | JSWOA     | 0.00   | 0.00   | F10     |     | JSWOA     | 2.56 x 10^{-10} | 1.23 x 10^{-9}  |
|          |     | m-EO      | 0.00   | 0.00   |       |     | m-EO      | -      | -      |
|          |     | OBCWOA    | 0.00   | 0.00   | IHBA   |     | 0.00      | 0.00   | 0.00   |
|          |     | RDWOA     | 0.00   | 0.00   |       |     |          | -      | -      |
|          |     | IHBA      | 0.00   | 0.00   |       |     |          | -      | -      |
|          |     |           |        |        |          |     |           |        |        |
|          |     |           |        |        |          |     |           |        |        |
|          |     |           |        |        |          |     |           |        |        |

**Table 4.** Comparison of solving results of each algorithm for fixed-dimensional multimodal functions.
Table 3 shows that, from a vertical perspective, the IHBA discovered theoretically optimal values for functions F1~F4, F7, and F9 in various dimensions, but, in the comparison algorithm, the m-EO did not find theoretically optimal values for functions F2~F4 in each dimension. Although the gaussian mutation method and the exploratory search mechanism based on the concept of population division and reconstruction used by the m-EO keep the diversity of the population and avoid the optimization stagnation of the algorithm, their performance in the optimization accuracy is not outstanding. In addition, the RDWOA did not find an optimal value at 500\textit{dim} on function F7, because an increase in dimensions could lead to a decrease in accuracy. For function F5, the optimization accuracy of the IHBA is higher than that of the comparison algorithms, which verifies the effectiveness of the improved strategy proposed in this paper. For function F6, the optimization accuracy of the IHBA is lower than the RDWOA and JSWOA. Both of these comparison algorithms introduce the adaptive inertia weight strategy to update the population position, which improves the exploration ability of the population in the early stage and the development ability in the later stage, and the optimization effect is improved. The optimization precision of each algorithm for function F8 is the same, and it is not comparable. For function F10, the accuracy of the IHBA in 30\textit{dim} is higher than that of the RDWOA, and the accuracy in 100\textit{dim} and 500\textit{dim} is second only to the RDWOA. Overall, the IHBA and RDWOA performed better on the unimodal and multimodal test functions shown in Table 3. From a horizontal perspective, the solution accuracy of the algorithm has a decreasing trend with the increase in the dimension, because the increase in the population dimension is accompanied by the expansion of the spatial range and the increase in the amount of calculation. However, compared with the four algorithms, the IHBA still has the highest optimization accuracy and the smallest standard deviation on most functions. Therefore, the IHBA has strong optimization performance and robustness in solving low-dimensional and high-dimensional problems, which further illustrates the competitive advantage of the IHBA in solving unimodal and multimodal functions.

It can be seen from Table 4 that when the IHBA solves functions F13~F15 and F18, the optimization accuracy ranks first. For the functions F11, F16, and F17, the IHBA’s optimization accuracy is second only to the RDWOA, JSWOA, and m-EO, but the IHBA and these three comparison algorithms have little difference in optimization accuracy and little influence. For function F12, the optimization effect of the IHBA is not significant, only better than the m-EO, but the optimization accuracy is in the same order of magnitude as the other three comparison algorithms. From the overall point of view in Table 4, the IHBA has outstanding performance on fixed-dimensional multimodal functions, in most functions, the search accuracy is closer to the theoretical optimal value and the standard deviation is smaller, which further verifies that the IHBA integrating the three improved strategies is competitive.

### 4.5. Wilcoxon Rank Sum Test

Wilcoxon rank sum test was performed at the significance level of $p = 0.05$ for the IHBA, HBA, whale optimization algorithm (WOA) [35], arithmetic optimization algorithm (AOA) [36], particle swarm optimization (PSO) [37], butterfly optimization algorithm (BOA) [38], HBA1, HBA2, and HBA3 to further evaluate the robustness and reliability of IHBA statistically. To verify that the experiment was fair, each algorithm was run 30 times independently, with a total of 500 iterations and a population size of 30, and an algorithm dimension of 30. When $p < 0.05$, reject the null hypothesis, indicating
statistically significant differences between the two comparison algorithms, otherwise, accept the null hypothesis, indicating no significant differences between the two comparison algorithms. In Table 5, NaN indicates it is not applicable, and "−", "−" and "×" indicate that IHBA is superior to, inferior to, and similar to comparison algorithms, respectively. Table 5 shows that the majority of the p-values are less than 0.05, showing substantial differences between the IHBA and other algorithms.

Table 5. Wilcoxon rank sum test results.

