energies

Article

An Efficient Parameter Estimation Algorithm for Proton Exchange Membrane Fuel Cells

Mohamed Abdel-Basset [1], Reda Mohamed [1] and Victor Chang [2,*]

[1] Department of Computer Science, Faculty of Computers and Informatics, Zagazig University, Zagazig 44519, Egypt; mohamedbasset@zu.edu.eg (M.A.-B.); redamoh@zu.edu.eg (R.M.)
[2] Cybersecurity, Information Systems and AI Research Group, School of Computing, Engineering and Digitial Technologies, Teesside University, Middlesbrough TS1 3BX, UK
* Correspondence: victorchang.research@gmail.com or V.Chang@tees.ac.uk

Abstract: The proton exchange membrane fuel cell (PEMFC) is a favorable renewable energy source to overcome environmental pollution and save electricity. However, the mathematical model of the PEMFC contains some unknown parameters which have to be accurately estimated to build an accurate PEMFC model; this problem is known as the parameter estimation of PEMFC and belongs to the optimization problem. Although this problem belongs to the optimization problem, not all optimization algorithms are suitable to solve it because it is a nonlinear and complex problem. Therefore, in this paper, a new optimization algorithm known as the artificial gorilla troops optimizer (GTO), which simulates the collective intelligence of gorilla troops in nature, is adapted for estimating this problem. However, the GTO is suffering from local optima and low convergence speed problems, so a modification based on replacing its exploitation operator with a new one, relating the exploration and exploitation according to the population diversity in the current iteration, has been performed to improve the exploitation operator in addition to the exploration one. This modified variant, named the modified GTO (MGTO), has been applied for estimating the unknown parameters of three PEMFC stacks, 250 W stack, BCS-500W stack, and SR-12 stack, used widely in the literature, based on minimizing the error between the measured and estimated data points as the objective function. The outcomes obtained by applying the GTO and MGTO on those PEMFC stacks have been extensively compared with those of eight well-known optimization algorithms using various performance analyses, best, average, worst, standard deviation (SD), CPU time, mean absolute percentage error (MAPE), and mean absolute error (MAE), in addition to the Wilcoxon rank-sum test, to show which one is the best for solving this problem. The experimental findings show that MGTO is the best for all performance metrics, but CPU time is competitive among all algorithms.

Keywords: fuel cells; PEMFC; artificial gorilla troops optimizer; modeling

check for updates

Citation: Abdel-Basset, M.; Mohamed, R.; Chang, V. An Efficient Parameter Estimation Algorithm for Proton Exchange Membrane Fuel Cells. *Energies* **2021**, *14*, 7115. https://doi.org/10.3390/en14217115

Academic Editors: Josep M. Guerrero, Ferdinanda Ponci, Helen C. Leligou, Elisa Peñalvo-López and Constantinos S. Psomopoulos

Received: 28 September 2021
Accepted: 22 October 2021
Published: 1 November 2021

1. Introduction

The proton exchange membrane fuel cell (PEMFC) is an important renewable energy source that has attracted the attention of the world over the last decades. The main advantage is to convert hydrogen fuel into electrical energy, continuously and directly, for disposing of environmental pollution caused by the traditional energy sources [1,2]. Polarization curves that depict the relationship between current and voltage are so important and need accurate modeling. Several mechanistic and empirical, or semi-empirical, models have been designed to model the PEMFCs; among them, the semi-empirical model is more suitable for engineering purposes due to its publicity and reputation to solve problems more easily [1,3–7]. However, unfortunately, in this model, some non-mechanistic terms have to be accurately identified to design a model simulating the real PEMFCs performance better; this problem is known as the parameter estimation of PEMFCs.

This problem belongs to the optimization problems and could be tackled by a specific optimization algorithm, but not all these algorithms, especially traditional algorithms, could achieve the required accuracy due to the complex nonlinearity, multi-variability, and strong couple of the PEMFC system. Having a strong optimization algorithm to deal with this problem is inevitable. Therefore, in literature, researchers have tried new modern algorithms, namely metaheuristic algorithms, having a strong performance for solving several nonlinear optimization problems with high dimensions in a reasonable time [8–11]. Some of those that have worked to estimate these parameters will be reviewed in the rest of this section, to articulate their contributions besides their advantages and disadvantages.

In [12], two metaheuristic algorithms, namely Harris Hawks' optimization (HHO) and atom search optimization (ASO), have been investigated regarding their performance for finding the unknown parameters of the PEMFCs using the sum of square errors as the objective function. Those algorithms were assessed using three different commercial PEMFC stacks, 500 W SR-12 PEM, BCS 500-W PEM, and 250 W stack, under various operation conditions. In addition, the outcomes obtained by those algorithms have been extensively compared with those of several existing optimization techniques to determine their accuracy and speed; those experiments affirm that HHO is the best. However, this algorithm still suffers from falling into local optimal and low convergence speed. Those are observed from the employed maximum of function evaluations which might surpass 60,000 on account of the used population size.

For finding more accurate parameter values, in [13], a new objective function has been designed and employed with the chaos embedded particle swarm optimization algorithm (CEPSO) for estimating the unknown parameters of three commercial PEMFCs stack. The findings have shown the superiority of this algorithm compared to some existing algorithms. This algorithm used a population size and a maximum iteration of 100 and 100, respectively, equal to 10,000 function evaluations. This large number notifies that this algorithm has low convergence speed as its main disadvantage. Additionally, Singla et al. [14] have adopted a newly-published metaheuristic algorithm, known as black widow optimization (BWO), for finding the parameter estimation of the PEMFCs. BWO's outcomes were compared with those obtained by five metaheuristic algorithms: particle swarm optimization (PSO), multi-verse optimizer (MVO), whale optimization algorithm (WOA), sine cosine algorithm (SCA), and grey wolf optimization (GWO). The experimental outcomes affirmed that BWO is better than all.

Zhu [15] has employed another metaheuristic algorithm, known as the Adaptive Sparrow Search Algorithm (ASSA), to tackle this problem by minimizing the error between the measured and estimated current as of the objective function. The experimental findings for three case studies, Ballard Mark V, Horizon H-12, and NedStack PS6, elaborated the superiority of ASSA compared to three other algorithms. In [16], the slime mold optimizer (SMA), which was recently proposed for tackling optimization problems and could fulfill superior outcomes, was employed for tackling this problem. The outcomes obtained using SMA outperform those of the compared algorithms. Diab et al. [17] suggested a new parameter estimation model for PEMFCs based on the coyote optimization algorithm using the sum of square error as an objective function. In order to demonstrate its efficiency, it was evaluated using two PEMFCs stacks and compared four optimization algorithms to show its superiority. Table 1 describes the contributions and disadvantages of some recently published metaheuristic algorithms for estimating the parameter of PEMFCs.

Table 1. Reviewing some published parameter estimation techniques for the PEMFCs.

Algorithms	Contributions	Disadvantages
Slime mold algorithm (SMA) [16]	• Employing SMA for estimating the unknown parameters of the fuel cell. • Its performance has been compared to six other metaheuristic algorithms and could overcome all.	• It still suffers from low convergence speed because it needed about 1000 iterations to be converged to the best-obtained solution. In addition, the authors did not state the used population size.
Modified farmland fertility optimization algorithm (MFFA) [18]	• Proposing two stages-based modifications to the standard farmland fertility algorithm: the first is employing the levy flight strategy to improve its exploration and exploitation operators and the other integrates a sine-cosine function to generate various solutions around the best-so-far solution. • MFFA has been employed for finding the unknown parameters of the PEMFC models and compared with some rival optimizers to check their superiority.	• Creating around 100 solutions to search for the optimal fitness value, and this number consumes exactly 10,000 functions evaluations when the maximum iteration is of 100, as used in MFFA. From this, it is shown that the convergence speed of MFFA still needs significant improvements.
Moth–flame optimization (MFO) [19]	• The classical MFO has been combined with the notion of uncertainty for finding the unknown parameters of three well-known commercial PEMFC models and the obtained outcomes were compared with those of some well-established algorithms.	• Both convergence speed and CPU time of this algorithm have not been stated in the published article and hence this algorithm is not preferred because it might need expensive computational costs and a huge number of function evaluations to find the near-optimal solution.
Coyote optimization algorithm (COA) [20]	• In this paper, the COA was adapted for tackling the parameter estimation of the PEMFC models. • Investigated using two PEMFCs, 250 W stack and Ned Stack PS6, and compared to four optimizers to observe its effectiveness. • The obtained outcomes affirmed its superiority.	• This algorithm consumes a huge number of function evaluations up to 40,000 to converge to the near-optimal solution. • The CPU time consumed is not stated.
Modified monarch butterfly optimization (MMBO) [21]	• Integrating mutation and self-adaptive mechanisms with the standard monarch butterfly optimization (MBO) has been performed to propose a new MBO, called modified MBO. It has better exploration and exploitation operators that enable it to avoid becoming stuck in local minima by accelerating the convergence speed. • Employing MMBO for finding the unknown parameters of 250 MW PEMFC stack under various operating conditions, in addition to comparing its performance to four state-of-the-arts, has been performed in this paper.	• The stability of the performance of this algorithm with more PEMFC stacks with various characteristics has not been investigated and hence its performance might deteriorate with other PEMFC stacks.

Table 1. *Cont.*

Algorithms	Contributions	Disadvantages
Shark smell optimizer (SSO) [22]	• In this paper, a new nature-inspired optimization algorithm known as SSO has been adapted for estimating the unknown parameters of the PEMFC models. • Five commercial PEMFC stacks were employed to investigate its performance, and the obtained outcomes were compared with five optimization algorithms to observe its proficiency. • The experimental findings affirmed that SSO is better.	• It still suffers from low convergence speed.
Improved grass fibrous root optimization algorithm [23]	• This paper presented a new parameter estimation technique for PEMFCs based on improving the grass fibrous root optimization algorithm. • The outcomes of this algorithm have been compared with those of some well-known algorithms to clarify its superiority.	• This algorithm employed 100 search agents to search for a better solution within 100 iterations, any around 10,000 function evaluations and this shows that the algorithms need more improvements to overcome its low convergence speed.
Improved barnacles mating optimization algorithm (IBMO) [24]	• In this work, an improved variant of the barnacles mating optimization algorithm has been suggested for finding the unknown parameters of two various PEMFC stacks: Horizon 500 W PEMFC and NedStack PS6. • The findings of IBMO were extensively compared to those of five well-known optimization techniques.	• Still suffers from the low convergence speed and falling into local optima.
Hybrid teaching learning based optimization–differential evolution algorithm [25]	• The teaching learning-based optimization algorithm has been combined with the differential evolution to propose a hybrid variant, namely TLBO-DE, employed for finding the unknown parameters of the PEMFC stacks. • This algorithm was investigated using a 250 MW PEMFC stack for various operating conditions then the obtained outcomes were compared with those of four well-established metaheuristic algorithms.	• This algorithm used 200,000 function evaluations and this huge number notifies that the algorithm also has a problem in terms of the convergence speed.

There are several other parameter estimation PEMFC techniques based on meta-heuristic algorithms: improved chaotic grey wolf optimization algorithm [26], modified farmland fertility optimizer [18], hunger games search algorithm [27], improved version of the Archimedes optimization algorithm [28], moth–flame optimization [19], Levenberg–Marquardt backpropagation algorithm [29], whale optimization algorithm [30], marine predator algorithm optimizer [31], pathfinder algorithm [32], hybrid water cycle moth–flame optimization algorithm [33], improved fluid search optimization algorithm [34], Seeker optimization algorithm [35], improved grass fibrous root optimization algorithm [36], developed coyote optimization algorithm [37], improved TLBO with elite strategy [38], developed owl search algorithm [39], modified artificial electric field algorithm [40], Supply–Demand-Based Optimization Algorithm [41], convolutional neural network optimized by balanced deer hunting optimization algorithm [42], and chaos game optimization technique [43].

The algorithms mentioned above have still suffered from two common problems, low convergence speed and falling into local minima, which prevent them from reaching accurate outcomes in fewer function evaluations. Therefore, in this paper, a new metaheuristic algorithm, namely the artificial gorilla troops optimizer (GTO), proposed recently for tackling the CEC optimization problems, has been adapted for tackling this problem due to its significant success achieved for the CEC problems. In addition, an effective modification has been performed on the GTO to improve its exploration and exploitation capability in a new strong variant called the modified GTO (MGTO). Four well-known commercial PEMFC stacks were employed to investigate the performance of the GTO and MGTO, and the obtained outcomes were compared with eight well-known metaheuristic algorithms to check its superiority for finding the unknown parameters which minimize the error between the measured and estimated current. Finally, those conducted experiments show that the MGTO is better than all the others for accuracy, convergence speed, and stability. The main contributions within this paper are:

- To adapt the GTO for tackling the parameter estimation of PEMFC, in addition to making a strong modification to produce a new variant, abbreviated as MGTO, having better exploration and exploitation capabilities.
- Comparing the performance of the GTO and MGTO with eight metaheuristic algorithms has shown that the MGTO is superior in terms of convergence speed, stability, and final accuracy.

