APPLICATIONS OF
MODELLING AND SIMULATION
http://arqiipubl.com/ams eISSN 2600-8084 VOL 8, 2024, 283-300

Optimal Parameter Estimation of PEMFC Model Using an Improved Atomic Orbital Search Algorithm

Burcin Özkaya*

Department of Electrical Engineering, Faculty of Engineering and Natural Sciences, Bandirma Onyedi Eylul University, 10220, Bandirma, Balikesir, Turkey

*Corresponding author: bozkaya@bandirma.edu.tr

Submitted 16 June 2024, Revised 29 September 2024, Accepted 03 October 2024, Available online 26 October 2024.

Abstract: Proton exchange membrane fuel cells (PEMFCs) have drawn much attention lately for their parameter extraction. It is important to carefully determine the optimal values of the uncertain parameters in the PEMFC model to guarantee accuracy and dependability. However, because of their nonlinearity and multi-variability, PEMFC modeling and optimization present a significant difficulty. Therefore, an improved atomic orbital search algorithm based on Lévy flight and chaotic maps, called CLAOS, was proposed in this study for the PEMFC parameter estimation problem, where the sum of square error was minimized. In order to test the effect of the Levy flight and chaotic maps on the performance of the Atomic Orbital Search (AOS) algorithm, ten different AOS variations were created and applied to solve the CEC2020 and CEC2022 benchmark problem suites. Their results were analyzed using Friedman and Wilcoxon tests, and the best variant was called the CLAOS algorithm. To validate the effectiveness of the proposed CLAOS, extensive simulations and performance evaluations were conducted on the PEMFCs model, where a 250W PEMFC stack was considered. Two search ranges for the unknown parameters and two operational conditions were considered. The performance of the proposed algorithm was compared with the six meta-heuristic search algorithms. Accordingly, the proposed algorithm achieved 4.722489 and 0.152027 for Case-1 and Case-2, respectively, which were the best objective function values among its rivals. Moreover, the results of the CLAOS algorithm were compared with the results reported in the literature for both cases. Accordingly, the CLAOS achieved the minimum error value compared to its rivals for both cases. To evaluate the performance of the algorithms statistically, the Friedman and Wilcoxon tests were applied to the results of the algorithms. The Friedman test results show that the proposed CLAOS algorithm ranked first with 1.1667 and 1.0000 score values for Case-1 and Case-2, respectively. All simulation and analysis results demonstrated that the proposed algorithm outperformed its rivals in solving the PEMFC parameter estimation problem.

Keywords: Atomic orbital search; Chaotic map; Lévy flight; Parameter estimation; Proton exchange membrane fuel cell.

1. INTRODUCTION

Researchers have recently focused more on developing new technologies in power generation systems due to rising energy demand, depletion of traditional fuel sources, and environmental pollution. Fuel cells (FCs), among the most popular new technologies, are a promising option due to their low environmental impact, high efficiency, and low noise [1, 2]. Fuel cells produce electrical energy through the chemical conversion of fuel and an oxidant [3]. There are many different kinds of FCs, but because of their high-power efficiency, low operating temperature, and lack of contamination, proton exchange membrane fuel cells (PEMFCs) are the subject of extensive research and show great potential for a wide range of applications [1-3]. The efficient design of PEMFC systems requires accurate models. The polarization curve of the PEMFC, a crucial feature that illustrates the connection between the output voltage and current, is typically the focus of the modeling [1]. A popular and helpful modeling strategy is to build mathematical models based on empirical or semi-empirical equations. However, the estimation of the unknown parameters of the PEMFC model is difficult because of their multi-variable, highly non-linear, and tightly coupled nature [2]. Furthermore, since numerous local minima are encountered in the PEMFC parameter estimation problem, classical optimization techniques are not preferred [4].

Recently, owing to the important developments of artificial intelligence-based methodologies, numerous researchers have employed meta-heuristic search (MHS) algorithms to extract the undefined parameters of the PEMFC model. Considering the PEMFC parameter estimation as an optimization problem, MHS algorithms are the most dependable and efficient technique for application. Some of the MHS algorithms, that have been employed, are bald eagle search algorithm [2], real coded genetic algorithm [5], grey wolf optimization algorithm [6], multi-verse optimizer [7], slime mould algorithm [8], gradient-based

optimizer [9], honey badger optimizer [10], prairie dog optimization algorithm [11], and many more [11-14]. On the other hand, researchers have applied improved MHS algorithms to solve the PEMFC parameter estimation problem. Some of these improved MHS algorithms are a hybrid adaptive differential evolution algorithm [15], modified monarch butterfly optimization algorithm [16], improved barnacle mating optimizer [17], improved artificial ecosystem optimizer [18], hybrid sine-cosine crow search algorithm [19], enhanced transient search optimization algorithm [20], improved bonobo optimizer [21], improved crow search algorithm [22], improved red fox optimizer [23], repairable grey wolf optimization algorithm [24], improved fish migration optimization [25], improved artificial bee colony algorithm [26], a hybrid algorithm including the osprey optimization algorithm and the coati optimization algorithm [27], and many more [28-32]. Besides, to improve the performance of algorithms, the chaos theory and Levy flight are also used in the literature. Askarzadeh and dos Santos Coelho [33] improved the backtracking search algorithm by using chaotic map, which was used to solve the PEMFC parameter estimation problem. Yuan *et al*. [34] developed the coyote optimization algorithm including the chaotic map and Lévy flight (LF) mechanism for determining the model parameters of PEMFCs. Özdemir [35] proposed an improved particle swarm optimization algorithm using chaotic maps to identify three types of PEMFCs. Qin *et al*. [36] improved the fluid search optimization algorithm based on the chaos theory to optimize the parameters of three types of PEMFCs. Ćalasan *et al*. [37] proposed an improved evaporation rate water cycle algorithm using chaotic maps for extracting the unidentified parameters of the BCS 500 W and Ballard Mark V 5 kW stacks. Xuebin *et al*. [12] presented a comparative study based on artificial hummingbird algorithm and improved artificial hummingbird algorithm using Lévy flight for identifying the unknown parameters of the PEMFC stacks. The literature review indicates that identifying uncertain parameters in PEMFCs is a significant area of research. Numerous optimization techniques have been proposed to achieve lower errors, enhance convergence speed, and improve statistical performance. The "No Free Lunch" theory influences the competition, which suggests that no unique algorithm can effectively address all engineering optimization problems.