| Function | HBA | WOA | AOA | PSO | BOA | HBA1 | HBA2 | HBA3 |
|----------|-----|-----|-----|-----|-----|------|------|------|
|          |     |     |     |     |     |      |      |      |
| p1       | 8.01 × 10⁻⁹ | 8.01 × 10⁻⁹ | 8.01 × 10⁻⁹ | 8.01 × 10⁻⁹ | 8.01 × 10⁻⁹ | NaN  |     |      |
| p2       | 8.01 × 10⁻⁹ | 8.01 × 10⁻⁹ | NaN  | 8.01 × 10⁻⁹ | 8.01 × 10⁻⁹ | NaN  | 8.01 × 10⁻⁹ | NaN  |
| p3       | 8.01 × 10⁻⁹ | 8.01 × 10⁻⁹ | 8.01 × 10⁻⁹ | 8.01 × 10⁻⁹ | 8.01 × 10⁻⁹ | NaN  | 8.01 × 10⁻⁹ | NaN  |
| p4       | 8.01 × 10⁻⁹ | 8.01 × 10⁻⁹ | 8.01 × 10⁻⁹ | 8.01 × 10⁻⁹ | 8.01 × 10⁻⁹ | NaN  | 8.01 × 10⁻⁹ | NaN  |
| p5       | 8.01 × 10⁻⁹ | 8.01 × 10⁻⁹ | 8.01 × 10⁻⁹ | 8.01 × 10⁻⁹ | 8.01 × 10⁻⁹ | NaN  | 8.01 × 10⁻⁹ | NaN  |
| p6       | 8.01 × 10⁻⁹ | 8.01 × 10⁻⁹ | 8.01 × 10⁻⁹ | 8.01 × 10⁻⁹ | 8.01 × 10⁻⁹ | NaN  | 8.01 × 10⁻⁹ | NaN  |
| p7       | 8.01 × 10⁻⁹ | 8.01 × 10⁻⁹ | 8.01 × 10⁻⁹ | 8.01 × 10⁻⁹ | 8.01 × 10⁻⁹ | NaN  | 8.01 × 10⁻⁹ | NaN  |
| p8       | 8.01 × 10⁻⁹ | 8.01 × 10⁻⁹ | 8.01 × 10⁻⁹ | 8.01 × 10⁻⁹ | 8.01 × 10⁻⁹ | NaN  | 8.01 × 10⁻⁹ | NaN  |

4.6. CEC2014 Test Function Optimization Comparison

Compared with the basic test function in Table 1, the CEC2014 test function structure is more complex and optimization is more difficult. Optimization comparison was made between the IHBA and AOA, BOA, grey wolf optimizer (GWO) [39], sine cosine algorithm (SCA) [40], slime mold algorithm (SMA) [41], and tunicate swarm algorithm (TSA) [42] on the 30 CEC2014 test functions shown in Table 6. The CEC2014 function selected includes unimodal functions (UF), multimodal functions (MF), hybrid functions (HF), and composite functions (CF). To reduce the experimental error, each algorithm was independently run 30 times, with a total of 500 iterations and a population size of 30, and an algorithm dimension of 30. The mean and standard deviation of the experimental results independently run by each algorithm were taken and recorded in Table 7.

Table 6. CEC2014 test function.

| Function | Type | Range   | Optimal Value | Function | Type | Range   | Optimal Value |
|----------|------|---------|----------------|----------|------|---------|----------------|
| CEC01    | UF   | [-100, 100] | 100            | CEC16    | MF   | [-100, 100] | 1600           |
| CEC02    | UF   | [-100, 100] | 200            | CEC17    | HF   | [-100, 100] | 1700           |
| CEC03    | UF   | [-100, 100] | 300            | CEC18    | HF   | [-100, 100] | 1800           |
| CEC04    | MF   | [-100, 100] | 400            | CEC19    | HF   | [-100, 100] | 1900           |
| CEC05    | MF   | [-100, 100] | 500            | CEC20    | HF   | [-100, 100] | 2000           |
| CEC06    | MF   | [-100, 100] | 600            | CEC21    | HF   | [-100, 100] | 2100           |
| CEC07    | MF   | [-100, 100] | 700            | CEC22    | HF   | [-100, 100] | 2200           |
| Function | Index | AOA | BOA | GWO | SCA | SMA | TSA | HBA | IHBA |
|----------|-------|-----|-----|-----|-----|-----|-----|-----|-----|
| CEC08    | MF    | [−100, 100] | 800  | CEC23 | CF | [−100, 100] | 2300 |
| CEC09    | MF    | [−100, 100] | 900  | CEC24 | CF | [−100, 100] | 2400 |
| CEC10    | MF    | [−100, 100] | 1000 | CEC25 | CF | [−100, 100] | 2500 |
| CEC11    | MF    | [−100,100] | 1100 | CEC26 | CF | [−100, 100] | 2600 |
| CEC12    | MF    | [−100, 100] | 1200 | CEC27 | CF | [−100, 100] | 2700 |
| CEC13    | MF    | [−100, 100] | 1300 | CEC28 | CF | [−100, 100] | 2800 |
| CEC14    | MF    | [−100, 100] | 1400 | CEC29 | CF | [−100, 100] | 2900 |
| CEC15    | MF    | [−100, 100] | 1500 | CEC30 | CF | [−100, 100] | 3000 |

Table 7. CEC2014 test function optimization results comparison.
It can be seen from Table 7 that for the functions CEC06, CEC09 and CEC22, the average value of the GWO is the closest to the theoretical value. For the functions CEC08, CEC10 and CEC26, the average value of the PSO is the closest to the theoretical value. For the functions CEC29 and CEC30, the average value of the BOA is closest to the theoretical value. For most of the remaining functions, the average value of the IHBA is closest to the theoretical value, indicating that the IHBA has strong optimization ability. In addition, from the overall view of Table 7, the standard deviation of the IHBA is generally lower, thus confirming the stability of the IHBA.