The remainder of this paper is organized as follows: Section 2 explains the mathematical model of the PEMFC; Section 3 presents the standard GTO; Section 4 discusses the steps of the proposed parameter estimation algorithm, MGTO; comparison and discussions are shown in Section 5; and the last section involves the conclusion and future work.

2. The Mathematical Model of PEMFC

A PEMFC is compounded of two electrodes, an anode and a cathode, as well as an electrolyte between them, as depicted in Figure 1 [44]. The chemical reactions start with the hydrogen (H_2) converted at the anode to ions (H^+) and electrons (e^-) based on the catalyst layer action, as described in (1). Afterward, both H^+ and e^- move to the cathode through the electrolyte and the external circuit, respectively. Thereafter, to generate water and heat, the protons and electrons react together with the oxygen in the catalyst layer of the cathode, as also depicted in Figure 1 and described in (2); meanwhile, liberated electrons move through the external circuit and generate electricity. Ultimately, the overall reactions are described in (3) [45,46].

$$\text{Anode}: \ H_2 \rightarrow 2H^+ + 2e^- \tag{1}$$

$$\text{Cathode}: \ \frac{1}{2}O_2 + 2H^+ + 2e^- \rightarrow H_2O + heat \tag{2}$$

$$\text{Overall reaction}: \ 2H_2 + O_2 \rightarrow 2H_2O + Heat + Electricity \tag{3}$$

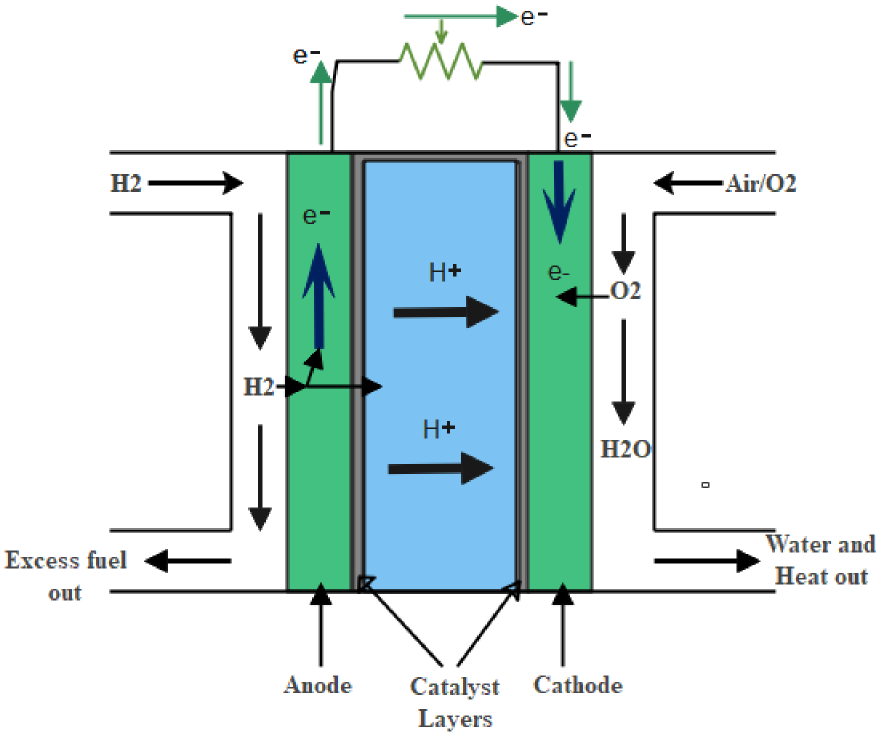


Figure 1. The schematic configuration of the PEMFC [47].

The output voltage of several fuel cells connected in series, while ignoring the irreversibility losses and entropy, is computed using the following equation according to several references [36,48–50]:

$$V_s = N_{cells}(E - v_{act} - v_R - v_c) \tag{4}$$

where N_{cells} stands for the number of fuel cells connected in series, E indicates the open-circuit voltage per cell, and estimated by the following equation:

$$E = 1.229 - 0.85 * 10^{-3}\left(T_{fc} - 298.15\right) + 4.3085 \times 10^{-5} T_{fc} ln\left(P_{H_2}\sqrt{Po_2}\right) \tag{5}$$

$$P_{H_2} = \frac{(RH_a * P_{H2O})}{2}\left[1/\left(\frac{(RH_a \times P_{H2O})}{P_a} e^{\frac{(4.192 I_{fc}/A)}{T_{fc}^{1.334}}}\right) - 1\right] \tag{6}$$

$$P_{O_2} = (RH_c * P_{H2O})\left[1/\left(\frac{(RH_a \times P_{H2O})}{P_c} e^{\frac{(4.192 I_{fc}/A)}{T_{fc}^{1.334}}}\right) - 1\right] \tag{7}$$

$$P_{H2O} = 2.95 * 10^{-2} T_c - 9.18 * 10^{-5} T_c^2 + 1.44 * 10^{-7} T_c^3 - 2.18 \,\Big|\, T_c = T_{fc} - 273.15 \tag{8}$$

v_{act}, which indicates the activation overpotential per cell, is computed by (9), and v_R is computed using (11) to determine the ohmic voltage drop in the cells, and v_c is used to compute the concentration over-potential in cells is calculated according to (13).

$$v_{act} = -\left[\xi_1 + \xi_2 T_{fc} + \xi_3 T_{fc} ln(C_{O_2}) + \xi_4 T_{fc} ln\left(I_{fc}\right)\right] \tag{9}$$

$$C_{O_2} = \frac{P_{O_2}}{5.08 \times 10^6} e^{\frac{498}{T_{fc}}} \tag{10}$$

$$V_R = I_{fc}(R_m + R_c) \, \Bigg| \, R_m = \frac{\rho_m^l}{A} \tag{11}$$

$$\rho_m = \frac{181.6\left[1+0.03\left(\frac{I_{fc}}{A}\right)+\ 0.062\left(\frac{I_{fc}}{303}\right)^2\left(\frac{I_{fc}}{A}\right)^{2.5}\right]}{\left[\lambda-0.634-3\left(\frac{I_{fc}}{A}\right)\right]e^{4.18\left(\frac{T_{fc}-303}{T_{fc}}\right)}} \tag{12}$$

$$v_c = -\beta_2 \ln\left(\frac{J_{max}-J}{J_{max}}\right) \tag{13}$$

where T_{fc} is a phrase about the operating FC temperature in Kelvin Po_2 , P_{H_2}, and P_{H2O} are the partial pressure of the oxygen (O_2), the hydrogen (H_2), and H_2O, respectively. RH_c indicates the relative moisture of vapor at the cathode (atm) and RH_a is the relative moisture of vapor at anode. P_c is the inlet pressure of the cathode, while P_a is the inlet pressure of the anode e in (atm). The concentration of the O_2 is symbolized using C_{O_2} in mol/cm^3. R_m is the membrane's resistance, while R_c is the resistance of connections. I_{fc} is the FC operating current. β_2 stands for a parametric coefficient. l (CM) and ρ_m are the thickness and the resistance of the membrane. J and J_{max} are the density of the actual current and the maximum of J (A/cm^2). It is obvious from the previously described equation that the values of seven unknown parameters (ξ_1, ξ_2, ξ_3, ξ_4, λ, R_c, and β_2) have to be accurately extracted to build up an accurate PEMFC model. This problem is known as the parameter estimation of PEMFC and belongs to the optimization problem and hence could be solved using the metaheuristic algorithms. Therefore, in this paper, an effective metaheuristic algorithm, known as GTO, will be herein adapted, with some modification on its performance, for tackling this problem.

3. The Standard Artificial Gorilla Troops Optimizer

In [51], a newly proposed metaheuristic optimizer inspired by the social intelligence of gorilla troops, namely artificial gorilla troops optimizer (GTO), has been proposed for tackling global optimization problems and their outcomes proved its superiority for those problems. The GTO algorithm is divided into two phases, exploration and exploitation, which are described next in detail.

3.1. Exploration Operator

In this phase, the GTO will seek to explore the search space of the optimization problem for finding the most promising regions, which might include the near-optimal solution as modeled in the following equations:

$$G_i(t+1) = \begin{cases} \vec{lb} + \left(\vec{ub} - \vec{lb}\right) * r & r_1 < p \\ (r_2 - C) \times (X_r(t)) + H \times L & r_1 \geq 0.5 \\ X_i(t) - L \times (L \times (X_i(t) - G_r(t)) + r_3 \times (X_i(t) - G_r(t))) & r_1 < 0.5 \end{cases} \tag{14}$$

$$C = F \times \left(1 - \frac{t}{max_t}\right) \tag{15}$$

$$F = \cos(2 \times r_4) + 1 \tag{16}$$

$$L = C \times l \tag{17}$$

$$H = \vec{Z} \times X(t) \Big| Z = \ [-C,\ C] \tag{18}$$

where $\vec{lb}$ and $\vec{ub}$ are two vectors including the lower bound and upper bound of the optimization problem dimensions, and r, r_1, r_2, r_3, and r_4 include randomly generated numbers between 0 and 1. t and max_t indicate the current iteration and maximum iteration, respectively. l is a variable including a number created randomly between -1 and 1. G_i includes the positions of the ith solution created in the next iteration, while X_i refers to the current position of the same solution. G_r is a solution selected randomly from the updated positions, while X_r also is a solution selected randomly from the current solutions. p is a predefined probability between 0 and 1. $\vec{Z}$ is a vector including random numbers generated between $-C$ and C. After that, the updated solution $G_i(t+1)$ replaces the current position $X_i(t)$ if it is better.

3.2. Exploitation Operator

The exploitation operator in the GTO is achieved based on two behaviors: following the best-so-far solution and competition for adult females. Following the best-so-far solution X^* is mathematically implemented in the GTO using the following equation:

$$G_i(t+1) = L \times M \times (X_i(t) - X^*(t)) + X_i(t) \tag{19}$$

$$M = \left(\left| \frac{1}{N} \sum_{i=0}^{N} G_i(t) \right|^g \right)^{\frac{1}{g}} \tag{20}$$

$$g = 2^L \tag{21}$$

where N represents the population size. The second behavior, known as competition for adult females, is mathematically described using the following formula:

$$G_i(t+1) = -(Q \times X_i(t) - Q \times X^*(t)) \times A + X^*(t) \tag{22}$$

$$Q = 2 \times r_5 - 1 \tag{23}$$

$$A = E \times \beta_1 \tag{24}$$

$$E = \begin{cases} N_1, & r_6 \geq 0.5 \\ N_2, & r_6 < 0.5 \end{cases} \tag{25}$$

where r_5 and r_6 are two numbers selected randomly in the range of 0 and 1. β_1 is a value predefined by the researchers before starting the optimization process. N_1 is a vector assigned randomly using the normal distribution, while N_2 is a random number also generated randomly using the normal distribution. Exchanging between those two behaviors is based on a predefined variable W and factor C presented in (15) as defined in the following equation:

$$G_i(t+1) = \begin{cases} execute\ Eq(30)\ if\ C \geq w \\ execute\ Eq(23)\ if\ C < w \end{cases} \tag{26}$$

The steps of the GTO are shown in Algorithm 1.

Algorithm 1 Pseudocode for GTO

```
Initialize $N$ solutions randomly within the search space, $X_i(i \in N)$.
Initialize $\beta_1$ , $W$, and $p$.
Evaluate each solution and determine the best-so-far one, $X^*$, with the lowest fitness value.
$t = 0$;
while ($t < max_t$)
update both $C$ and $L$ using (15) and (17), respectively.
    // Exploration
    $for$ each $i$ solution
        update the location of the current solution using (14)
        Evaluate $G_i$ and replace $X_i$ with $G_i$ if $G_i$ is better, in addition to updating $X^*$ if this updated solution is better
    End
// Exploitation
    $for$ each $i$ solution
        update the location of the current solution using (26)
        Evaluate $G_i$ and replace $X_i$ with $G_i$ if $G_i$ is better, in addition to updating $X^*$ if this updated solution is better
    End
    $t = t + 1$;
End while
Return $X^*$.
```

4. The Proposed Algorithm: MGTO

In this section, the steps of the proposed algorithm, modified GTO, will be discussed in detail. Those steps are initialization, the objective function, and the modified GTO (MGTO). Our main idea here is based on replacing an updating scheme in the standard GTO with another one, maximizing its performance toward preserving the population diversity for exploring more regions within the optimization process. This is an attempt to avoid becoming stuck in local minima while accelerating the convergence speed in the right direction of the near-optimal solution.