The aforementioned has encouraged the author to investigate the solution to the PEMFC parameter estimation problem by improving the performance of the atomic orbital search (AOS) algorithm proposed by Azizi [38] in the literature in 2021. The fundamental idea of the AOS algorithm is derived from quantum physics, which applies some of the ideas of quantum mechanics, and the quantum-based atomic model, which places electron placements surrounding the nucleus in context. The AOS algorithm has been used in varying optimization problems. In this context, Ali *et al*. [39] used the AOS algorithm to extract the unknown parameters of solar cells. Azizi *et al*. [40] applied the AOS to solve various engineering design problems. Compared to other meta-heuristic optimization techniques, the AOS has yielded encouraging results. However, the practical implementation of the system has revealed some drawbacks. Specifically, the original version of the algorithm lacks adequate exploration. It fails to balance exploration and exploitation, leading to premature convergence and stagnation at a local optimum.

The literature shows that modeling the complex PEMFC model yields favorable results when employing MHS algorithms. In this context, the primary contribution of this paper is the development of an enhanced version of the AOS algorithm, which tackles the parameter extraction problem of the PEMFC model for the first time. When solving the PEMFC parameter estimation problem, the AOS algorithm encountered issues like premature convergence and local minima. Therefore, the chaotic map functions and Lévy flight were integrated into the AOS algorithm to overcome these disadvantages. Thus, the aim was to increase the AOS's exploration ability and establish a good balance between exploration and exploitation of the AOS algorithm. In this study, the proposed algorithm was called a chaotic and Lévy based atomic orbital search (CLAOS) algorithm. Thus, the main contributions of this paper are summarized as follows:

- A robust MHS algorithm called the CLAOS algorithm was presented to the literature.
- Evaluation of the CLAOS algorithm on CEC2020 and CEC2022 benchmark problems shows its superiority over the basic AOS algorithm.
- The fact that CLAOS works well for extracting parameters in the PEMFC model under a range of operating conditions shows that it is better than the base AOS algorithm.
- The CLAOS algorithm has achieved the best optimal solutions among the six most commonly used MHS algorithms in the literature in solving the PEMFC parameter estimation problem.
- A comprehensive analysis using statistical analysis methods, namely Friedman and Wilcoxon tests, was carried out to validate the proposed algorithm's performance in solving the benchmark problems and the PEMFC parameter estimation problem.

The remaining part of the study is organized as follows: Section 2 presents the modeling of the PEMFC and the objective function considered in the study. Section 3 presents an overview of the AOS algorithm, the definition of the Lévy flight, an overview of the chaotic map functions, and the design of the proposed algorithm. Section 4 gives the results of the proposed algorithm, and the base algorithm obtained from both benchmark problems and the PEMFC parameter estimation problem. Section 5 presents the conclusion of the study.

2. PROBLEM FORMULATION

2.1 Modeling of PEMFC

A type of fuel cell called PEMFC uses an electrochemical reaction with oxygen to transform chemical energy from hydrogen into electrical energy. The catalyst, anode and cathode electrodes, fuel cell stack, and proton exchange membrane are the essential parts of a PEMFC. The anode and cathode sides supply different gases into PEMFCs, with the anode feeding hydrogen-containing gases and the cathode feeding oxygen-containing gases, respectively [1-4]. The electrochemical reactions

can be represented as [4]:

$$\frac{1}{2}O_2 + H_2 \rightarrow H_2O + Heat + Electricity \quad (1)$$

As a result of the reaction, heat, DC electricity and water vapor are produced. The DC voltage at the PEMFC output is caused by some voltage losses such as ohmic (V_{ohmic}) loss, concentration (V_{conc}) loss, and activation (V_{act}) loss. The definition of a single PEMFC voltage is as follows [4]:

$$V_{FC} = -(V_{ohmic} + V_{con} + V_{act}) + E_n \quad (2)$$

where E_n is the Nernst voltage given in Equation (3) [4].

$$E_n = 4.3085 \times 10^{-5} * T * In\left(P_{H_2}\sqrt{P_{O_2}}\right) - 0.85 * 10^{-3} \times (T - 298.15) \quad (3)$$

In Equation (3), P_{H_2} and P_{O_2} are the partial gas pressures of oxygen and hydrogen (atm), respectively, and T represents the cell's temperature (K). When the reactants are air and H_2, P_{O_2} is calculated by Equation (4). In both cases of reactive conditions, P_{H_2} is calculated by Equation (5) [4].