### 5. Parameter Identification of Solar Photovoltaic Cell

This section applies the IHBA to the parameter identification of the SDM, DDM, and TDM of commercial silicon R.T.C French solar cells with a diameter of 57 mm and compares the results with other algorithms to verify the feasibility and superiority of the IHBA. The smaller the RMSE of the objective function, the closer the measured and computed values are, and the more accurate the algorithm is in finding the model parameters. Table 8 shows the value ranges of the parameters to be identified. The experiment’s temperature was set to 33 °C, and the irradiation intensity was set to 1000 W/m².

#### Table 8. Photovoltaic cell model parameters.

| Model Parameters | Range |
|------------------|-------|
| $I_{ph}/A$       | [0, 1]|
| $I_{sat}, I_{s}, I_{sh}/\mu A$ | [0, 1]|
| $A_1, A_2$       | [1, 2]|
| $A_3$            | [2, 5]|
| $R_s/\Omega$     | [0, 0.5]|
| $R_0/\Omega$     | [0, 100]|

| Parameter | Mean | Std |
|-----------|------|-----|
| CEC21     | $5.15 \times 10^7$ | $5.81 \times 10^7$ |
| CEC22     | $1.17 \times 10^4$ | $1.13 \times 10^4$ |
| CEC23     | $2.50 \times 10^3$ | $3.04 \times 10^{-10}$ |
| CEC24     | $2.60 \times 10^3$ | $9.94 \times 10^{-3}$ |
| CEC25     | $2.70 \times 10^3$ | $2.89 \times 10^{-11}$ |
| CEC26     | $2.80 \times 10^3$ | $1.86 \times 10^{-10}$ |
| CEC27     | $4.09 \times 10^3$ | $4.44 \times 10^{-1}$ |
| CEC28     | $5.11 \times 10^3$ | $2.89 \times 10^{-1}$ |
| CEC29     | $4.95 \times 10^8$ | $2.28 \times 10^{-4}$ |
| CEC30     | $5.72 \times 10^6$ | $3.61 \times 10^4$ |
5.1. Parameter Identification of SDM

The results of the IHBA’s experimental running are compared to those of the HBA, WOA, SMA, PSO, GWO, BOA, sparrow search algorithm (SSA) [43], and others. Each algorithm was run independently 30 times, with a total of 2000 iterations and a population size of 50. Table 9 shows the average results of parameter identification for different algorithms. Table 10 displays the minimum, maximum, mean, and standard deviation of the objective function RMSE solved by each algorithm. Figure 8 shows the RMSE mean convergence graph.

![RMSE convergence curve of SDM.](image)

### Table 9. Average result of SDM parameter identification.

| Algorithm | $I_{pk}$  | $I_{d}$  | $A_1$  | $R_s$  | $R_{sh}$ | RMSE   |
|-----------|-----------|-----------|--------|--------|---------|--------|
| HBA       | $7.6003 \times 10^{-1}$ | $4.2514 \times 10^{-1}$ | 1.5018 | $3.3511 \times 10^{-2}$ | $6.0306 \times 10^{-3}$ | $3.8553 \times 10^{-3}$ |
| WOA       | $7.5951 \times 10^{-1}$ | $5.9128 \times 10^{-1}$ | 1.5164 | $2.9602 \times 10^{-2}$ | $5.8346 \times 10^{-3}$ | $9.4148 \times 10^{-3}$ |
| SMA       | $6.8771 \times 10^{-1}$ | $5.7851 \times 10^{-1}$ | 1.6169 | $2.7496 \times 10^{-2}$ | $7.3965 \times 10^{-3}$ | $6.4019 \times 10^{-3}$ |
| PSO       | $7.6189 \times 10^{-1}$ | $8.0083 \times 10^{-1}$ | 1.5746 | $2.7496 \times 10^{-2}$ | $7.3965 \times 10^{-3}$ | $6.4019 \times 10^{-3}$ |
| SSA       | $7.6019 \times 10^{-1}$ | $4.1585 \times 10^{-1}$ | 1.5042 | $3.5581 \times 10^{-2}$ | $7.7523 \times 10^{-3}$ | $1.2845 \times 10^{-2}$ |
| GWO       | $7.6252 \times 10^{-1}$ | $6.2001 \times 10^{-1}$ | 1.5389 | $3.0076 \times 10^{-2}$ | $4.7638 \times 10^{-3}$ | $7.1697 \times 10^{-3}$ |
| BOA       | $7.5872 \times 10^{-1}$ | $6.4646 \times 10^{-1}$ | 1.5378 | $3.0754 \times 10^{-2}$ | $5.5493 \times 10^{-3}$ | $1.9456 \times 10^{-2}$ |
| IHBA      | $7.6101 \times 10^{-1}$ | $3.9445 \times 10^{-1}$ | 1.4951 | $3.4789 \times 10^{-2}$ | $5.538 \times 10^{-3}$ | $1.0272 \times 10^{-3}$ |