4.1. Initialization Step

Before starting the optimization process, a number N of solutions with seven unknown parameters will be randomly distributed within the search space of this problem using the following equation:

$$\overrightarrow{lb} + \left(\overrightarrow{ub} - \overrightarrow{lb}\right) * \overrightarrow{r} \tag{27}$$

where $\overrightarrow{r}$ is a vector of seven variables initialized randomly between 0 and 1. After that, those initialized solutions will be evaluated using the objective function discussed later and then the solution with the lowest objective value, namely the best-so-far solution, is employed in the next generation in the hope of finding a better one.

4.2. Objective Function

The MGTO is employed to find the values of seven unknown parameters based on minimizing the error value between the measured and estimated current using an objective function known as the sum of the squared errors (SSE). The SSE is mathematically formulated as shown in (25) [48]:

$$SSE = f(X_i) = \sum_{c=1}^{LD} \left| V_{S,\ measured}(c) - V_{S,\ estimated}(c) \right|^2 \tag{28}$$

where LD indicates the number of the measured data points and X_i includes the values of the parameters obtained by the ith solution.

4.3. The Modified Artificial Gorilla Troops Optimizer: MGTO

The exploitation operator still suffers from slow convergence speed because it is based significantly on using random numbers, which might maximize the step sizes. Hence, the new solution will jump far away from the best-so-far solution, which might involve the desired near-optimal solution. Furthermore, equality between exploration and exploitation processes might reduce the performance of the algorithm in some cases which need higher exploration operators and vice versa. This is the second advantage of the standard MGTO, which we seek to solve within a new modified variant called the MGTO. In the MGTO, the exploitation operator is replaced with another one, aiding in exploring more regions around the best-so-far solution and other places within the search space in the hope of finding more promising regions which might involve the desired near-optimal solution. This new exploitation operator is based on two folds, and the first one is mathematically described as that:

$$G_i(t+1) = X^* + \beta \times (X_r(t) - X_i(t)) \ + \ (1-\beta) \times (X_{r1}(t) - X_{r2}(t)) \tag{29}$$

where β is a controlling factor generated randomly between 0 and 1. In this fold, two-step sizes are generated based on the selected solution, where the first step size is based on multiplying the difference between a solution $X_r(t)$ selected randomly from the current population and the current solution $X_i(t)$ by β, and the second one is created by multiplying the difference between two solutions, $X_{r1}(t)$ and $X_{r2}(t)$, selected randomly from the current population by $(1-\beta)$. Finally, the best-so-far solution will be updated using those two steps. This phase will relate searching around the nearly best-so-far solution with the diversity of the population, for example, if the difference between the two solutions selected randomly from the population is small, the generated step sizes will update the current solution to a region so near to the best-so-far solution; otherwise, the current solution will update to a far region from the best-so-far solution. Hence, this will help in avoiding becoming stuck in local minima with aiding to explore all regions gradually around the best-so-far solution. The second fold is modeled using the following equation:

$$G_i(t+1) \ = X_r(t) + \ a \times (X_{r1}(t) - X_i(t)) + \ b \times (X_{r2}(t) - X_{r3}(t)) \tag{30}$$

where $X_r(t)$ is a solution selected randomly from the current population, a and b are two random numbers selected between 0 and 1. Moreover, in this fold, two steps are generated using two different random numbers to generate various steps sizes exploring the regions extensively around the randomly selected solution as a new attempt to avoid becoming stuck in local minima, in addition to preserving the population diversity as much as possible. Exchanging between those two folds is achieved with a probability of 0.5, as shown in (31).

$$G_i(t+1) = \begin{cases} execute\ Eq(29)\ if\ C \leq 0.5 \\ execute\ Eq(30)\ if\ C > 0.5 \end{cases} \tag{31}$$

Finally, the steps of the proposed algorithm, MGTO, are presented in Algorithm 2 and depicted in Figure 2.

Algorithm 2 Pseudocode for modified GTO (MGTO)

```
Initialize $N$ solutions randomly within the search space, $X_i(i \in N)$.
Initialize $\beta_1$ , $W$, and $p$.
Evaluate each solution and determine the best-so-far one, $X^*$, with the lowest fitness value.
$t$ = 0;
while ($t < max_t$)
    update both $C$ and $L$ using (15) and (17), respectively.
    // Exploration $for$ each $i$ solution
        update the location of the current solution using (14)
        Evaluate $G_i$ and replace $X_i$ with $G_i$ if $G_i$ is better, in addition to updating $X^*$ if this updated solution is better
    End
// Modified exploitation
    for each $i$ solution
        update the location of the current solution using (31)
        Evaluate $G_i$ and replace $X_i$ with $G_i$ if $G_i$ is better, in addition to updating $X^*$ if this updated solution is better
     End
    $t = t + 1$;
   End while
Return $X^*$.
```

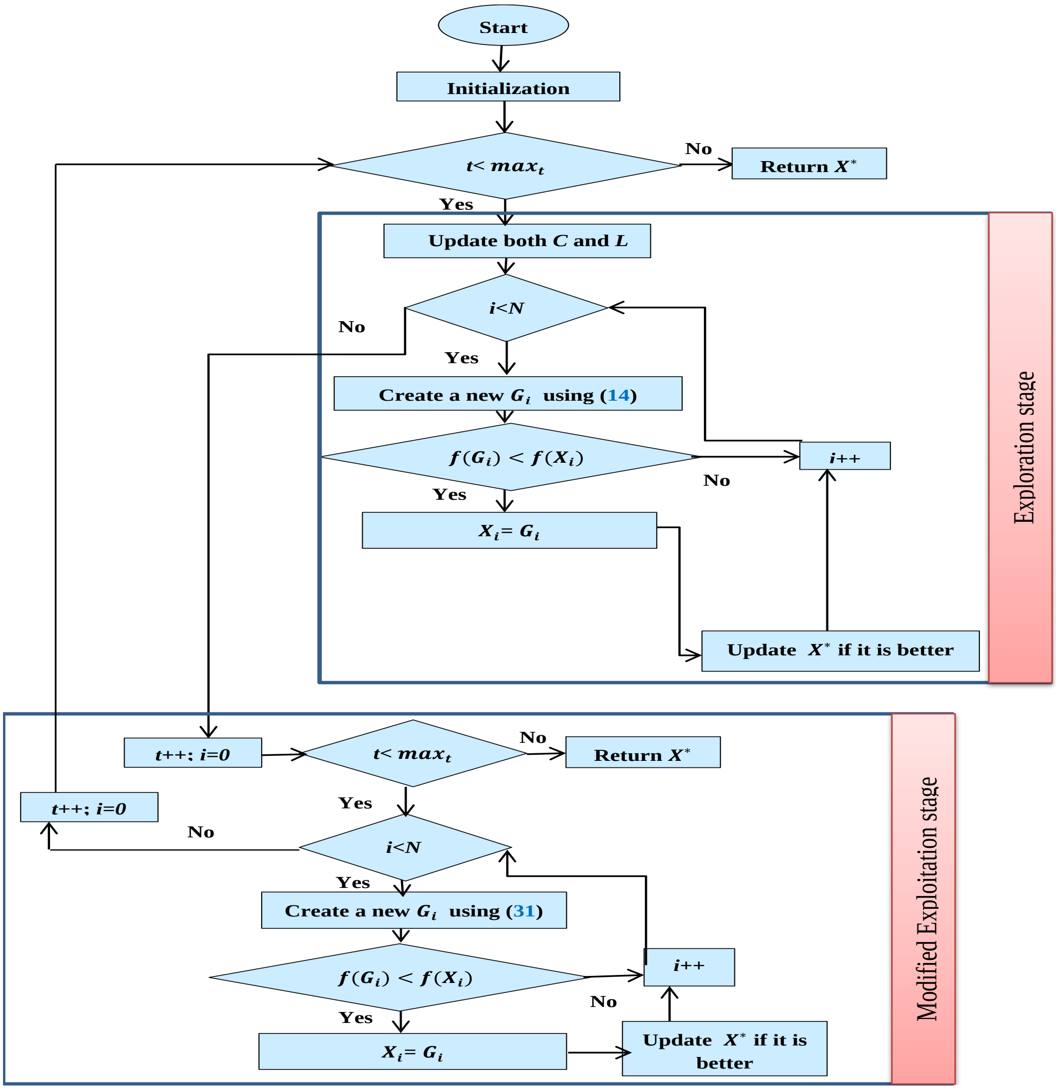


Figure 2. Flowchart of MGTO algorithm.

5. Findings and Discussions

Our proposed algorithm is validated using four commercial PEMFC stacks, 250 W stack, BCS-500W stack, and SR-12 stack, due to widespread use in the literature [9,34,45,52–58]. Those PEMFC stacks have I–V curves specified in the manufacturers' datasheets and have to be estimated by finding the unknown parameters of its mathematical model. Each unknown parameter has a search boundary that contains the near-optimal solution in an unknown region within. Generally, the characteristics of the employed PEMFCs stack, and the lower and upper bound of the unknown parameters, are presented in Table 2. Furthermore, the MGTO is compared with nine well-known optimizers to show its efficiency as a strong alternative to tackle the parameter estimation of PEMFC stacks; those algorithms are differential evolution (DE) [59], grey wolf optimizer (GWO) [59], hybrid DE with GWO (DEGWO) [59], bonobo optimizer (BO) [60], flower pollination algorithm (FPA) [61], slime mold algorithm (SMA) [16], seagull optimization algorithm [62], horse herd optimization algorithm [63], and classical GTO [51]. Regarding the parameters of those algorithms, they are set as found in the cited paper, except classical GTO, such that its parameters assignment will be discussed within the next section.

Table 2. Lb and ub of each unknown parameter in addition to the employed PEMFC characteristics.

PEMFC Type	250 W Stack	BCS-500W Stack	SR-12 Stack	Parameters	ub	Lb
N_{cells}	24	32	48	ξ_1	−1.1997	−0.8532
A (cm^2)	27	64	62.5	$\xi_2 \times 10^{-3}$	1.0000	5.0000
l (μm)	178	178	25	$\xi_3 \times 10^{-5}$	3.6000	9.8000
J_{max}, mA/cm^2	860	469	672	$\xi_4 \times 10^{-5}$	−26.0000	−9.5400
T_{fc}, K	338.15	333	323	λ	13.0000	23.0000
P_{O_2}, atm	1	1	0.2095	$R_c \times 10^{-4}$	1.0000	8.0000
P_{H_2}, atm	1	1	1.47628	$\beta \times 10^{-2}$	1.3600	50.0000

All experiments conducted herein are implemented using the MATLAB platform, using a device with the capabilities:

- 32 GB of RAM;
- Intel(R) Core(TM) i7-4700MQ CPU @ 2.40 GHz;
- 64-bit Windows 10.

Finally, there are two well-known metrics used to evaluate the accuracy of the obtained parameters for minimizing the error between the measured and estimated data; those metrics are mean absolute percentage error (MAPE) and mean absolute error (MAE), which are mathematically described using the following equation:

$$\text{MAE} = \frac{1}{LD}\sum_{c=1}^{LD}\left|v_{fc,\ measured}(c) - v_{fc,\ estimated}(c)\right| \tag{32}$$

$$\text{MAPE} = \frac{1}{LD}\sum_{c=1}^{LD}\left|\frac{v_{fc,\ measured}(c) - v_{fc,\ estimated}(c)}{v_{fc,\ measured}(c)}\right| \tag{33}$$

5.1. Parameter Settings

The classical GTO has three parameters, p, W, and β, that have to be accurately picked to maximize its performance. Therefore, extensive experiments have been conducted using various values for each parameter. For example, the best value for both p and W has been picked after conducting extensive experiments with various values of 0.0, 0.01, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.99, and 1.0, which show that the best value for p is of 0.01, as depicted in Figure 3a, while the best for W is competitive for 0.6, 0.8, and 0.99. Generally, values of 0.01 and 0.8 are used within the conducted experiments next for p and W, respectively. Likewise, for the parameter β, several experiments have been conducted under virus values, and their outcomes have been depicted in Figure 3c, which shows that a value of 2 for this parameter is better than all the other values.