$$P_{O_2} = \left(RH_c * P_{H_2O}^{sat}\right) \times \left[\left(exp\left(\frac{4.192(I/A)}{T^{1.334}}\right) * \left(\frac{RH_c * P_{H_2O}^{sat}}{P_c}\right)\right)^{-1} - 1\right] \quad (4)$$

$$P_{H_2} = 0.5\left(RH_a * P_{H_2O}^{sat}\right) \times \left[\left(exp\left(\frac{1.635(I/A)}{T^{1.334}}\right) * \left(\frac{RH_a * P_{H_2O}^{sat}}{P_a}\right)\right)^{-1} - 1\right] \quad (5)$$

In Equations (4) and (5), RH_c and RH_a represent the relative humidity of the vapor in the electrodes, respectively. P_c and P_a represent the cathode and anode inlet pressures (atm), respectively. $P_{H_2O}^{sat}$ is the saturation pressure of water vapor (atm) represented as follows [4]:

$$log_{10}\left(P_{H_2O}^{sat}\right) = 2.95 \times 10^{-2}(T - 273.15) - 9.18 \times 10^{-5}(T - 273.15)^2 + 1.44 \times 10^{-7}(T - 273.15)^3 - 2.18 \quad (6)$$

Based on the cell's operating region, PEMFCs typically have three overpotentials: activation voltage in the lower current region, voltage drop due to ohmic resistance, and concentration overvoltage in the higher current region. The activation voltage drop is represented by V_{act} computed using the following mathematical formula [4]:

$$V_{act} = -\left(\xi_3 In\left(\frac{P_{O_2}}{5.08 \times 10^6 exp^{-(498/T)}}\right) + \xi_2 T + \xi_1\right) + \xi_4 T In\left(I_{fc}\right) \quad (7)$$

where ξ_1, ξ_2, ξ_3, and ξ_4 are semi-empirical parameters, and I_{fc} is the stack current. The V_{ohmic} can be computed as [4]:

$$V_{ohmic} = I_{fc}(R_m + R_c) = I_{fc}\left(\frac{\rho_m l}{A} + R_c\right) \quad (8)$$

where R_m and R_c are the resistances of membrane and connections, respectively. ρ_m is the membrane resistivity calculated using Equation (9) [4].

$$\rho_m = \frac{181.6\left[1 + 0.03\left(\frac{I_{fc}}{A}\right) + 0.062\left(\frac{T}{303}\right)^2\left(\frac{I_{fc}}{A}\right)^{2.5}\right]}{\left[\lambda - 0.634 - 3\left(\frac{I_{fc}}{A}\right)\right]\exp\left(4.18\left(\frac{T-303}{T}\right)\right)} \quad (9)$$

The concentration voltage drop is defined in Equation (10), where J_{max} and J are the cell's maximum current and current density [4].

$$V_{conc} = -\beta In\left(\frac{J}{J_{max}}\right) \quad (10)$$

On the other hand, many PEMFCs are connected in series to generate a larger voltage that meets the application's needs. Equation (11), where N_c is the series-connected fuel cell, can be used to estimate the total theoretical net stack voltage [4].

$$V_s = N_c * (-(V_{ohmic} + V_{con} + V_{act}) + E_n) \quad (11)$$

2.2 Objective Function

In the mathematical model of the PEMFC cell, seven unknown parameters need to be identified. Accordingly, the goal is to minimize the relevant objective function by determining the optimal values for these parameters in the parameter estimation

of the PEMFC problem. This study considers the objective function as the sum of the square errors (SSE). When calculating the SSE value, the error is the difference between the actual and estimated output voltage. Then, the square of the sum of these error values is calculated. As a result, the problem formulation of the PEMFC parameter estimation problem can be defined as [4]:

$$OF = Minimize\ SSE(\boldsymbol{x}) = \sum_{j=1}^{N}\left(V_{data,j} - V_{meas,j}\right)^2, \text{where } \boldsymbol{x} = [\xi_1, \xi_2, \xi_3, \xi_4, \lambda, R_c, \beta] \tag{12}$$

where V_{data} and V_{meas} are the experimental and measured voltage, respectively, N is the number of data.

3. PROPOSED ALGORITHM

3.1 Overview of Atomic Orbital Search Algorithm

The AOS is a population-based MHS optimization algorithm introduced by Azizi in 2021 [38]. It is based on quantum physics and describes the locations of electrons around a nucleus by combining ideas from the atomic model and quantum mechanics. In this algorithm, the electrons in the orbitals around a nucleus are denoted as the solution candidates. The search space is the distributed volume of electrons surrounding the nucleus, separated into thin, spherical, and concentric layers. Each electron is associated with a solution candidate within this search space, and certain decision variables determine their positions. Every electron in the quantum-based atomic model has an energy state, represented mathematically as a solution candidate's objective function value. Higher energy levels are associated with solution candidates having worse objective function values, whereas lower energy levels are associated with those with better objective function values [38].