### Table 10. RMSE statistics result of SDM.

| RMSE  | Minimum   | Maximum  | Mean      | Std       |
|-------|-----------|----------|-----------|-----------|
| HBA   | $9.8602 \times 10^{-4}$ | $4.6014 \times 10^{-3}$ | $2.7154 \times 10^{-3}$ | $8.1932 \times 10^{-3}$ |
| WOA   | $1.0285 \times 10^{-3}$ | $3.8245 \times 10^{-2}$ | $5.4127 \times 10^{-3}$ | $1.1131 \times 10^{-2}$ |
| SMA   | $7.7339 \times 10^{-2}$ | $3.0085 \times 10^{-3}$ | $2.0451 \times 10^{-4}$ | $5.4230 \times 10^{-2}$ |
| PSO   | $1.2722 \times 10^{-3}$ | $3.8151 \times 10^{-2}$ | $1.4249 \times 10^{-2}$ | $1.7193 \times 10^{-3}$ |
| SSA   | $9.8700 \times 10^{-4}$ | $1.4847 \times 10^{-3}$ | $1.2713 \times 10^{-3}$ | $3.0391 \times 10^{-4}$ |
| GWO   | $1.1261 \times 10^{-3}$ | $3.8169 \times 10^{-3}$ | $6.1608 \times 10^{-3}$ | $1.0272 \times 10^{-2}$ |
| BOA   | $4.7867 \times 10^{-3}$ | $1.2368 \times 10^{-3}$ | $1.9886 \times 10^{-2}$ | $2.0707 \times 10^{-2}$ |
| IHBA  | $9.8262 \times 10^{-4}$ | $1.4480 \times 10^{-3}$ | $1.0836 \times 10^{-3}$ | $3.7091 \times 10^{-4}$ |
In the parameter identification results in Table 9, there is little difference in the identification results of $I_{ph}$ and $A_1$ among the different algorithms, while there is some difference in the identification results of other parameters. It can be seen from Table 10 that the RMSE solved by the IHBA has the smallest minimum, maximum, and mean, indicating that the IHBA has the highest accuracy for parameter identification. In addition, the standard deviation of RMSE solved by the IHBA is the smallest, that is, the stability of the identification process is good. In Figure 8, the IHBA can find the smallest RMSE value in the shortest time. To sum up, the IHBA has the highest parameter recognition accuracy for SDM and the fastest identification speed.

5.2. Parameter Identification of DDM

The results of the IHBA’s experimental running are compared to those of the HBA, WOA, SMA, PSO, GWO, BOA, SSA, and others. Each algorithm was run independently 30 times, with a total of 2000 iterations and a population size of 50. Table 11 shows the average results of parameter identification for the different algorithms. Table 12 displays the minimum, maximum, mean, and standard deviation of the objective function RMSE solved by each algorithm. Figure 9 shows the RMSE mean convergence graph.

![Figure 9. RMSE convergence curve of DDM.](image)

| Index | HBA | WOA | SMA | PSO | SSA | GWO | BOA | IHBA |
|-------|-----|-----|-----|-----|-----|-----|-----|------|
| $I_{ph}$ | $7.6081 \times 10^{-1}$ | $7.6060 \times 10^{-1}$ | $7.4359 \times 10^{-1}$ | $7.6235 \times 10^{-1}$ | $7.6273 \times 10^{-1}$ | $7.5591 \times 10^{-1}$ | $7.6100 \times 10^{-1}$ |
| $I_{sd1}$ | $3.7332 \times 10^{-1}$ | $5.5524 \times 10^{-1}$ | $5.8655 \times 10^{-1}$ | $8.5168 \times 10^{-1}$ | $6.4446 \times 10^{-1}$ | $6.7507 \times 10^{-1}$ | $3.6919 \times 10^{-1}$ |
| $I_{sd2}$ | $2.9999 \times 10^{-1}$ | $5.0478 \times 10^{-1}$ | $6.3973 \times 10^{-1}$ | $3.6667 \times 10^{-1}$ | $5.3711 \times 10^{-1}$ | $3.0643 \times 10^{-1}$ | $4.7833 \times 10^{-1}$ |
| $A_2$ | $1.4935$ | $1.5231$ | $1.6267$ | $1.5844$ | $1.4926$ | $1.5402$ | $1.5511$ | $1.4903$ |
| $R_s$ | $3.4893 \times 10^{-2}$ | $3.0552 \times 10^{-2}$ | $7.3144 \times 10^{-3}$ | $2.5054 \times 10^{-2}$ | $3.5916 \times 10^{-2}$ | $2.8466 \times 10^{-2}$ | $3.0321 \times 10^{-2}$ | $3.4988 \times 10^{-2}$ |
| $R_{sh}$ | $5.2926 \times 10^{-1}$ | $6.0604 \times 10^{-1}$ | $5.0031 \times 10^{-1}$ | $7.0622 \times 10^{-1}$ | $6.9732 \times 10^{-1}$ | $4.1533 \times 10^{-1}$ | $5.7412 \times 10^{-1}$ | $5.3916 \times 10^{-1}$ |
| RMSE | $2.2968 \times 10^{-3}$ | $6.8009 \times 10^{-3}$ | $2.0474 \times 10^{-3}$ | $1.0450 \times 10^{-2}$ | $1.8936 \times 10^{-3}$ | $8.8967 \times 10^{-3}$ | $1.6850 \times 10^{-2}$ | $1.2743 \times 10^{-3}$ |
Table 12. RMSE statistics result of DDM.