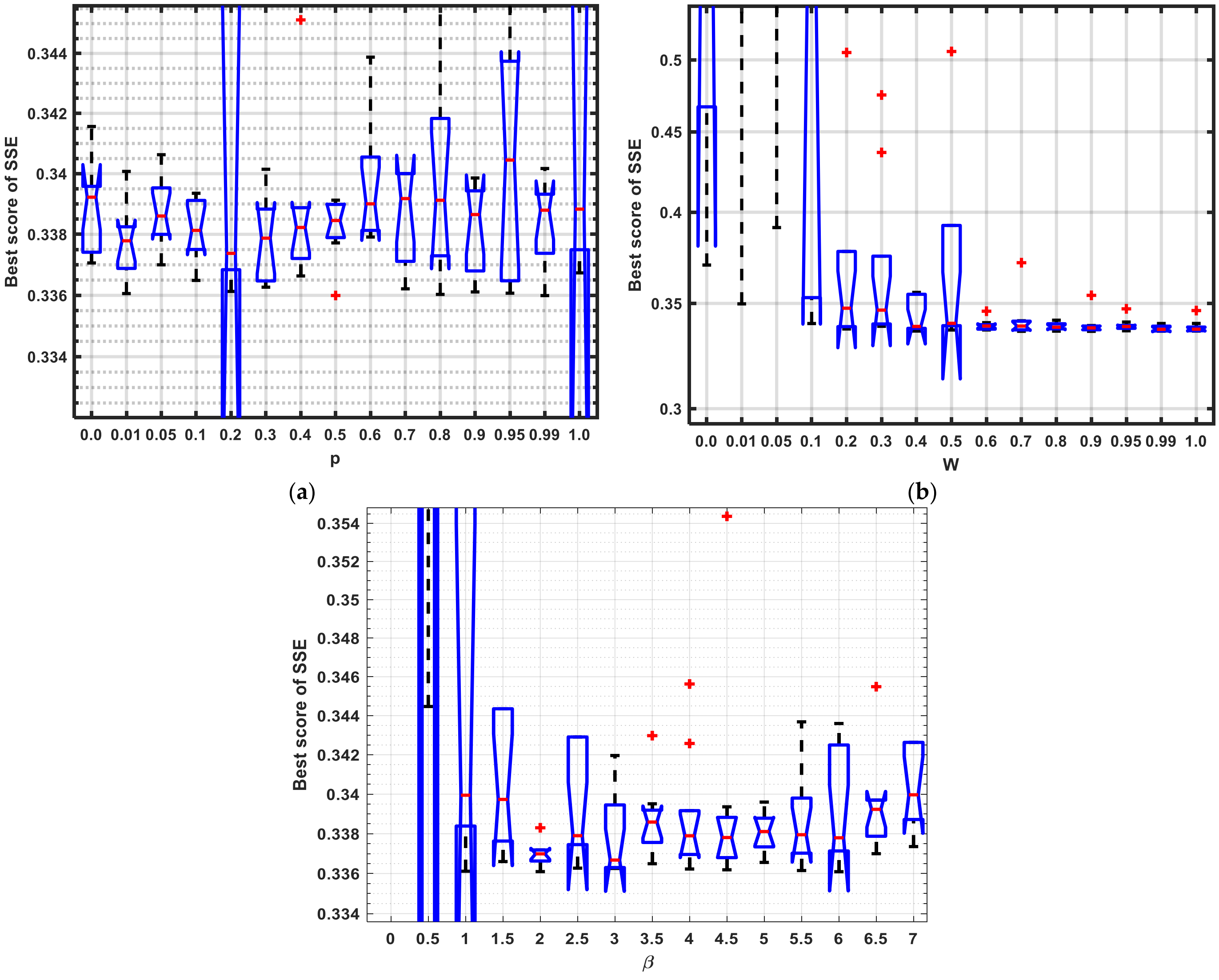


Figure 3. Tuning the parameters of GTO: (**a**) tuning the parameter *p*, (**b**) tuning the parameter *W*, (**c**) tuning the parameter *β*.

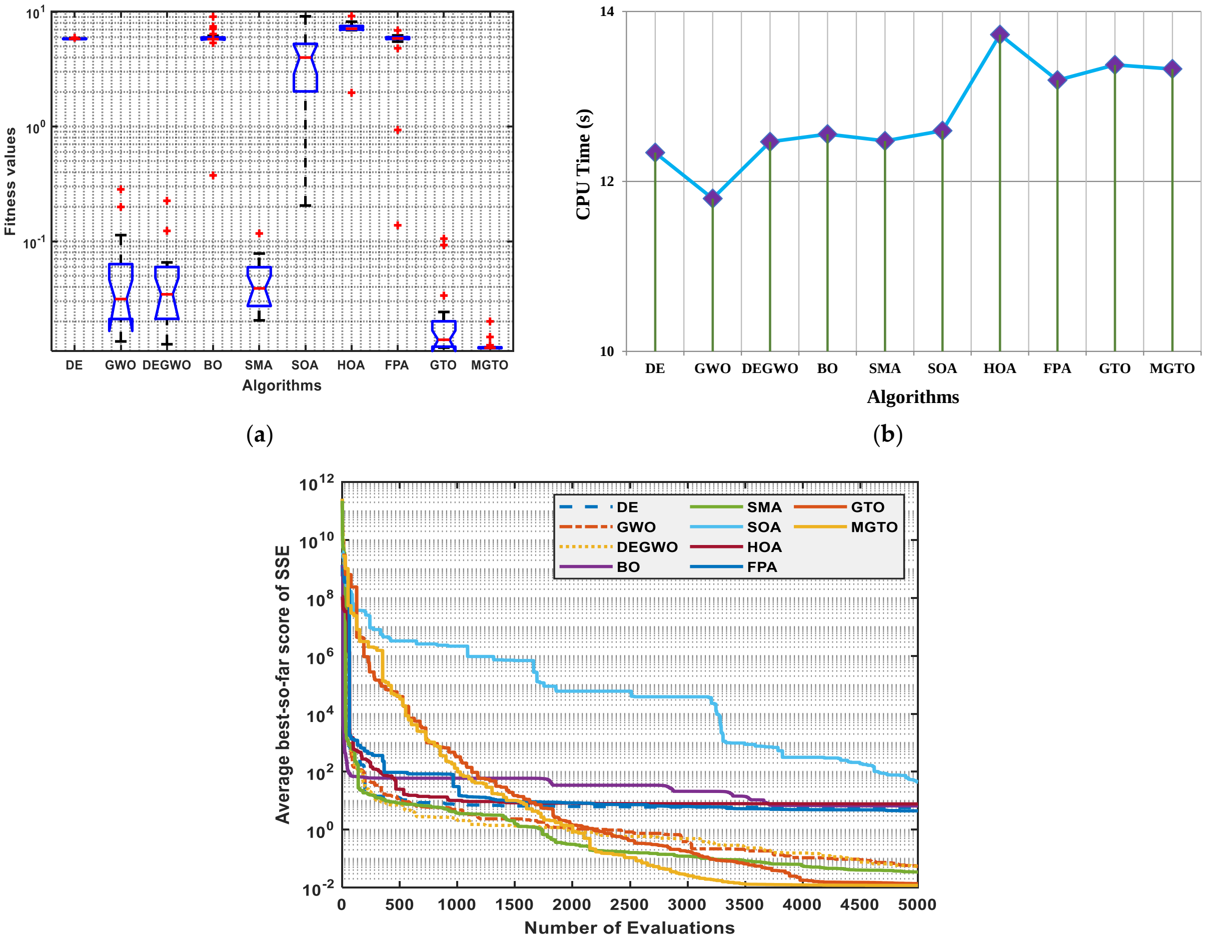
5.2. Test Case 1: 500 W Stack

This section investigates the performance of our proposed algorithm, MGTO, using the first test case based on using a well-known commercial PEMFC stack called 500 W stack. At the onset, all algorithms have been prepared using the same population size and maximum of function evaluation of 25 and 5000 to achieve a fair comparison, and then all are executed 25 independent runs. The outcomes of those runs have been analyzed in terms of the best, average, worst, and standard deviation (SD), which are presented in Table 3. In addition, this table presents the best-obtained parameters by each algorithm. As a result of observation, the MGTO could be the best for all those terms and the GTO comes in the second rank as the second-best one after the MGTO, while the SOA is the worst. Moreover, this table shows that the MGTO is better for the two additional metrics: MAE and MAPE.

Table 3. Comparison among algorithms using test case 1.

Parameter	DE	GWO	DEGWO	BO	SMA	SOA	HOA	FPA	GTO	MGTO
ξ_1(V)	−0.9677	−0.9978	−0.9328	−0.8748	−0.8548	−0.9488	−0.8532	−1.0586	−0.9082	−1.0305
ξ_2(V/k)	0.0037	0.0038	0.0027	0.0025	0.0033	0.0026	0.0022	0.0031	0.0032	0.0029
ξ_3(V/k)	9.800×10^{-5}	9.437×10^{-5}	4.565×10^{-5}	5.348×10^{-5}	9.026×10^{-5}	3.706×10^{-5}	3.600×10^{-5}	5.350×10^{-5}	7.862×10^{-5}	3.900×10^{-5}
ξ_4(V/k)	-9.54×10^{-5}	-1.80×10^{-4}	-1.82×10^{-4}	-9.54×10^{-5}	-1.776×10^{-4}	-1.012×10^{-4}	-9.540×10^{-5}	-9.540×10^{-5}	-1.910×10^{-4}	-1.931×10^{-4}
λ	13.0000	22.1506	21.9455	13.0000	15.6367	16.3142	13.0000	13.4848	22.9584	21.1529
$R_c(\Omega)$	8.000×10^{-4}	5.076×10^{-4}	6.778×10^{-4}	8.000×10^{-4}	1.977×10^{-4}	3.663×10^{-4}	1.908×10^{-4}	6.129×10^{-4}	4.245×10^{-4}	1.199×10^{-4}
β_2	0.0238	0.0192	0.0157	0.0177	0.0136	0.0730	0.0309	0.0264	0.0155	0.0162
SSE (best)	5.7768	0.0176	0.0146	0.0117	0.0171	0.1975	6.7438	0.3019	0.0117	0.0117
SSE (avg)	5.8192	0.0794	0.0373	5.4968	0.0325	17.0048	7.1814	5.8182	0.0169	0.0120
SSE (worst)	5.9802	0.2871	0.1470	14.2898	0.1272	87.9465	8.0204	6.9522	0.0336	0.0162
SSE (SD)	0.0517	0.0813	0.0276	2.6105	0.0251	19.7260	0.2916	1.3542	0.0062	0.0009
MAE (best)	0.4316	0.0850	0.0505	0.3388	0.0614	1.1226	0.4780	0.4480	0.0141	0.0128
MAPE (best)	1.9153	0.4111	0.2328	1.3925	0.2760	5.6113	2.1577	1.9928	0.0647	0.0610

To measure the convergence speed, five-number summary, and CPU time of the MGTO, Figure 4 is presented to expose all those outcomes for each algorithm. From this figure, it is observed that the MGTO is the best in terms of faster reaching the near-optimal solution and the five-number summary depicted in the boxplot. Broadly speaking, Figure 4a, which depicts the outcomes of various algorithms on this test case using the boxplot, shows that the proposed algorithm is the best, and the GTO is the second-best one, while the SOA is the worst one. Regarding the convergence speed shown in Figure 4c, the MGTO could come true with the best convergence speed, and the GTO is the second-best one, while the FPA is the worst one. For CPU time depicted in Figure 4b, the MGTO could come as the eighth one after the GWO, DEGWO, DE, FPA, SMA, SOA, and BO, but its superiority for the other metrics, apart from converging the CPU time with the best eight algorithms, makes it the best for tackling this problem.