Similar to other population-based MHS algorithms, the initial positions of electrons in the electron cloud are randomly generated using Equation (13) in the AOS algorithm [38].

$$z_j^i = z_{j,min}^i + r\left(z_{j,max}^i - z_{j,min}^i\right),\ j = 1,2,\dots,s;\ i = 1,2,\dots,d \tag{13}$$

In Equations (14) and (15), Z_j^q and E_j^q are the j^{th} solution candidate and its objective function value in the q^{th} imaginary layer. n and d are the maximum number of imaginary layers and the problem dimension, respectively [38].

$$Z^q = \begin{bmatrix} Z_1^q \\ \vdots \\ Z_j^q \\ \vdots \\ Z_s^n \end{bmatrix} = \begin{bmatrix} z_1^1 & \cdots & z_1^i & \cdots & z_1^d \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ z_j^1 & \cdots & z_j^i & \cdots & z_j^d \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ z_s^1 & \cdots & z_s^i & \cdots & z_s^d \end{bmatrix},\ j = 1,2,\dots,s; i = 1,2,\dots,d; q = 1,2,\dots,n \tag{14}$$

$$E^q = \begin{bmatrix} E_1^q & \cdots & E_j^q & \cdots & E_s^n \end{bmatrix},\ j = 1,2,\dots,s; q = 1,2,\dots,n \tag{15}$$

The quantum-based atomic model assumes that the electrons surrounding the nucleus are in their lowest energy state. The solution candidates in hypothetical layers do not know about candidates positioned in the same or different layers. Based on the location and values of the objective function of solution candidates in each layer, the mathematical model computes the binding energy, which is the energy needed to extract an electron from its shell. Thus, in these theoretical layers, the binding state and energy of solution candidates are found by averaging the positions and values of the objective functions of all the candidates in that layer. The following mathematical equations are given as follows [38]:

$$BS^q = \frac{\sum_{j=1}^{s} Z_j^q}{s}\ and\ BE^q = \frac{\sum_{j=1}^{s} E_j^q}{s},\ j = 1,2,\dots,s; q = 1,2,\dots,n \tag{16}$$

where BE^q and BS^q are the binding energy and state of the q^{th} layer.

Similarly, an atom's the Binding State and Binding Energy have been framed using the average of objective function values and positions of all solution candidates in the search space as given in Equation (17). Here, Z_j and E_j are the position and objective function values of the j^{th} solution candidate, respectively. m is the total number of solution candidates [38].

$$BS = \frac{\sum_{j=1}^{m} Z_j}{m}\ and\ BE = \frac{\sum_{j=1}^{m} E_j}{m}\ j = 1,2,\dots,m \tag{17}$$

A random uniformly distributed (δ) value in the range of (0, 1) is generated for every solution candidate, theoretically representing the act photon on electrons surrounding the nucleus. In addition, the Photon Rate (*PR*) is determined as a parameter that represents the probability of considering the act of photons on electrons. If δ is higher than or equal to *PR*, then photon emission and absorption determine the motion of electrons. In this case, the binding energy BE^q of the associated hypothetical layer is compared to the energy level (E_j^q) of each electron. If E_j^q is higher than BE^q, the photon emission is considered. The electron moves to the atom's lowest energy level (*LE*) and then to the binding state (*BS*) at the same time. The following mathematical formula is used in this method to update the positions of solution candidates [38]:

$$Z_{j+1}^{q} = Z_{j}^{q} + \frac{\psi_i(\zeta_i * LE - \mu_i * BS)}{q},\ j = 1,2,\dots,s; q = 1,2,\dots,n \tag{18}$$

where Z_{j+1}^{q} is the position of the $(j+1)^{th}$ solution candidate, ψ_i, ζ_i, and μ_i are the vectors, including randomly generated numbers within (0, 1). When a solution candidate's energy level in a particular layer is less than the layer's binding energy ($E_j^q < BE^q$), photon absorption is considered. In this process, the position update of the solution candidates occurs as given in Equation (19) [38].

$$Z_{j+1}^{q} = Z_{j}^{q} + \psi_i(\zeta_i * LE^q - \mu_i * BS^q),\ j = 1,2,\dots,s; q = 1,2,\dots,n \tag{19}$$

If δ is less than PR, predicting how a photon will affect an electron is impossible. Electron migration between layers is analyzed regarding its and interactions with magnetic fields and other particles are analyzed. In this regard, the solution candidate's position is updated using Equation (20). Here, r_j is the vector consisting of randomly generated numbers [38].

$$Z_{j+1}^{q} = Z_{j}^{q} + r_j,\ j = 1,2,\dots,s; q = 1,2,\dots,n \tag{20}$$

3.2 Lévy Flight

Lévy flight is a type of random walk where the step lengths have a heavy-tailed probability distribution. It means that a particle moving by Lévy flight takes numerous regular "small" steps and infrequent "big" steps. The particle moves locally at first, taking several tiny movements, then takes a significant step, and then moves locally again. Lévy flight is often used to enhance the ability to explore the MHS algorithm [41]. It can be defined as follows:

$$L(s,\eta,\chi) = \begin{cases} \sqrt{\eta/2\pi}\, exp\left(-\frac{\eta}{2(s-\chi)}\right)\frac{1}{(s-\chi)^3/_2}, & 0 < \chi < s < \infty \\ 0, & otherwise \end{cases} \tag{21}$$

where s, η, and χ represent the sample, control parameters, and minimum step size of the Lévy flight distribution. Depending on the values of the s and η parameters, the step size in the search space between the two flights is determined.

3.3 Chaotic Map Functions

An effective way to improve the performance of optimization algorithms is to employ mathematical structures called chaotic maps, which display deterministic yet unexpected behavior. These maps enhance exploration and exploitation capabilities and avoid premature convergence [42]. Ten frequently used chaotic maps are presented in Table 1.