| Algorithm | Minimum       | Maximum       | Mean          | Std           |
|-----------|---------------|---------------|---------------|---------------|
| HBA       | $9.8602 \times 10^{-4}$ | $4.6014 \times 10^{-2}$ | $2.7154 \times 10^{-3}$ | $8.1932 \times 10^{-3}$ |
| WOA       | $1.0285 \times 10^{-3}$ | $3.8245 \times 10^{-2}$ | $5.4127 \times 10^{-3}$ | $1.1131 \times 10^{-2}$ |
| SMA       | $7.7339 \times 10^{-2}$ | $3.0085 \times 10^{-1}$ | $2.0451 \times 10^{-1}$ | $5.4230 \times 10^{-2}$ |
| PSO       | $1.2722 \times 10^{-3}$ | $3.8151 \times 10^{-2}$ | $1.4249 \times 10^{-2}$ | $1.7193 \times 10^{-2}$ |
| SSA       | $9.8700 \times 10^{-4}$ | $1.4847 \times 10^{-3}$ | $1.2713 \times 10^{-3}$ | $3.3391 \times 10^{-4}$ |
| GWO       | $1.1261 \times 10^{-3}$ | $3.8169 \times 10^{-2}$ | $6.1608 \times 10^{-3}$ | $1.0272 \times 10^{-2}$ |
| BOA       | $4.7867 \times 10^{-3}$ | $1.2368 \times 10^{-1}$ | $1.9886 \times 10^{-2}$ | $2.0707 \times 10^{-2}$ |
| IHBA      | $9.8163 \times 10^{-4}$ | $1.4480 \times 10^{-3}$ | $1.0836 \times 10^{-3}$ | $3.1091 \times 10^{-4}$ |

In the parameter identification results in Table 11, there is little difference in the identification results of $I_{ph}$, $A_1$ and $A_2$ among the different algorithms, while there is some difference in the identification results of other parameters. It can be seen from Table 12 that the minimum, maximum, mean, and standard deviation of RMSE solved by the IHBA are the smallest, indicating that the IHBA has the highest accuracy of parameter identification and the identification process is stable. In Figure 9, the IHBA has the fastest identification speed. To summarize, the IHBA outperforms the comparison algorithms in terms of identification speed, accuracy, and stability.

5.3. Parameter Identification of TDM

The results of the IHBA’s experimental running are compared to those of the HBA, WOA, SMA, PSO, GWO, BOA, SSA, and others. Each algorithm was run independently 30 times, with a total of 2000 iterations and a population size of 50. Table 13 shows the average results of parameter identification for the different algorithms. Table 14 displays the minimum, maximum, mean, and standard deviation of the objective function RMSE solved by each algorithm. Figure 10 shows the RMSE mean convergence graph.
Table 13. Average result of TDM parameter identification.

| Index | HBA     | WOA    | SMA    | PSO    | SSA    | GWO    | BOA    | IHBA   |
|-------|---------|--------|--------|--------|--------|--------|--------|--------|
| $I_{ph}$ | 7.6081 × 10^{-1} | 7.6090 × 10^{-1} | 7.2117 × 10^{-1} | 7.6309 × 10^{-1} | 7.6309 × 10^{-1} | 7.6309 × 10^{-1} | 7.6309 × 10^{-1} |
| $I_{sd}$ | 3.9365 × 10^{-1} | 6.0423 × 10^{-1} | 5.4580 × 10^{-1} | 8.2321 × 10^{-1} | 8.2321 × 10^{-1} | 8.2321 × 10^{-1} | 8.2321 × 10^{-1} |
| $I_{sh}$ | 5.3927 × 10^{-1} | 4.3814 × 10^{-1} | 4.8935 × 10^{-1} | 5.6667 × 10^{-1} | 5.6667 × 10^{-1} | 5.6667 × 10^{-1} | 5.6667 × 10^{-1} |
| $I_{sd}$ | 5.6383 × 10^{-1} | 5.9062 × 10^{-1} | 5.0189 × 10^{-1} | 5.3360 × 10^{-1} | 5.3360 × 10^{-1} | 5.3360 × 10^{-1} | 5.3360 × 10^{-1} |
| $A_1$ | 1.4970 | 1.5352 | 1.5951 | 1.5771 | 1.5771 | 1.5771 | 1.5771 |
| $A_2$ | 1.5736 | 1.5708 | 1.5031 | 1.4333 | 1.4333 | 1.4333 | 1.4333 |
| $A_3$ | 4.0897 | 3.6015 | 3.6414 | 3.6986 | 3.6986 | 3.6986 | 3.6986 |
| $R_s$ | 3.4750 × 10^{-2} | 3.2910 × 10^{-2} | 5.4821 × 10^{-2} | 2.2278 × 10^{-2} | 2.2278 × 10^{-2} | 2.2278 × 10^{-2} | 2.2278 × 10^{-2} |
| $R_{sh}$ | 6.0829 × 10^{-2} | 6.1846 × 10^{-2} | 5.5039 × 10^{-2} | 5.9474 × 10^{-2} | 5.9474 × 10^{-2} | 5.9474 × 10^{-2} | 5.9474 × 10^{-2} |
| RMSE | 3.7358 × 10^{-3} | 5.4162 × 10^{-3} | 1.9641 × 10^{-3} | 2.0756 × 10^{-3} | 1.2340 × 10^{-3} | 6.3537 × 10^{-3} | 1.4184 × 10^{-2} | 1.0291 × 10^{-3} |

Table 14. RMSE statistics result of TDM.