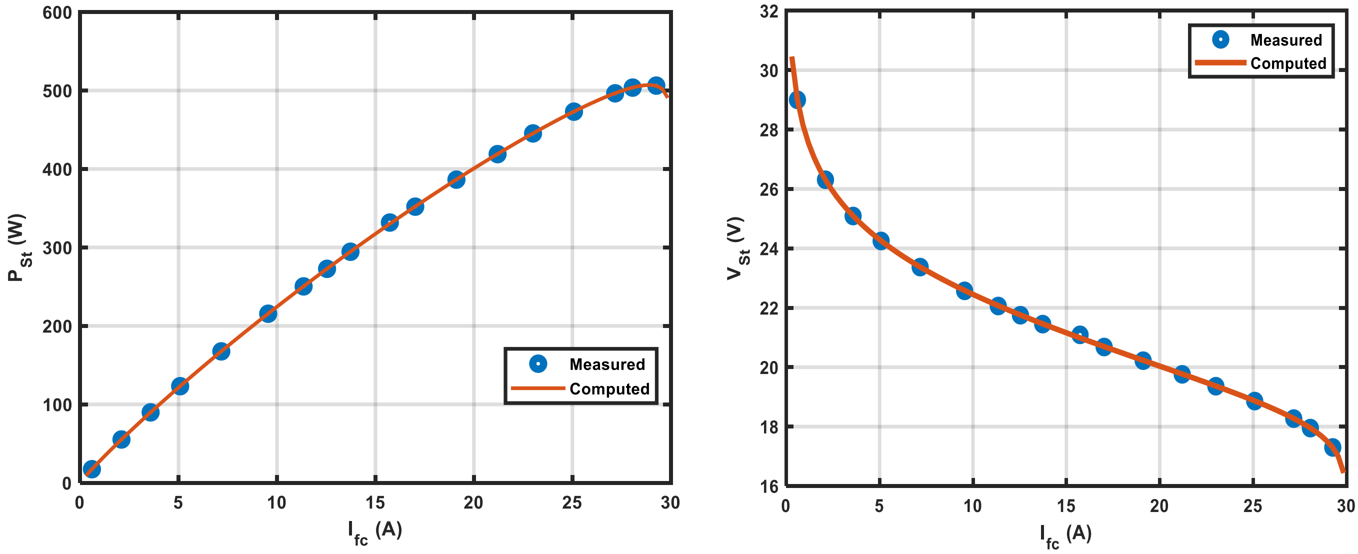
(**a**)

(**b**)

(**c**)

Figure 4. Comparison among algorithms in terms of five-number summary (boxplot), convergence speed, and CPU time on test case 1: (**a**) boxplot of the obtained outcomes, (**b**) CPU time, (**c**) convergence curve.

Furthermore, Figure 5a,b depicts the I–V and I–P polarization curves obtained by depicting the estimated and measured data; the best-so-far parameters estimated by the MGTO receive the estimated data points. From this figure, the MGTO could significantly find accurate parameters that minimize the error between measured and estimated data.



(**a**) (**b**)

Figure 5. Depiction of I–V and I–P curves obtained using MGTO on test case 1: (**a**) I–P measured against I–P computed, (**b**) I–V measured against I–V computed.

In Table 4, the *p*-value under the Wilcoxon rank-sum test, which determines if the outcomes obtained by the MGTO on test case 1 are significantly different from those of each rival algorithms, are shown. This table shows that all *p*-values are less than 0.05, making the acceptance moves toward the alternative hypothesis, which says that there is a significant difference between the outcomes obtained by the MGTO and each of the others.

Table 4. Comparison under Wilcoxon rank-sum test on test case 1.

	DE	GWO	DEGWO	BO	SMA	SOA	HOA	FPA	GTO
h-value	1	1	1	1	1	1	1	1	1
p-value	1.4156×10^{-9}	1.4156×10^{-9}	2.2854×10^{-9}	2.5260×10^{-8}	1.4156×10^{-9}	1.4156×10^{-9}	1.4156×10^{-9}	1.4156×10^{-9}	8.8609×10^{-6}

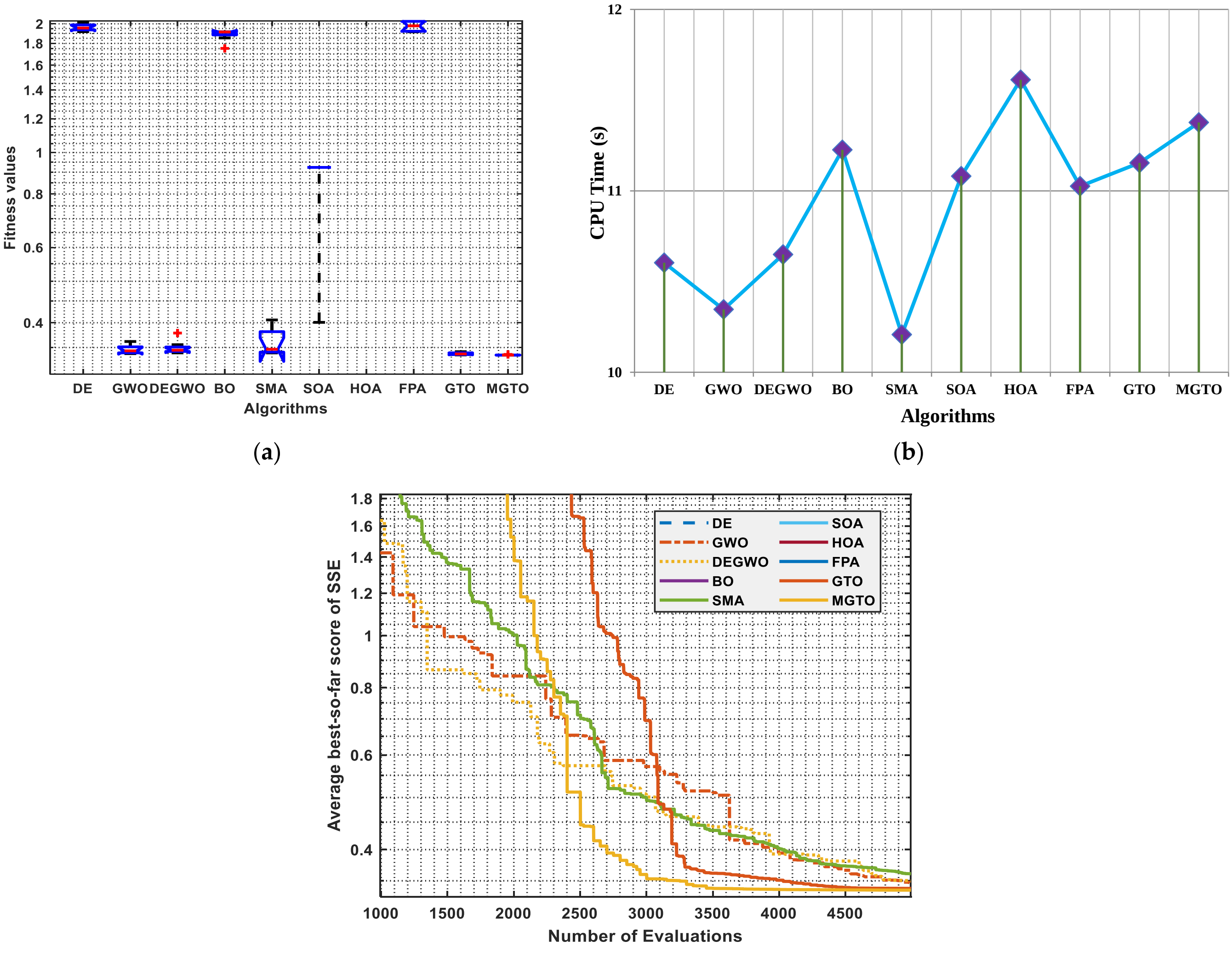
5.3. Test Case 2: 250 W Stack

Another well-known commercial PEMFC stack, called 250 W stack, is used to investigate the performance of the MGTO compared with some of the rival algorithms. The outcomes obtained by running the MGTO and the other rival algorithms 30 independent times on this stack are given in Table 5. This table shows that the MGTO is the best for all employed metrics, except MAE and MAPE, which are better for GTO.

In addition, the boxplot, the convergence speed, and CPU time of each algorithm are presented in Figure 6, which shows the superiority of MGTO in terms of the five-number summary depicted using the boxplot, and the convergence speed, while their performance is competitive for CPU time. Broadly explaining, Figure 6a shows that the proposed algorithm is the best and GTO is the second-best, while SOA is the worst. Additionally, Figure 7 is presented to show I–V and I–P curves between measured and estimated data points. From this figure, it is obvious that estimated data points are highly consistent with those obtained practically.

Table 5. Comparison among algorithms on test case 2.

Parameter	DE	GWO	DEGWO	BO	SMA	SOA	HOA	FPA	GTO	MGTO
$\xi_1(\mathrm{V})$	−0.9126	−1.1459	−1.1997	−1.1749	−0.8712	−1.0017	−0.8532	−1.1678	−1.0071	−0.9184
$\xi_2(\mathrm{V/k})$	0.0030	0.0033	0.0044	0.0037	0.0033	0.0034	0.0023	0.0038	0.0035	0.0029
$\xi_3(\mathrm{V/k})$	6.826×10^{-5}	3.842×10^{-5}	9.378×10^{-5}	6.015×10^{-5}	8.822×10^{-5}	7.339×10^{-5}	3.600×10^{-5}	7.012×10^{-5}	7.518×10^{-5}	5.342×10^{-5}
$\xi_4(\mathrm{V/k})$	-9.540×10^{-5}	-1.627×10^{-4}	-1.700×10^{-4}	-9.540×10^{-5}	-1.682×10^{-4}	-1.021×10^{-4}	-9.540×10^{-5}	-9.540×10^{-5}	-1.746×10^{-4}	-1.741×10^{-4}
λ	14.0436	20.6210	22.7543	14.1055	18.8055	19.7886	13.0000	13.5684	19.5978	20.4346
$R_c(\Omega)$	8.000×10^{-4}	4.526×10^{-4}	2.872×10^{-4}	8.000×10^{-4}	1.115×10^{-4}	5.990×10^{-4}	1.000×10^{-4}	2.947×10^{-4}	1.006×10^{-4}	2.625×10^{-4}
β_2	0.0136	0.0176	0.0196	0.0136	0.0136	0.0294	0.0136	0.0140	0.0139	0.0146
SSE (best)	1.9144	0.3407	0.3369	0.3360	0.3362	0.3508	2.0445	0.9151	0.3361	0.3360
SSE (avg)	1.9577	0.3934	0.3489	2.1610	0.3442	6.2246	2.0501	1.9129	0.3404	0.3364
SSE (worst)	2.0445	1.2163	0.4271	6.2395	0.3791	30.7484	2.0653	2.0451	0.3740	0.3383
SSE (SD)	0.0408	0.1730	0.0211	1.1183	0.0105	8.4842	0.0064	0.2352	0.0079	0.0007
MAE (best)	0.2759	0.1449	0.1509	0.2739	0.1299	0.3095	0.2899	0.2805	0.1252	0.1263
MAPE (best)	1.4663	0.8079	0.8521	1.4520	0.7061	1.7270	1.5555	1.4887	0.6891	0.6934



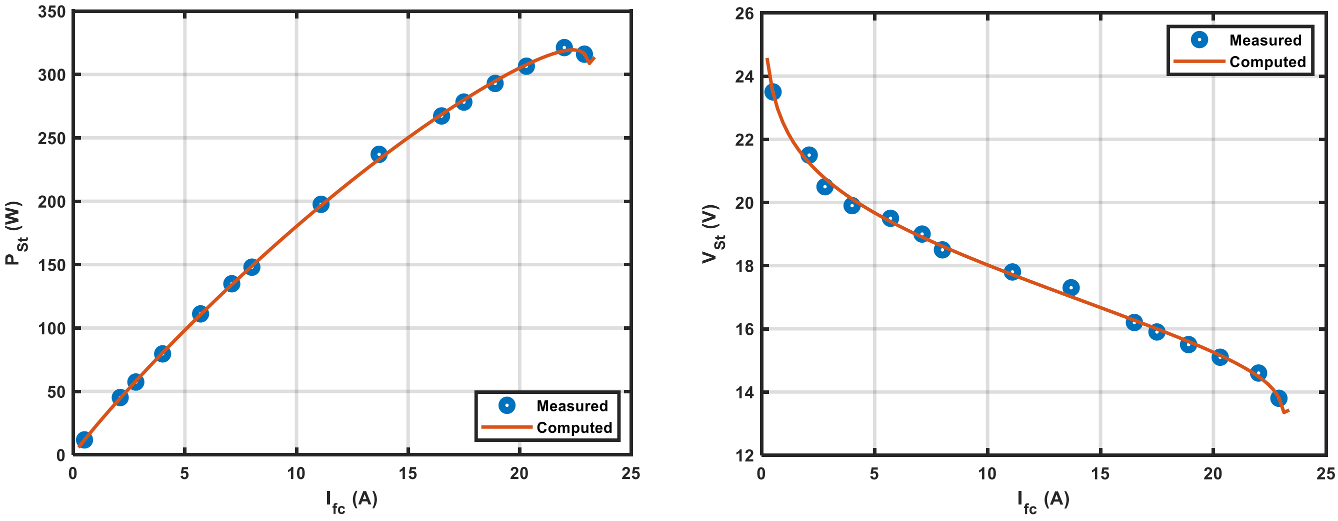
(**a**)

(**b**)

(**c**)

Figure 6. Comparison among algorithms in terms of five-number summary (boxplot), convergence speed, and CPU time on test case 2: (**a**) boxplot of the obtained outcomes, (**b**) CPU time, (**c**) convergence curve.