Table 1. The description of the Chaotic map functions [42]

No	Name	Chaotic Map	Range
1	Chebyshev	$u_{i+1} = \cos(n\cos^{-1}(u_i))$	(-1, 1)
2	Circle	$u_{i+1} = \mathrm{mod}(u_i + a - (b/2\pi)\sin(2\pi u_k), 1)$, a=0.2 and b=0.5	(0, 1)
3	Gauss/Mouse	$u_{i+1} = \begin{cases} 1, & u_i = 0 \\ \frac{1}{mod(u_i,1)}, & otherwise \end{cases}$	(0, 1)
4	Iterative	$u_{i+1} = \sin(b\pi/u_i)$, b=0.7	(-1, 1)
5	Logistic	$u_{i+1} = bu_i(1-u_i)$, b=0.4	(0, 1)
6	Piecewise	$u_{i+1} = \begin{cases} u_i/R, & 0 \le u_i \le R \\ \frac{u_i-R}{0.5-R}, & R \le u_i \le 0.5 \\ \frac{1-u_i-R}{0.5-R}, & 0.5 \le u_i \le 1-R \\ \frac{1-u_i}{R}, & 1-R \le u_i \le 1 \end{cases}$, R=0.4	(0, 1)
7	Sine	$u_{i+1} = \left(b/4\right)\sin(u_i\pi)$, b=4	(0, 1)
8	Singer	$u_{i+1} = \varpi(7.86u_i - 23.31u_i^2 + 28.75u_i^3 - 13.302875u_i^4)$, $\varpi = 1.07$	(0, 1)
9	Sinusoidal	$u_{i+1} = bu_i^2\sin(u_i\pi)$, b=2.3	(0, 1)
10	Tent	$u_{i+1} = \begin{cases} u_i/0.7, & u_i < 0.7 \\ 10/3\,(1-u_i), & u_i \ge 0.7 \end{cases}$	(0, 1)

3.4 Proposed Chaotic and Lévy based Atomic Orbital Search Algorithm

In the literature, researchers have conducted different studies to improve the search performance of the AOS algorithm [43,44]. In this study, the chaotic maps and levy flight distribution were integrated into the AOS algorithm to enhance its overall performance, especially its exploration and exploitation abilities. This improved algorithm was called the chaotic and Lévy-based atomic orbital search (CLAOS) algorithm. The design of the proposed algorithm was carried out as follows.

In the original AOS algorithm, the δ value is a random uniformly distributed number in the range of (0, 1) and generated for each solution candidate in the population. As a result of comparing the δ value with the *PR* value, if δ is higher than or equal to *PR*, photon emission and absorption determine the movement of electrons. Otherwise, predicting how a photon will affect the electron is impossible. This process was explained in detail in sub-section 3.1.

Firstly, the δ value is generated using chaotic maps before the search process life cycle of the algorithm for maximum fitness function evaluations (*maxFEs*) number in the proposed algorithm. Thus, the aim is to improve the exploration capability of the AOS algorithm. Ten variations of the CLAOS algorithm have been created using the ten chaotic maps given in Table 1 to determine the δ value, and the description of them was given in Table 2. Secondly, the ψ vector given in Equations (18) and (19), where the population updates are performed, is randomly generated. In the proposed algorithm, this vector is obtained using the Lévy flight. In this way, it aims to ensure the balance between exploration and exploitation and to prevent the algorithm from getting caught in local solution traps.

The pseudocode of the proposed CLAOS algorithm is given in Algorithm-1. According to Algorithm-1, the initial population is created randomly using Equation (13) in line 1. Then, the fitness values of them are computed in line 2. The *BS* and *BE* of an atom are determined using Equation (17) in line 3. The solution candidate with the lowest energy level in an atom, which is the best candidate so far, is determined in line 4. In line 5, the *chaos_vector* is generated according to the case number in Table 2. For example, if Case-1 is used, the Chebyshev map function generates the *chaos_vector*. From lines 6 to 35, the search process life cycle is performed until the termination criterion is met. From lines 6 to 14, the whole process occurs in the same way as the main algorithm. However, in line 15, the δ parameter corresponds to the current *FE* value in *chaos_vector*. Then, in line 16, the ψ vector used in Equations (18) and (19) is generated using the Lévy flight. ζ and μ are produced in line 17, and the *PR* value is specified in line 18. From lines 19 to 35, the remaining process occurs in the same way as the main AOS algorithm. Figure 1 shows the flowchart of the proposed CLAOS algorithm. The orange blocks show where chaotic maps are used, and the yellow blocks show where Lévy flight is used.

Table 2. Ten different CLAOS variations created using chaotic maps.

Case	Chaotic Map	Case	Chaotic Map
Case-1	Chebyshev	Case-6	Piecewise
Case-2	Circle	Case-7	Sine
Case-3	Gauss/Mouse	Case-8	Singer
Case-4	Iterative	Case-9	Sinusoidal
Case-5	Logistic	Case-10	Tent

Algorithm-1. The pseudocode of the proposed CLAOS algorithm.