| RMSE | Minimum | Maximum | Mean | Std |
|------|---------|---------|------|-----|
| HBA  | 9.8602 × 10^{-4} | 3.8151 × 10^{-2} | 2.2831 × 10^{-3} | 6.7797 × 10^{-3} |
| WOA  | 9.9843 × 10^{-4} | 4.6014 × 10^{-2} | 8.5781 × 10^{-3} | 1.4332 × 10^{-2} |
| SMA  | 8.3990 × 10^{-2} | 3.0677 × 10^{-1} | 2.1226 × 10^{-1} | 6.0214 × 10^{-2} |
| PSO  | 9.8903 × 10^{-4} | 5.1840 × 10^{-3} | 2.0009 × 10^{-3} | 7.7885 × 10^{-4} |
| SSA  | 9.8891 × 10^{-4} | 1.5074 × 10^{-3} | 1.2064 × 10^{-3} | 6.7854 × 10^{-4} |
| GWO  | 1.1439 × 10^{-3} | 3.8121 × 10^{-2} | 6.3979 × 10^{-3} | 1.0433 × 10^{-2} |
| BOA  | 4.3489 × 10^{-3} | 4.5074 × 10^{-2} | 1.6710 × 10^{-2} | 8.4708 × 10^{-3} |
| IHBA | 9.8015 × 10^{-4} | 3.8151 × 10^{-2} | 1.0049 × 10^{-3} | 2.0265 × 10^{-4} |

In the parameter identification results in Table 13, there is little difference in the identification results of $I_{ph}$, $A_1$, $A_2$, and $A_3$ among the different algorithms, while there is some difference in the identification results of other parameters. It can be seen from Table 14 that the RMSE solved by the IHBA has the smallest minimum, maximum, mean, and standard deviation, indicating that the IHBA is more accurate in identifying model parameters and the identification process is stable. As can be seen from Figure 10, the IHBA has the fastest parameter recognition speed for the model. In conclusion, the IHBA can accurately identify the parameters of the TDM at the fastest speed while maintaining a steady identification process.

6. Conclusions

To accurately identify the parameters of the solar photovoltaic cell model, this paper proposes an improved honey badger algorithm called the IHBA. The IHBA incorporates three improved strategies: a spiral exploration mechanism, the density update factor with quasi-cosine varying the law, and a pinhole imaging strategy. Through 18 benchmark functions, Wilcoxon rank sum statistical test, and 30 CEC2014 functions, it is verified that the improved algorithm can reduce the probability of the population falling into the local optimum, and improve the optimization accuracy and convergence speed of the original algorithm. Finally, the IHBA is applied to the SDM, DDM, and TDM parameter identification of a 57 mm diameter commercial silicon R.T.C French solar cell. Compared with the eight algorithms, the IHBA can calculate the minimum objective function RMSE of the three models with the fastest speed, that is, it can accurately identify the parameters to be solved for the models. In future work, we will consider applying the IHBA to multi-objective optimization and dynamic optimization problems.

Author Contributions: Conceptualization, W.L.; methodology, W.L.; software, W.L.; validation, W.L., Q.H., L.Y. and H.J.; formal analysis, Q.H. and H.J.; writing—original draft preparation, W.L.;
writing—review and editing, W.L., Q.H. and H.J.; resources, Q.H.; visualization, L.Y. All authors have read and agreed to the published version of the manuscript.

**Funding:** The research was funded by the National Natural Science Foundation of China “Research on the Evidence Chain Construction from the Analysis of the Investigation Documents (62166006)”, National Natural Science Foundation of China “Rural spatial restructuring in poverty-stricken mountainous areas of Guizhou based on Spatial equity: A case study of Dianqiangui Rocky Desertification Area (41861038)”, National Key Research and Development Program “Research and Development of Emergency Response and Collaborative Command System with Holographic Perception of Traffic Network Disaster (2020YFC1512002)”, Guizhou Provincial Science and Technology Projects (Guizhou Science Foundation-ZK [2021] General 335).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** There are no conflict of interest.

**References**

1. Shahbaz, M.; Raghubita, C.; Chittedi, K.R.; Jiao, Z.; Vo, X.V. The effect of renewable energy country attractive index. *Energy* 2020, 207, 118162. https://doi.org/10.1016/j.energy.2020.118162.

2. Adams, S.; Klobodu, E.K.M.; Apio, A. Renewable and non-renewable energy, regime type and economic growth. *Renew. Energy* 2018, 125, 755–767. https://doi.org/10.1016/j.renene.2018.02.135.

3. Güney, T. Renewable energy, non-renewable energy and sustainable development. *Int. J. Sustain. Dev. World Ecol.* 2019, 26, 389–397. https://doi.org/10.1080/13504509.2019.1595214.

4. Ahmad, M.; İşik, C.; Jabeen, G.; Ali, T.; Ozturk, I.; Atchike, D.W. Heterogeneous links among urban concentration, non-renewable energy use intensity, economic development, and environmental emissions across regional development levels. *Sci. Total Environ.* 2021, 765, 144527. https://doi.org/10.1016/j.scitotenv.2020.144527.

5. Rasoulinezhad, E.; Saboori, B. Panel estimation for renewable and non-renewable energy consumption, economic growth, CO₂ emissions, the composite trade intensity, and financial openness of the commonwealth of independent states. *Environ. Sci. Pollut. Res.* 2018, 25, 17354–17370. https://doi.org/10.1007/s11356-018-1827-3.