Finally, the Wilcoxon rank-sum test is used to see if the outcomes of the MGTO are different from the rival algorithms or not. The outcome results from applying the Wilcoxon rank-sum test are presented in Table 6, which shows that the alternative hypothesis is ace with all the rival algorithms because the *p*-value of each algorithm is less than 5%.



(**a**) (**b**)

Figure 7. Depiction of I–V and I–P curves obtained using MGTO on test case 2: (**a**) I–P measured against I–P computed; (**b**) I–V measured against I–V computed.

Table 6. Comparison under Wilcoxon rank-sum test on test case 2.

	DE	GWO	DEGWO	BO	SMA	SOA	HOA	FPA	GTO
h-value	1	1	1	1	1	1	1	1	1
p-value	1.4157×10^{-9}	1.4157×10^{-9}	4.1259×10^{-9}	4.4532×10^{-8}	3.9965×10^{-8}	1.4157×10^{-9}	1.4157×10^{-9}	1.4157×10^{-9}	1.6165×10^{-7}

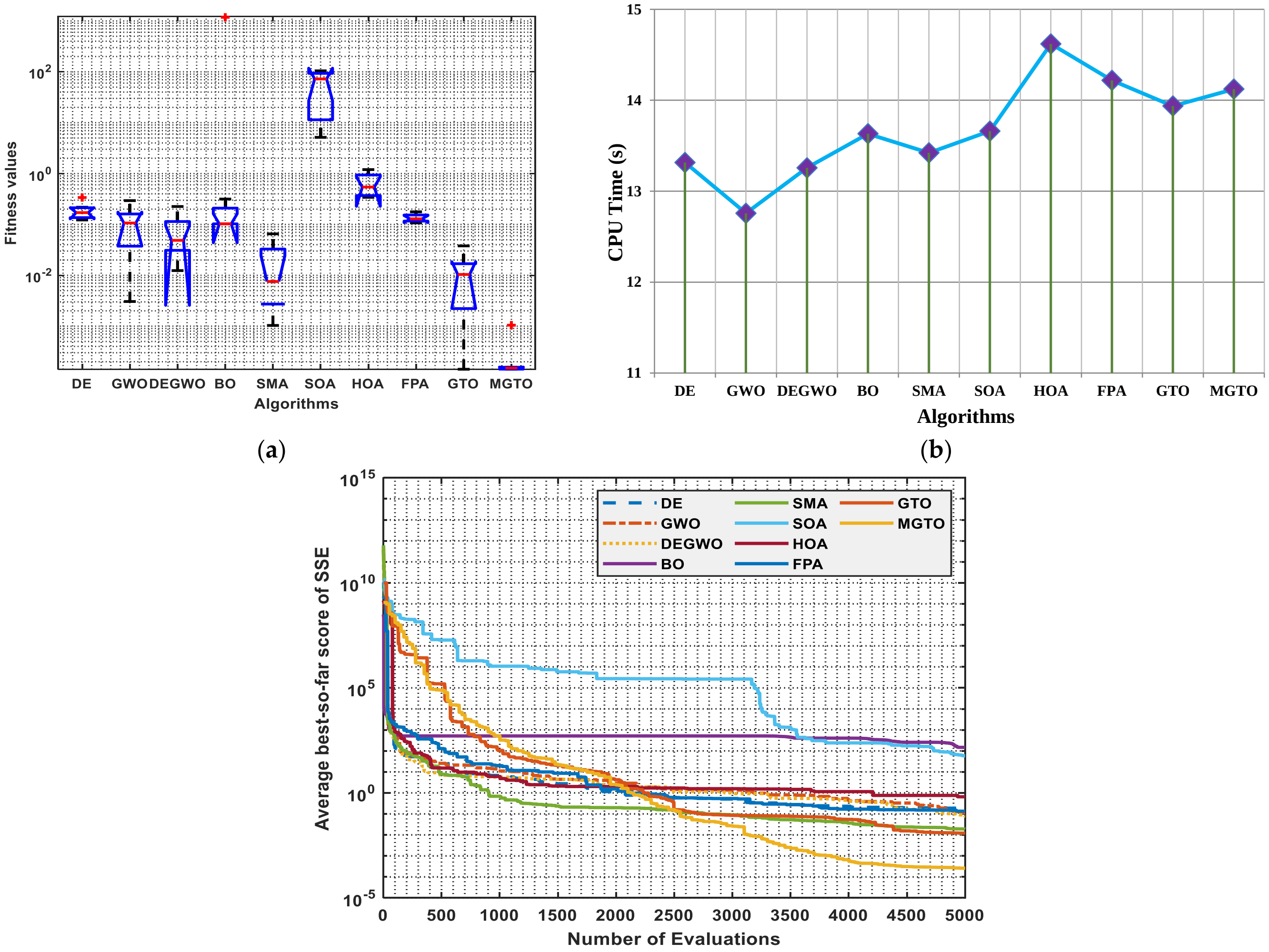
5.4. Test Case 3: SR-12 PEMFC Stack

In this section, an additional commercial PEMFC stack, namely SR-12 500 W, utilized widely in the literature, is used to affirm the effectiveness of our proposed algorithm; the proposed and rival algorithms have been executed for 30 independent runs under the same settings and their analyzed outcomes are given in Table 7. This table (Table 7) shows that the proposed algorithm, MGTO, is the best in terms of the best, average, worst, SD, MAE, and MAPE, while the SOA is the worst for all those metrics. In addition, Figure 8 has been presented to show the performance of the proposed algorithm graphically compared to the others in terms of CPU time, convergence speed, and five-number summary. It is concluded from this figure that the MGTO is the best for the five-number summary, shown in Figure 8a, and the convergence speed, displayed in Figure 8c, while its CPU time is competitive with the others, as depicted in Figure 8b. Furthermore, Table 8 presents the outcomes resulting from applying the Wilcoxon rank-sum test on the outcomes obtained by the proposed algorithm against those of each one of the rival algorithms. According to the outcomes presented in this table, the MGTO's outcomes are significantly different from those obtained by the others because the *p*-value under each rival algorithm is less than 0.05. This makes the alternative hypothesis in the Wilcoxon rank-sum test accepted.

Figure 9 is presented to show the consistency of the estimated I–V and I–P curves against the measured ones, which affirms that the estimated parameters by the MGTO could reach estimated characteristics that are highly consistent with the measured ones and hence it is a strong alternative to all the existing parameter estimation techniques.

Table 7. Comparison among algorithms using test case 3.

Parameter	DE	GWO	DEGWO	BO	SMA	SOA	HOA	FPA	GTO	MGTO
$\xi_1(\mathrm{V})$	−0.9491	−0.8568	−1.0377	−0.8532	−1.0505	−0.9463	−0.8532	−0.8577	−1.0774	−1.1403
$\xi_2(\mathrm{V/k})$	0.0027	0.0031	0.0033	0.0029	0.0030	0.0030	0.0023	0.0027	0.0037	0.0040
$\xi_3(\mathrm{V/k})$	3.9×10^{-5}	7.51×10^{-5}	5.71×10^{-5}	6.35×10^{-5}	3.61×10^{-5}	4.97×10^{-5}	3.60×10^{-5}	5.59×10^{-5}	7.24×10^{-5}	7.43×10^{-5}
$\xi_4(\mathrm{V/k})$	-9.00×10^{-5}	-1.02×10^{-4}	-1.00×10^{-4}	-9.54×10^{-5}	-1.01×10^{-4}	-1.87×10^{-4}	-9.54×10^{-5}	-9.54×10^{-5}	-1.06×10^{-4}	-1.06×10^{-4}
λ	17.5081	14.7370	13.0595	14.7560	21.0273	14.8101	13.0000	13.9576	22.9814	18.9405
$R_c(\Omega)$	4.6×10^{-4}	5.05×10^{-4}	1.10×10^{-4}	8.00×10^{-4}	7.36×10^{-4}	3.85×10^{-4}	2.05×10^{-4}	7.97×10^{-4}	3.63×10^{-4}	2.33×10^{-4}
β_2	0.1483	0.1440	0.1512	0.1425	0.1439	0.0984	0.1548	0.1418	0.1491	0.1498
SSE (best)	0.1048	0.0032	0.0028	0.0833	0.0004	0.0282	0.3003	0.0250	1.4441×10^{-4}	1.4212×10^{-4}
SSE (avg)	0.2349	0.0497	0.0517	10.2931	0.0229	20.8772	0.7558	0.1313	5.948×10^{-3}	1.6760×10^{-4}
SSE (worst)	0.6134	0.1623	0.2103	254.4047	0.0817	126.4416	2.4341	0.2059	3.0422×10^{-2}	6.4372×10^{-4}
SSE (SD)	0.1210	0.0463	0.0536	50.8566	0.0236	27.2663	0.5015	0.0413	9.3330×10^{-3}	9.9469×10^{-5}
MAE (best)	0.0819	0.0310	0.0565	0.0494	0.0336	0.8413	0.1166	0.0503	5.8197×10^{-3}	2.2472×10^{-3}
MAPE (best)	0.2382	0.1012	0.1656	0.1432	0.1064	2.6281	0.3885	0.1474	1.8892×10^{-2}	7.0079×10^{-3}



(**a**) (**b**)

(**c**)

Figure 8. Comparison among algorithms in terms of five-number summary (boxplot), convergence speed, and CPU time on test case 3: (**a**) boxplot of the obtained outcomes, (**b**) CPU time, (**c**) convergence curve.

Table 8. Comparison under Wilcoxon rank-sum test using test case 3.

	DE	GWO	DEGWO	BO	SMA	SOA	HOA	FPA	GTO
h-value	1	1	1	1	1	1	1	1	1
p-value	1.417×10^{-9}	1.417×10^{-9}	1.417×10^{-9}	1.417×10^{-9}	1.802×10^{-9}	1.417×10^{-9}	1.417×10^{-9}	1.417×10^{-9}	2.297×10^{-8}

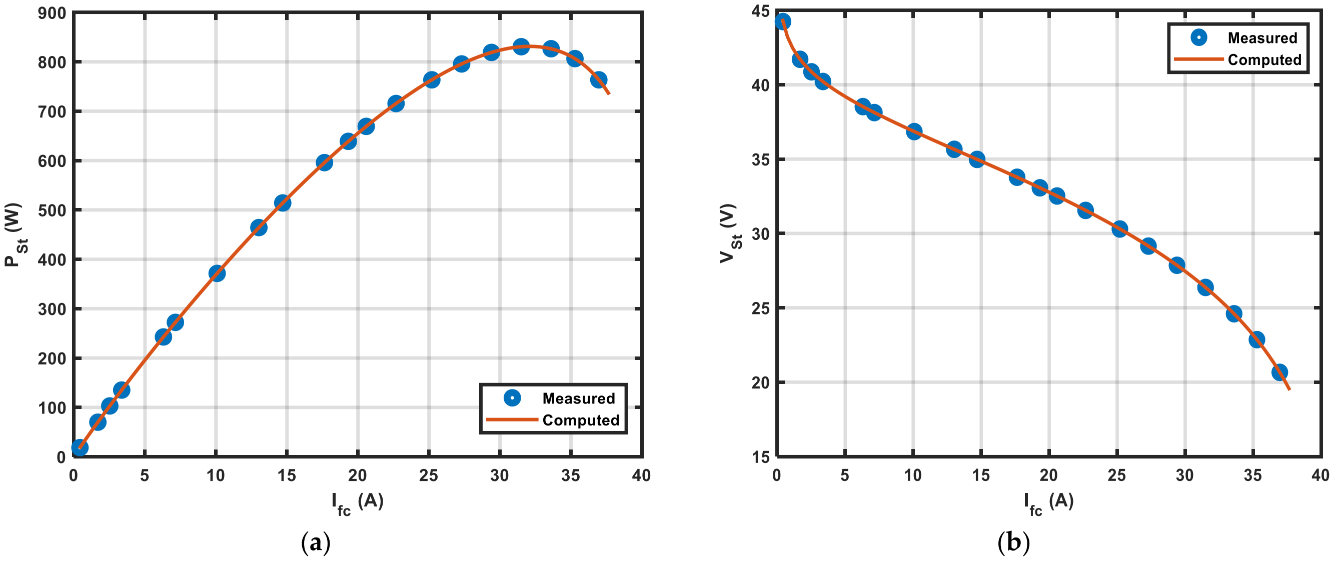


Figure 9. Depiction of I–V and I–P curves obtained using MGTO on test case 3: (**a**) I–P measured against I–P computed, (**b**) I–V measured against I–V computed.