1. Create the initial population using Equation (13).
2. Compute the fitness values for the initial population and update *FE*.
3. Specify the *BS* and *BE* of an atom using Equation (17).
4. Identify the solution candidate having the lowest energy (*LE*) level of the atom
5. Generate the *chaos_vector* with the chaotic map used in the relevant case given in Table 2.
6. **while** *FE* < *maxFEs*
7. Generate several imaginary layers (*n*).
8. Create these imaginary layers.
9. Sort the solution candidates either in descending or ascending order.
10. Distribute the sorted solution candidates into the imaginary layers according to the PDF.
11. **for** $q = 1 : n$
12. Specify the BS^q and BE^q of the q^{th} layer using Equation (16).
13. Identify the solution candidate having the lowest energy (LE^q) level in the q^{th} layer.
14. **for** $j = 1 : s$
15. δ = *chaos_vector* (*FE*)
16. Generate ψ, used in Equations (18) and (19), with Lévy flight by Equation (21).
17. Produce ζ and μ.
18. Specify the *PR*.
19. **if** $\delta \geq PR$
20. **if** $E_j^q \geq BE^q$
21. Update the positions of solution candidates using Equation (18).
22. Compute the fitness value and update *FE*.
23. **else if**
24. Update the positions of solution candidates using Equation (19).

```
                    Compute the fitness value and update FE.
                end if
            else if δ < PR
                Update the positions of solution candidates using Equation (20).
                Compute the fitness value and update FE.
            end if
        end for
    end for
    Update the BS and BE of the atom.
    Update the solution candidate with the atom's lowest energy (LE) level.
end while
```

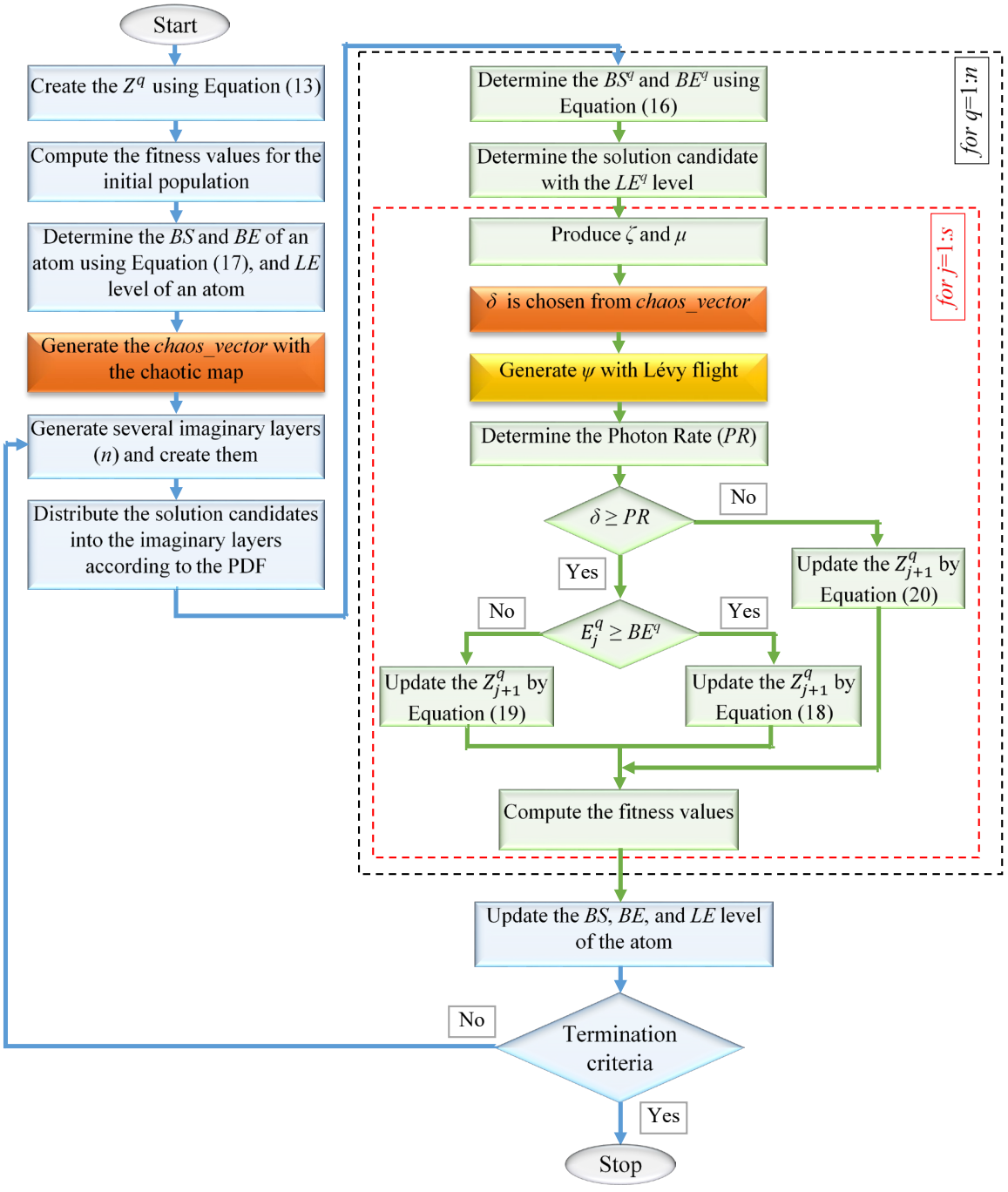

Figure 1. Flowchart of the proposed CLAOS algorithm.