6. Agrawal, S.; Soni, R. Renewable Energy: Sources, Importance and Prospects for Sustainable Future. *Energy Cris. Chall. Solut.* 2021, 131–150. https://doi.org/10.1002/9781119741503.ch7.

7. Xu, X.; Wei, Z.; Ji, Q.; Wang, C.; Gao, G. Global renewable energy development: Influencing factors, trend predictions and countermeasures. *Restor. Policy* 2019, 63, 101470. https://doi.org/10.1016/j.recpsolpolr.2019.101470.

8. Gielen, D.; Boshell, F.; Saygin, D.; Bazilian, M.D.; Wagner, N.; Gorini, R. The role of renewable energy in the global energy transformation. *Energy Strateg. Rev.* 2019, 24, 38–50. https://doi.org/10.1016/j.esr.2019.01.006.

9. Al-Shahri, A.O.; Ismail, F.B.; Hannan, M.A.; Lipu, M.S.H.; Al-Shetwi, A.Q.; Begum, R.A.; Al-Muhsen, N.F.O.; Soujeri, E. Solar photovoltaic energy optimization methods, challenges and issues: A comprehensive review. *J. Clean. Prod.* 2021, 284, 125465. https://doi.org/10.1016/j.jclepro.2020.125465.

10. Benda, V.; Černá, L. PV cells and modules—State of the art, limits and trends. *Helion* 2020, 6, e05666. https://doi.org/10.1002/1751-6101.2020.e05666.

11. Yilduran, N.; Tacer, E. Identification of photovoltaic cell single diode discrete model parameters based on datasheet values. *Sol. Energy* 2016, 127, 175–183. https://doi.org/10.1016/j.solener.2016.01.024.

12. Hassan Hosseini, S.M.; Keymanesh, A.A. Design and construction of photovoltaic simulator based on dual-diode model. *Sol. Energy* 2016, 137, 594–607. https://doi.org/10.1016/j.solener.2016.09.001.

13. Çalışan, M.; Abdel Aleem, S.H.E.; Zoba, A.F. A new approach for parameters estimation of double and triple diode models of photovoltaic cells based on iterative Lambert W function. *Sol. Energy* 2021, 218, 392–412. https://doi.org/10.1016/j.solener.2021.02.038.

14. Lineykin, S.; Averbukh, M.; Kuperman, A. An improved approach to extract the single-diode equivalent circuit parameters of a photovoltaic cell/panel. *Renew. Sustain. Energy Rev.* 2014, 30, 282–289. https://doi.org/10.1016/j.rser.2013.10.015.

15. Pardhu, B.S.S.G.; Kota, V.R. Radial movement optimization based parameter extraction of double diode model of solar photovoltaic cell. *Sol. Energy* 2021, 213, 312–327. https://doi.org/10.1016/j.solener.2020.11.046.

16. Oliva, D.; Elaziz, M.A.; Elsheikh, A.H.; Ewees, A.A. A review on meta-heuristics methods for estimating parameters of solar cells. *J. Power Sources* 2019, 435, 126683. https://doi.org/10.1016/j.jpowsour.2019.05.089.

17. Ben Hmamou, D.; Elyaquouti, M.; Arjdal, E.; Chaoufi, J.; Saadaoui, D.; Lidaighbi, S.; Aqel, R. Particle swarm optimization approach to determine all parameters of the photovoltaic cell. *Mater. Today Proc.* 2022, 52, 7–12. https://doi.org/10.1016/j.matpr.2021.10.083.
Sustainability 2022, 14, 8897

18. Toledo, F.J.; Blanes, J.M.; Galvano, V. Two-Step Linear Least-Squares Method for Photovoltaic Single-Diode Model Parameters Extraction. IEEE Trans. Ind. Electron. 2018, 65, 6301–6308. https://doi.org/10.1109/TIE.2018.2793216.

19. Ayodele, T.R.; Ogunjuyigbe, A.S.O.; Eko, E.E. Evaluation of numerical algorithms used in extracting the parameters of a single-diode photovoltaic model. Sustain. Energy Technol. Assess. 2016, 13, 51–59. https://doi.org/10.1016/j.sreta.2015.11.003.

20. Chaibi, Y.; Allouhi, A.; Salhi, M. A simple iterative method to determine the electrical parameters of photovoltaic cell. J. Clean. Prod. 2020, 269, 122363. https://doi.org/10.1016/j.jclepro.2020.122363.

21. Rasheed, M.; SuhaShihab; Alabdali, O.; Hassan, H.H. Parameters Extraction of a Single-Diode Model of Photovoltaic Cell Using False Position Iterative Method. J. Phys. Conf. Ser. 2021, 1879, 032113. https://doi.org/10.1088/1742-6596/1879/3/032113.

22. Çimen, M.E.; Garip, Z.; Boz, A.F.; Karayel, D. Firefly Algorithm and Particle Swarm Optimization for photovoltaic parameters identification based on single model. In Proceedings of the 2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Ankara, Turkey, 19–21 October 2018; pp. 1–5. https://doi.org/10.1109/ISMSIT.2018.8567288.

23. Elazab, O.S.; Hasanien, H.M.; Alsaidan, I.; Abdelaziz, A.Y.; Muyeen, S.M. Parameter Estimation of Three Diode Photovoltaic Model Using Grasshopper Optimization Algorithm. Energies 2020, 13, 497. https://doi.org/10.3390/en13020497.