5.5. Accumulative Grade Point Assessment

In [64], a new assessment mechanism, known as an accumulative grade point assessment (CGPA), has been proposed to rank and evaluate the performance of the various algorithms based on six factors:

- Absolute Error (E);
- Computational Time (t);
- Standard Deviation of Error (σ_e);
- Standard Deviation of Time (σ_e);
- Consistency of Rs (λRs);
- Consistency of Rsh (λRsh).

Herein, this mechanism is employed to evaluate the performance of the proposed algorithms relative to the others. However, only the first four factors are employed because it is corresponding to the current problem, parameter estimation of the PEMFC, while the other two factors correspond to the parameter estimation of the photovoltaic model. This mechanism calculates the grade point assessment (GPA) for each factor from those four factors using the following formula:

$$\text{GPA} = w * \frac{(\beta_{max} - \beta)}{(\beta_{max} - \beta_{min})} \tag{34}$$

where β indicates the obtained value by an algorithm for an arbitrary factor, w is the weight of this factor in proportion to the others, β_{max} and β_{min} are the maximum and minimum values obtained by the algorithms for this factor. For the first factor, the GPA weightage was set to 2 because the absolute error is considered the most important factor used to measure the efficiency of the algorithms for tackling this problem, while the other factors were set to 0.025. Finally, the total GPA (T-GPA) is a phrase about the average GPA values obtained on the four factors. In Tables 9–11, the total GPA for those four factors on three investigated test cases are presented. After observing those tables, it is concluded that the MGTO could reach the best T-GPA compared to the others although the proposed could not achieve the best value for the computational time for those test cases. Finally, the accumulative GPA values (CGPA), which result from calculating the average of the T-GPA values obtained by each algorithm on three investigated test cases, are introduced in Table 12, which affirms the superiority of the MGTO.

Table 9. Comparison under T-GPA on test case 1.

	DE	GWO	DEGWO	BO	SMA	SOA	HOA	FPA	GTO	MGTO
E	0.19180	1.82081	1.91974	1.47338	1.79757	0.59519	0.00000	0.28365	1.98509	2.00000
t	0.01801	0.02500	0.01635	0.01518	0.01621	0.01466	0.00000	0.00696	0.00462	0.00525
σ_e	0.00242	0.02452	0.02485	0.02195	0.02432	0.00995	0.00000	0.00116	0.02500	0.02497
σ_t	0.01784	0.02410	0.02500	0.02217	0.02194	0.02468	0.02420	0.02329	0.00000	0.01944
T-GPA	0.05752	0.47361	0.49649	0.38317	0.46501	0.16112	0.00605	0.07877	0.50368	0.51242

Table 10. Comparison under GPA on test case 2.

	DE	GWO	DEGWO	BO	SMA	SOA	HOA	FPA	GTO	MGTO
E	0.01989	1.74097	1.85527	0.21060	1.99434	0.39208	0.00000	0.29705	1.99173	2.00000
t	0.01795	0.02253	0.01715	0.00687	0.02500	0.00946	0.00000	0.01044	0.00815	0.00420
σ_e	0.00323	0.02500	0.02426	0.00309	0.02247	0.01086	0.00343	0.00000	0.02246	0.02248
σ_t	0.02772	0.00075	0.01202	0.02053	0.02371	0.01691	0.00000	0.02500	0.01976	0.01758
T-GPA	0.01720	0.44731	0.47717	0.06027	0.51638	0.10733	0.00086	0.08312	0.51052	0.51107

Table 11. Comparison under GPA on test case 3.

	DE	GWO	DEGWO	BO	SMA	SOA	HOA	FPA	GTO	MGTO
E	1.6956	1.8771	1.8282	1.8844	1.9349	0.0000	1.7813	1.8816	1.9982	2.0000
t	0.0175	0.0250	0.0183	0.0133	0.0161	0.0129	0.0000	0.0054	0.0092	0.0067
σ_e	0.0000	0.0227	0.0070	0.0165	0.0209	0.0189	0.0047	0.0015	0.0110	0.0250
σ_t	0.0000	0.0227	0.0070	0.0165	0.0209	0.0189	0.0047	0.0015	0.0110	0.0250
T-GPA	0.4283	0.4869	0.4651	0.4827	0.4982	0.0126	0.4477	0.4725	0.5073	0.5142

Table 12. Comparison under CGPA on three test cases.

	DE	GWO	DEGWO	BO	SMA	SOA	HOA	FPA	GTO	MGTO
T-GPA1	0.05752	0.47361	0.49649	0.38317	0.46501	0.16112	0.00605	0.07877	0.50368	0.51242
T-GPA1	0.4283	0.4869	0.4651	0.4827	0.4982	0.0126	0.4477	0.4725	0.5073	0.5142
T-GPA1	0.01720	0.44731	0.47717	0.06027	0.51638	0.10733	0.00086	0.08312	0.51052	0.51107
A.CGPA	0.16767	0.46927	0.47959	0.30871	0.49320	0.09368	0.15154	0.21146	0.50717	0.51256

6. Conclusions and Future Work

This paper presents a new parameter estimation technique for the PEMFC based on the artificial gorilla troops optimizer (GTO), which has been recently proposed for tackling global optimization problems. However, unfortunately, the GTO still suffers from falling into local optima and low convergence speed, so it is modified in this paper by replacing the exploitation operator with a new one, aiding in disposing of those problems. This modified variant, abbreviated MGTO, and the standard GTO are herein assessed using three well-known PEMFC stacks, 250 W stack, BCS-500W stack, and SR-12 stack, compared with eight optimization algorithms, SOA, DE, DEGWO, GWO, FPA, SMA, BO, and HOA, under various performance metrics such as best, average, worst, SD, CPU time, convergence curve, MAE, and MAPE, in addition to a statistical test, namely the Wilcoxon rank-sum test. The outcomes of the MGTO are better than those of the compared algorithms for the employed performance metrics on all investigated PEMFCs, except the CPU time which is competitive among the algorithms. Our future work involves finding another way to make further improvements to the MGTO to reduce the consumed CPU time, while keeping or improving the current accuracy.

Author Contributions: Conceptualization, M.A.-B., R.M. and V.C.; methodology, M.A.-B., R.M. and V.C.; software, M.A.-B., R.M. and V.C.; validation, M.A.-B., R.M. and V.C.; formal analysis, M.A.-B., R.M. and V.C.; investigation, M.A.-B., R.M. and V.C.; resources, M.A.-B., R.M. and V.C.; data curation, M.A.-B., R.M. and V.C.; writing—original draft preparation, M.A.-B., R.M. and V.C.; writing—review and editing M.A.-B., R.M. and V.C.; visualization, M.A.-B., R.M. and V.C.; supervision M.A.-B., R.M. and V.C.; project administration, M.A.-B., R.M. and V.C.; funding acquisition, V.C. All authors have read and agreed to the published version of the manuscript.

Funding: V.C. Research VCR 0000163.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Zhang, W.; Wang, N.; Yang, S. Hybrid artificial bee colony algorithm for parameter estimation of proton exchange membrane fuel cell. *Int. J. Hydrogen Energy* **2013**, *38*, 5796–5806. [CrossRef]
2. Askarzadeh, A. Parameter estimation of fuel cell polarization curve using BMO algorithm. *Int. J. Hydrogen Energy* **2013**, *38*, 15405–15413. [CrossRef]
3. Amphlett, J.C.; Baumert, R.M.; Mann, R.F.; Peppley, B.A.; Roberge, P.R.; Harris, T.J. Performance modeling of the Ballard Mark IV solid polymer electrolyte fuel cell: I. Mechanistic model development. *J. Electrochem. Soc.* **1995**, *142*, 1. [CrossRef]
4. Yao, K.; Karan, K.; McAuley, K.; Oosthuizen, P.; Peppley, B.; Xie, T. A Review of Mathematical Models for Hydrogen and Direct Methanol Polymer Electrolyte Membrane Fuel Cells. *Fuel Cells* **2004**, *4*, 3–29. [CrossRef]
5. Cheddie, D.; Munroe, N. Review and comparison of approaches to proton exchange membrane fuel cell modeling. *J. Power Sources* **2005**, *147*, 72–84. [CrossRef]
6. Rajasekar, N.; Jacob, B.; Balasubramanian, K.; Priya, K.; Sangeetha, K.; Babu, T.S. Comparative study of PEM fuel cell parameter extraction using Genetic Algorithm. *Ain Shams Eng. J.* **2015**, *6*, 1187–1194. [CrossRef]
7. Mann, R.F.; Amphlett, J.C.; Hooper, M.A.I.; Jensen, H.M.; Peppley, B.A.; Roberge, P.R. Development and application of a generalised steady-state electrochemical model for a PEM fuel cell. *J. Power Sources* **2000**, *86*, 173–180. [CrossRef]
8. Bairwa, A.K.; Joshi, S.; Singh, D. Dingo Optimizer: A Nature-Inspired Metaheuristic Approach for Engineering Problems. *Math. Probl. Eng.* **2021**, *2021*, 1–12. [CrossRef]
9. Sultan, H.M.; Menesy, A.S.; Kamel, S.; Jurado, F. JAYATree Growth Algorithm for Parameter Identification of Proton Exchange Membrane Fuel Cell Models. *Int. J. Interact. Multimed. Artif. Intell.* **2020**, *6*, 101–111.
10. Zhao, W.; Wang, L.; Zhang, Z. Artificial ecosystem-based optimization: A novel nature-inspired meta-heuristic algorithm. *Neural Comput. Appl.* **2019**, *32*, 9383–9425. [CrossRef]
11. Jayakumar, A.; Ramos, M.; Al-Jumaily, A. A Novel fuzzy schema to control the temperature and humidification of PEM fuel cell system. In Proceedings of the International Conference on Fuel Cell Science, Engineering and Technology, San Diego, CA, USA, 28 June–2 July 2015; American Society of Mechanical Engineers: New York, NY, USA, 2015.
12. Song, Y.; Tan, X.; Mizzi, S. Optimal parameter extraction of the proton exchange membrane fuel cells based on a new Harris Hawks Optimization algorithm. *Energy Sources Part A Recover. Util. Environ. Eff.* **2020**, 1–18. [CrossRef]
13. Özdemir, M.T. Optimal parameter estimation of polymer electrolyte membrane fuel cells model with chaos embedded particle swarm optimization. *Int. J. Hydrogen Energy* **2021**, *46*, 16465–16480. [CrossRef]
14. Singla, M.K.; Nijhawan, P.; Oberoi, A.S. Parameter estimation of proton exchange membrane fuel cell using a novel meta-heuristic algorithm. *Environ. Sci. Pollut. Res.* **2021**, *28*, 1–16. [CrossRef] [PubMed]
15. Zhu, Y.; Yousefi, N. Optimal parameter identification of PEMFC stacks using Adaptive Sparrow Search Algorithm. *Int. J. Hydrogen Energy* **2021**, *46*, 9541–9552. [CrossRef]
16. Gupta, J.; Nijhawan, P.; Ganguli, S. Optimal parameter estimation of PEM fuel cell using slime mould algorithm. *Int. J. Energy Res.* **2021**, *45*, 14732–14744. [CrossRef]
17. Diab, A.A.Z.; Sultan, H.M.; Do, T.D.; Kamel, O.M.; Mossa, M.A. Coyote Optimization Algorithm for Parameters Estimation of Various Models of Solar Cells and PV Modules. *IEEE Access* **2020**, *8*, 111102–111140. [CrossRef]
18. Menesy, A.S.; Sultan, H.M.; Korashy, A.; Kamel, S.; Jurado, F. A modified farmland fertility optimizer for parameters estimation of fuel cell models. *Neural Comput. Appl.* **2021**, *33*, 12169–12190. [CrossRef]
19. Ben Messaoud, R.; Midouni, A.; Hajji, S. PEM fuel cell model parameters extraction based on moth-flame optimization. *Chem. Eng. Sci.* **2021**, *229*, 116100. [CrossRef]
20. Abaza, A.; El-Sehiemy, R.A.; Mahmoud, K.; Lehtonen, M.; Darwish, M.M. Optimal Estimation of Proton Exchange Membrane Fuel Cells Parameter Based on Coyote Optimization Algorithm. *Appl. Sci.* **2021**, *11*, 2052. [CrossRef]