4. RESULTS AND DISCUSSION

In the study, the chaotic and CLAOS algorithm was proposed. In order to test and verify the performance of the proposed algorithm, a comprehensive simulation study was carried out on solving benchmark problems and the PEMFC parameter estimation problem. The CEC2020 [45] and CEC2022 [46] benchmark problems were considered in the simulation studies. The simulation settings used in this study are described as follows:

- The termination criterion was set as maxFEs for benchmark problems.
- The CEC2020 problems were addressed in 30, 50, and 100 dimensions, while the CEC2022 problems were addressed in 10 and 20 dimensions.
- Fifty-one independent trials were performed for the CEC2020 and CEC2022 benchmark problems, and 30 independent trials were performed for the PEMFC parameter estimation problem.
- The simulation studies were implemented in MATLAB®R2020a and conducted on Intel (R) Core™ i7-8700 CPU @ 3.20GHz, 16 GB RAM, and an x64-based processor.
- The parameter settings of the CLAOS variations were the same as the original AOS algorithm.

Accordingly, this section included two sub-sections based on the benchmark problems and the PEMFC parameter estimation problem explained below:

- In the first sub-section, the performance of the ten CLAOS variations and the base AOS algorithm on solving the CEC2020 and CEC2022 benchmark problems was compared. Their results were evaluated using the Friedman and Wilcoxon tests. On the other hand, the algorithms' convergence analysis was analyzed.
- In the second sub-section, the CLAOS, the base AOS algorithm, the genetic algorithm (GA) [47], the particle swarm optimization algorithm (PSO) [48], the differential evolution algorithm (DE) [49], the grey wolf optimization algorithm (GWO) [50], and the artificial hummingbird algorithm (AHA) [51] were applied to estimate the parameters of the PEMFC stack. Here, two case studies were considered, where two different operating conditions and the search range of the decision variables were used.

4.1 Determining the best CLAOS variation on solving benchmark problem suits

The performance of ten CLAOS variations and the base AOS algorithm were presented in this section. In the simulation study, two benchmark problem suites, CEC2020 and CEC2022, were considered. While the CEC2020 benchmark suite was tested in 30-, 50-, and 100-dimensional search spaces, the CEC2022 benchmark suite was tested in 10- and 20-dimensional search spaces. All algorithms were run 51 times for each problem in the benchmark suites. Accordingly, the number of data used in the performance comparison of algorithms is 30294 (11*10*3*51+11*12*2*51). In order to analyze these data, the Friedman and Wilcoxon nonparametric test methods were used.

4.1.1. Statistical Analysis Results

This section compared the search performance of the ten CLAOS variations and the base AOS algorithm. Firstly, the Friedman test was applied to analyze the results of eleven algorithms. Table 3 presents the Friedman test results. Table 3 also presents the mean value of Friedman scores obtained by each algorithm from five experiments in the last column. According to Table 3, in the first experiment on the CEC2020 benchmark suit and 30 dimensions, Case-8 achieved the best rank among all algorithms. The Case-2 algorithm demonstrated superiority over its competitors in the remaining four experiments. Besides, Case-3 for experiment 1, Case-3 for experiment 2, and Case-3, Case-7, and Case-9 for experiment 3 fell behind the base AOS algorithm. On the other hand, Case-2 consistently showed the best performance, achieving the lowest mean rank (5.3823), followed by Case-8 (5.4566) and Case-4 (5.4700). The worst algorithm was Case-3, according to the mean rank value.

Table 3. Friedman test results for all algorithms

Method	**CEC2020**			**CEC2022**		**Mean Rank**
	***Dim*=30**	***Dim*=50**	***Dim*=100**	***Dim*=10**	***Dim*=20**	
	#1	**#2**	**#3**	**#4**	**#5**	
Case-2	5.7510	**5.4078**	**5.1529**	**5.2925**	**5.3072**	**5.3823**
Case-8	**5.4608**	5.5392	5.5804	5.3742	5.3284	5.4566
Case-4	5.4961	5.6098	5.3765	5.3709	5.4967	5.4700
Case-6	5.6000	5.5216	5.5196	5.3922	5.3480	5.4763
Case-10	5.5235	5.4961	5.3922	5.6618	5.5196	5.5186
Case-1	5.6118	5.7804	5.5824	5.6258	5.4363	5.6073
Case-5	5.7863	5.6255	5.6686	5.7255	5.5833	5.6778
Case-7	5.6431	6.0196	6.1176	5.6520	5.5049	5.7875
Case-9	6.5382	6.5216	7.0039	6.1340	6.6258	6.5647
AOS	7.2088	6.6275	6.0706	8.0147	8.4248	7.2693
Case-3	7.3804	7.8510	8.5353	7.7565	7.4248	7.7896

In addition to the Friedman test, pairwise comparisons were performed between the AOS algorithm and CLAOS variations using the Wilcoxon test. Table 4 presents the Wilcoxon test results for all algorithms. In Table 4, “Case-1 vs. AOS” shows the experiment performed between Case-1 and AOS. “+” indicates the number of problems in which the compared method performs better than the base AOS, “=” indicates the number of problems in which the compared method and the base AOS

perform equally, and "-" indicates the number of problems in which the compared method performs worse than the base AOS. According to the results given in Table 4, Case-3 was defeated against the base AOS algorithm in the experiments carried out in 50 and 100 dimensions in the CEC2020 benchmark suite. Apart from Case-3, CLAOS variations outperformed the base AOS algorithm in all remaining pairwise comparisons. The general evaluation in the last column of Table 4 indicates that the base algorithm least defeated Case-2. While Case-2 was superior in 33 problems, it drew in 9 problems and lost in 12 problems against the AOS.