24. Aydin, O.; Gozde, H.; Dursun, M.; Taplamacıoğlu, M.C. Comparative Parameter Estimation of Single Diode PV-Cell Model by Using Sine-Cosine Algorithm and Whale Optimization Algorithm. In Proceedings of the 2019 6th International Conference on Electrical and Electronics Engineering (ICEEE), Istanbul, Turkey, 16–17 April 2019; pp. 65–68. https://doi.org/10.1109/ICEEE2019.2019.00020.

25. Saxena, A.; Sharma, A.; Shekhawat, S. Parameter extraction of solar cell using intelligent grey wolf optimizer. Evol. Intell. 2022, 15, 167–183. https://doi.org/10.1007/s12065-020-00499-1.

26. Kumar, C.; Raj, T.D.; Premkumar, M.; Raj, T.D. A new stochastic slime mould optimization algorithm for the estimation of photovoltaic cell parameters. Optik 2020, 223, 165277. https://doi.org/10.1016/j.ijleo.2020.165277.

27. Hashim, F.A.; Houssein, E.H.; Hussain, K.; Mabrouk, M.S.; Al-Atabany, W. Honey Badger Algorithm: New metaheuristic algorithm for solving optimization problems. Math. Comput. Simul. 2022, 192, 84–110. https://doi.org/10.1016/j.matcom.2021.08.013.

28. Han, E.; Ghadimi, N. Model identification of proton-exchange membrane fuel cells based on a hybrid convolutional neural network and extreme learning machine optimized by improved honey badger algorithm. Sustain. Energy Technol. Assess. 2022, 52, 102005. https://doi.org/10.1016/j.seta.2022.102005.

29. Hosny, R.A.; Abd Elaziz, M.; Ali Ibrahim, R. Enhanced Feature Selection Based on Integration Containment Neighborhoods Rough Set Approximations and Binary Honey Badger Optimization. Comput. Intell. Neurosci. 2022, 2022, 3991870. https://doi.org/10.1155/2022/3991870.

30. Ni, B.; Wang, S.; Ma, Y.; Li, G. An Optimized Extreme Learning Machine for Predicting Loan Default in Peer-to-peer Lending Based on an Enhanced Honey Badger Algorithm. In Proceedings of the 2022 7th International Conference on Control and Robotics Engineering (ICCRE), Beijing, China, 15–17 April 2022; pp. 139–146. https://doi.org/10.1109/ICCRE55123.2022.9770229.

31. Fan, Q.; Chen, Z.; Li, Z.; Xia, Z.; Yu, J.; Wang, D. A new improved whale optimization algorithm with joint search mechanisms for high-dimensional global optimization problems. Eng. Comput. 2021, 37, 1851–1878. https://doi.org/10.1007/s00366-019-00917-8.

32. Gupta, S.; Deep, K.; Mirjalili, S. An efficient equilibrium optimizer with mutation strategy for numerical optimization. Appl. Soft Comput. 2020, 96, 106542. https://doi.org/10.1016/j.asoc.2020.106542.

33. Chen, H.; Li, W.; Yang, X. A whale optimization algorithm with chaos mechanism based on quasi-opposition for global optimization problems. Expert Syst. Appl. 2020, 158, 113612. https://doi.org/10.1016/j.eswa.2020.113612.

34. Chen, H.; Yang, C.; Heidari, A.A.; Zhao, X. An efficient double adaptive random space reinforced whale optimization algorithm. Expert Syst. Appl. 2020, 154, 113018. https://doi.org/10.1016/j.eswa.2019.113018.

35. Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. Adv. Eng. Softw. 2016, 95, 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008.

36. Abualigah, L.; Diabat, A.; Mirjalili, S.; Abd Elaziz, M.; Gandomi, A.H. The Arithmetic Optimization Algorithm. Comput. Methods Appl. Mech. Eng. 2021, 376, 116309. https://doi.org/10.1016/j.cma.2020.116309.

37. Poli, R.; Kennedy, J.; Blackwell, T. Particle swarm optimization. Swarm Intell. 2007, 1, 33–57. https://doi.org/10.1007/s11721-007-0002-0.

38. Arora, S.; Singh, S. Butterfly optimization algorithm: A novel approach for global optimization. Soft. Comput. 2019, 23, 715–734. https://doi.org/10.1007/s00500-018-3102-4.

39. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. Adv. Eng. Softw. 2014, 69, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007.

40. Mirjalili, S.; Mirjalili, S.M.; Lewis, A. Grey Wolf Optimizer. Adv. Eng. Softw. 2014, 69, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007.

41. 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. https://doi.org/10.1016/j.future.2020.03.055.

42. Kaur, S.; Awasthi, L.K.; Sangal, A.L.; Dhiman, G. Tunicate Swarm Algorithm: A new bio-inspired based metaheuristic paradigm for global optimization. Eng. Appl. Artif. Intell. 2020, 90, 103541. https://doi.org/10.1016/j.engappai.2020.103541.

43. Xue, J.; Shen, B. A novel swarm intelligence optimization approach: Sparrow search algorithm. Syst. Sci. Control Eng. 2020, 8, 22–34. https://doi.org/10.1080/21642583.2019.1708830.