21. Yuan, Z.; Wang, W.; Wang, H. Optimal parameter estimation for PEMFC using modified monarch butterfly optimization. *Int. J. Energy Res.* **2020**, *44*, 8427–8441. [CrossRef]
22. Rao, Y.; Shao, Z.; Ahangarnejad, A.H.; Gholamalizadeh, E.; Sobhani, B. Shark Smell Optimizer applied to identify the optimal parameters of the proton exchange membrane fuel cell model. *Energy Convers. Manag.* **2019**, *182*, 1–8. [CrossRef]
23. Guo, H.; Tao, H.; Salih, S.Q.; Yaseen, Z.M. Optimized parameter estimation of a PEMFC model based on improved Grass Fibrous Root Optimization Algorithm. *Energy Rep.* **2020**, *6*, 1510–1519. [CrossRef]
24. Yang, Z.; Liu, Q.; Zhang, L.; Dai, J.; Razmjooy, N. Model parameter estimation of the PEMFCs using improved Barnacles Mating Optimization algorithm. *Energy* **2020**, *212*, 118738. [CrossRef]
25. Turgut, O.E.; Coban, M.T. Optimal proton exchange membrane fuel cell modelling based on hybrid Teaching Learning Based Optimization—Differential Evolution algorithm. *Ain Shams Eng. J.* **2016**, *7*, 347–360. [CrossRef]
26. Hao, P.; Sobhani, B. Application of the improved chaotic grey wolf optimization algorithm as a novel and efficient method for parameter estimation of solid oxide fuel cells model. *Int. J. Hydrogen Energy* **2021**, *46*, 36454–36465. [CrossRef]
27. Fahim, S.R.; Hasanien, H.M.; Turky, R.A.; Alkuhayli, A.; Al-Shamma'A, A.A.; Noman, A.M.; Tostado-Véliz, M.; Jurado, F. Parameter Identification of Proton Exchange Membrane Fuel Cell Based on Hunger Games Search Algorithm. *Energies* **2021**, *14*, 5022. [CrossRef]
28. Yao, B.; Hayati, H. Model parameters estimation of a proton exchange membrane fuel cell using improved version of Archimedes optimization algorithm. *Energy Rep.* **2021**, *7*, 5700–5709. [CrossRef]
29. Yang, B.; Zeng, C.; Wang, L.; Guo, Y.; Chen, G.; Guo, Z.; Zhu, J. Parameter identification of proton exchange membrane fuel cell via Levenberg-Marquardt backpropagation algorithm. *Int. J. Hydrogen Energy* **2021**, *46*, 22998–23012. [CrossRef]
30. Danoune, M.; Djafour, A.; Wang, Y.; Gougui, A. The Whale Optimization Algorithm for efficient PEM fuel cells modeling. *Int. J. Hydrogen Energy* **2021**, *46*, 37599–37611. [CrossRef]
31. Diab, A.A.Z.; Tolba, M.A.; El-Magd, A.G.A.; Zaky, M.M.; El-Rifaie, A.M. Fuel Cell Parameters Estimation via Marine Predators and Political Optimizers. *IEEE Access* **2020**, *8*, 166998–167018. [CrossRef]
32. Gouda, E.A.; Kotb, M.F.; El-Fergany, A.A. Investigating dynamic performances of fuel cells using pathfinder algorithm. *Energy Convers. Manag.* **2021**, *237*, 114099. [CrossRef]
33. Ben Messaoud, R. Parameters determination of proton exchange membrane fuel cell stack electrical model by employing the hybrid water cycle moth-flame optimization algorithm. *Int. J. Energy Res.* **2021**, *45*, 4694–4708. [CrossRef]
34. Qin, F.; Liu, P.; Niu, H.; Song, H.; Yousefi, N. Parameter estimation of PEMFC based on Improved Fluid Search Optimization Algorithm. *Energy Rep.* **2020**, *6*, 1224–1232. [CrossRef]
35. Dai, C.; Chen, W.; Cheng, Z.; Li, Q.; Jiang, Z.; Jia, J. Seeker optimization algorithm for global optimization: A case study on optimal modelling of proton exchange membrane fuel cell (PEMFC). *Int. J. Electr. Power Energy Syst.* **2011**, *33*, 369–376. [CrossRef]
36. El-Fergany, A.A. Electrical characterisation of proton exchange membrane fuel cells stack using grasshopper optimiser. *IET Renew. Power Gener.* **2018**, *12*, 9–17. [CrossRef]
37. Yuan, Z.; Wang, W.; Wang, H.; Yildizbasi, A. Developed Coyote Optimization Algorithm and its application to optimal parameters estimation of PEMFC model. *Energy Rep.* **2020**, *6*, 1106–1117. [CrossRef]
38. Niu, Q.; Zhang, H. and Li, K. An improved TLBO with elite strategy for parameters identification of PEM fuel cell and solar cell models. *Int. J. Hydrog.* **2014**, *39*, 3837–3854. [CrossRef]
39. Lai, G.; Li, L.; Zeng, Q.; Yousefi, N. Developed owl search algorithm for parameter estimation of PEMFCs. *Int. J. Ambient. Energy* **2020**, 1–10. [CrossRef]
40. Houssein, E.H.; Hashim, F.A.; Ferahtia, S.; Rezk, H. An efficient modified artificial electric field algorithm for solving optimization problems and parameter estimation of fuel cell. *Int. J. Energy Res.* **2021**, *45*, 20199–20218. [CrossRef]
41. Al-Shamma'a, A.A.; Ahmed Ali, F.A.; Alhoshan, M.S.; Alturki, F.A.; Farh, H.M.H.; Alam, J.; Al Sharabi, K. Proton Exchange Membrane Fuel Cell Parameter Extraction Using a Supply–Demand-Based Optimization Algorithm. *Processes* **2021**, *9*, 1416. [CrossRef]
42. Yuan, Z.; Wang, W.; Wang, H.; Ashourian, M. Parameter identification of PEMFC based on Convolutional neural network optimized by balanced deer hunting optimization algorithm. *Energy Rep.* **2020**, *6*, 1572–1580. [CrossRef]
43. Alsaidan, I.; Shaheen, M.; Hasanien, H.; Alaraj, M.; Alnafisah, A. Proton Exchange Membrane Fuel Cells Modeling Using Chaos Game Optimization Technique. *Sustainability* **2021**, *13*, 7911. [CrossRef]
44. Askarzadeh, A.; Rezazadeh, A. A new heuristic optimization algorithm for modeling of proton exchange membrane fuel cell: Bird mating optimizer. *Int. J. Energy Res.* **2013**, *37*, 1196–1204. [CrossRef]
45. Correa, J.; Farret, F.A.; Canha, L.; Simoes, M. An Electrochemical-Based Fuel-Cell Model Suitable for Electrical Engineering Automation Approach. *IEEE Trans. Ind. Electron.* **2004**, *51*, 1103–1112. [CrossRef]
46. Mohammadi, A.; Cirrincione, G.; Djerdir, A.; Khaburi, D.A. A novel approach for modeling the internal behavior of a PEMFC by using electrical circuits. *Int. J. Hydrogen Energy* **2018**, *43*, 11539–11549. [CrossRef]
47. Outeiro, M.T.; Chibante, R.; Carvalho, A.S.; de Almeida, A.T. A new parameter extraction method for accurate modeling of PEM fuel cells. *Int. J. Energy Res.* **2009**, *33*, 978–988. [CrossRef]
48. El-Fergany, A.A.; Hasanien, H.M.; Agwa, A.M. Semi-empirical PEM fuel cells model using whale optimization algorithm. *Energy Convers. Manag.* **2019**, *201*, 112197. [CrossRef]

49. Saleh, I.M.M.; Ali, R.; Zhang, H. Simplified mathematical model of proton exchange membrane fuel cell based on horizon fuel cell stack. *J. Mod. Power Syst. Clean Energy* **2016**, *4*, 668–679. [CrossRef]
50. Zhang, G.; Jiao, K. Three-dimensional multi-phase simulation of PEMFC at high current density utilizing Eulerian-Eulerian model and two-fluid model. *Energy Convers. Manag.* **2018**, *176*, 409–421. [CrossRef]
51. AAbdollahzadeh, B.; Gharehchopogh, F.S.; Mirjalili, S. Artificial gorilla troops optimizer: A new nature-inspired metaheuristic algorithm for global optimization problems. *Int. J. Intell. Syst.* **2021**, *36*, 5887–5958. [CrossRef]
52. Sultan, H.M.; Menesy, A.S.; Kamel, S.; Selim, A.; Jurado, F. Parameter identification of proton exchange membrane fuel cells using an improved salp swarm algorithm. *Energy Convers. Manag.* **2020**, *224*, 113341. [CrossRef]
53. Duan, B.; Cao, Q.; Afshar, N. Optimal parameter identification for the proton exchange membrane fuel cell using Satin Bowerbird optimizer. *Int. J. Energy Res.* **2019**, *43*, 8623–8632. [CrossRef]
54. Askarzadeh, A.; Rezazadeh, A. An Innovative Global Harmony Search Algorithm for Parameter Identification of a PEM Fuel Cell Model. *IEEE Trans. Ind. Electron.* **2012**, *59*, 3473–3480. [CrossRef]
55. Shaheen, M.A.; Hasanien, H.M.; El Moursi, M.S.; El-Fergany, A.A. Precise modeling of PEM fuel cell using improved chaotic MayFly optimization algorithm. *Int. J. Energy Res.* **2021**, *45*, 18754–18769. [CrossRef]
56. Menesy, A.S.; Sultan, H.M.; Korashy, A.; Banakhr, F.A.; Ashmawy, M.G.; Kamel, S. Effective Parameter Extraction of Different Polymer Electrolyte Membrane Fuel Cell Stack Models Using a Modified Artificial Ecosystem Optimization Algorithm. *IEEE Access* **2020**, *8*, 31892–31909. [CrossRef]
57. Houssein, E.H.; Helmy, B.E.D.; Rezk, H.; Nassef, A.M. An enhanced Archimedes optimization algorithm based on Local escaping operator and Orthogonal learning for PEM fuel cell parameter identification. *Eng. Appl. Artif. Intell.* **2021**, *103*, 104309. [CrossRef]
58. Askarzadeh, A.; Rezazadeh, A. A grouping-based global harmony search algorithm for modeling of proton exchange membrane fuel cell. *Int. J. Hydrogen Energy* **2011**, *36*, 5047–5053. [CrossRef]
59. Miao, D.; Chen, W.; Zhao, W.; Demsas, T. Parameter estimation of PEM fuel cells employing the hybrid grey wolf optimization method. *Energy* **2020**, *193*, 116616. [CrossRef]
60. Sultan, H.M.; Menesy, A.S.; Kamel, S.; Tostado-Veliz, M.; Jurado, F. Parameter Identification of Proton Exchange Membrane Fuel Cell Stacks Using Bonobo Optimizer. In Proceedings of the 2020 IEEE International Conference on Environment and Electrical Engineering and 2020 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe), Madrid, Spain, 9–12 June 2020.
61. Priya, K.; Rajasekar, N. Application of flower pollination algorithm for enhanced proton exchange membrane fuel cell modelling. *Int. J. Hydrogen Energy* **2019**, *44*, 18438–18449. [CrossRef]
62. Cao, Y.; Li, Y.; Zhang, G.; Jermsittiparsert, K.; Razmjooy, N. Experimental modeling of PEM fuel cells using a new improved seagull optimization algorithm. *Energy Rep.* **2019**, *5*, 1616–1625. [CrossRef]
63. MiarNaeimi, F.; Azizyan, G.; Rashki, M. Horse herd optimization algorithm: A nature-inspired algorithm for high-dimensional optimization problems. *Knowl. Based Syst.* **2021**, *213*, 106711. [CrossRef]
64. Adeel, M.; Hassan, A.K.; Sher, H.A.; Murtaza, A.F. A grade point average assessment of analytical and numerical methods for parameter extraction of a practical PV device. *Renew. Sustain. Energy Rev.* **2021**, *142*, 110826. [CrossRef]