Table 4. Wilcoxon test results for all algorithms

vs. AOS	**CEC2020**			**CEC2022**		**Overall**
(+ / = / -)	***Dim*=30**	***Dim*=50**	***Dim*=100**	***Dim*=10**	***Dim*=20**	**evaluation**
Case-1	6 / 2 / 2	5 / 1 / 4	5 / 0 / 5	8 / 2 / 2	8 / 3 / 1	32 / 8 / 14
Case-2	6 / 2 / 2	5 / 2 / 3	6 / 0 / 4	8 / 3 / 1	8 / 2 / 2	**33 / 9 / 12**
Case-3	6 / 0 / 4	4 / 1 / 5	4 / 1 / 5	8 / 1 / 3	8 / 0 / 4	30 / 3 / 21
Case-4	6 / 2 / 2	5 / 1 / 4	5 / 0 / 5	7 / 4 / 1	8 / 2 / 2	31 / 9 / 14
Case-5	6 / 1 / 3	5 / 1 / 4	5 / 0 / 5	8 / 3 / 1	8 / 2 / 2	32 / 7 / 15
Case-6	6 / 1 / 3	5 / 2 / 3	5 / 0 / 5	8 / 3 / 1	8 / 3 / 1	32 / 9 / 13
Case-7	6 / 1 / 3	5 / 1 / 4	5 / 0 / 5	8 / 2 / 2	8 / 2 / 2	32 / 6 / 16
Case-8	6 / 1 / 3	5 / 1 / 4	5 / 0 / 5	8 / 3 / 1	8 / 3 / 1	32 / 8 / 14
Case-9	6 / 1 / 3	5 / 0 / 5	5 / 0 / 5	7 / 4 / 1	7 / 2 / 3	30 / 7 / 17
Case-10	6 / 2 / 2	5 / 2 / 3	5 / 0 / 5	8 / 3 / 1	8 / 3 / 1	32 / 10 / 12

4.1.2. Convergence Performance Analysis

This sub-section presents the convergence performances of the ten CLAOS variations and the AOS algorithm. The CEC2020 and CEC2022 benchmark suites have four problems: unimodal, multimodal, hybrid, and composition-type problems. They are used to test the algorithms' abilities in exploitation, exploration, diversity balance, and balanced search. In order to examine the convergence capabilities of the algorithms, four types of problems were chosen from the CEC2020 benchmark problem suite: F1 (unimodal), F2 (multimodal), F6 (hybrid), and F9 (composition). Figure 2 shows the convergence graphs of the first five variations according to the Friedman ranking given in Table 3 and the AOS algorithm for CEC2020 benchmark problems. The convergence graphs were drawn based on the error values obtained from the algorithms.

For the F1-type problem, Case-2 demonstrated the best convergence performance across all dimensions. This performance shows that Case-2 was the algorithm with the best exploitation ability. For the F2-type problem, Case-2 again demonstrated superior convergence, achieving lower error values more rapidly than the other algorithms. On the other hand, except for Case-10, all other variations showed better search performance in all dimensions compared to the base algorithm. For the F6-type problem, Case-2 outperformed the other algorithms, showing the fastest convergence to the lowest error values. Other variations and the AOS algorithm obtained error values that were close to each other. For the F9-type problem, Case-2 consistently achieved faster and more effective convergence than others. While Case-10 showed the worst convergence performance for 30 and 50 dimensions, the base AOS algorithm showed the worst search performance for 100 dimensions. Upon general evaluation of all the results, Case-2 outperformed other variations and the base algorithm regarding exploitation, exploration, and balanced search ability.

The convergence analysis graphs above were drawn for the optimal solution obtained by the algorithms for the problems as a result of 51 independent runs. Therefore, box-plot plots are used to evaluate the performance of the algorithms for 51 independent runs. Accordingly, to evaluate the search performance of the algorithms, four types of problems were chosen from the CEC2022 benchmark problem suite: F1 (unimodal), F3 (multimodal), F7 (hybrid), and F12 (composition). Figure 3 presents the box plots of the first five variations according to the Friedman ranking given in Table 3 and the AOS algorithm. The box plots were drawn based on the fitness function values obtained from the algorithms. In Figure 3, the diamond marks on the box plots are the mean values of the fitness function values obtained by the algorithms due to 51 runs. In order to make a more precise comparison in terms of average value, the mean values of the algorithms are connected by lines. Moreover, the lines in the box plots are the median values of the fitness function values obtained by the algorithms due to 51 runs.

For the F1-type problem, while Case-2 had the lowest mean and median value, the AOS algorithm performed worst for all 10 dimensions. On the other hand, for 20-dimension, Case-4 achieved the lowest mean and median value, followed by Case-2. For the F3-type problem, Case-2 demonstrated much lower median error values for both dimensions than the others. Once again, the base AOS displayed the worst search performance. For the F7-type problem, Case-2 showed the smallest spread and achieved the lowest median value for both dimensions. For the F12-type problem, Case-2 had the most negligible dispersion, indicating more consistent performance. For all problems and dimensions, the AOS algorithm exhibited the widest spread and had the highest mean and median values. This result shows that the base AOS algorithm failed to solve the CEC2022 benchmark problems.

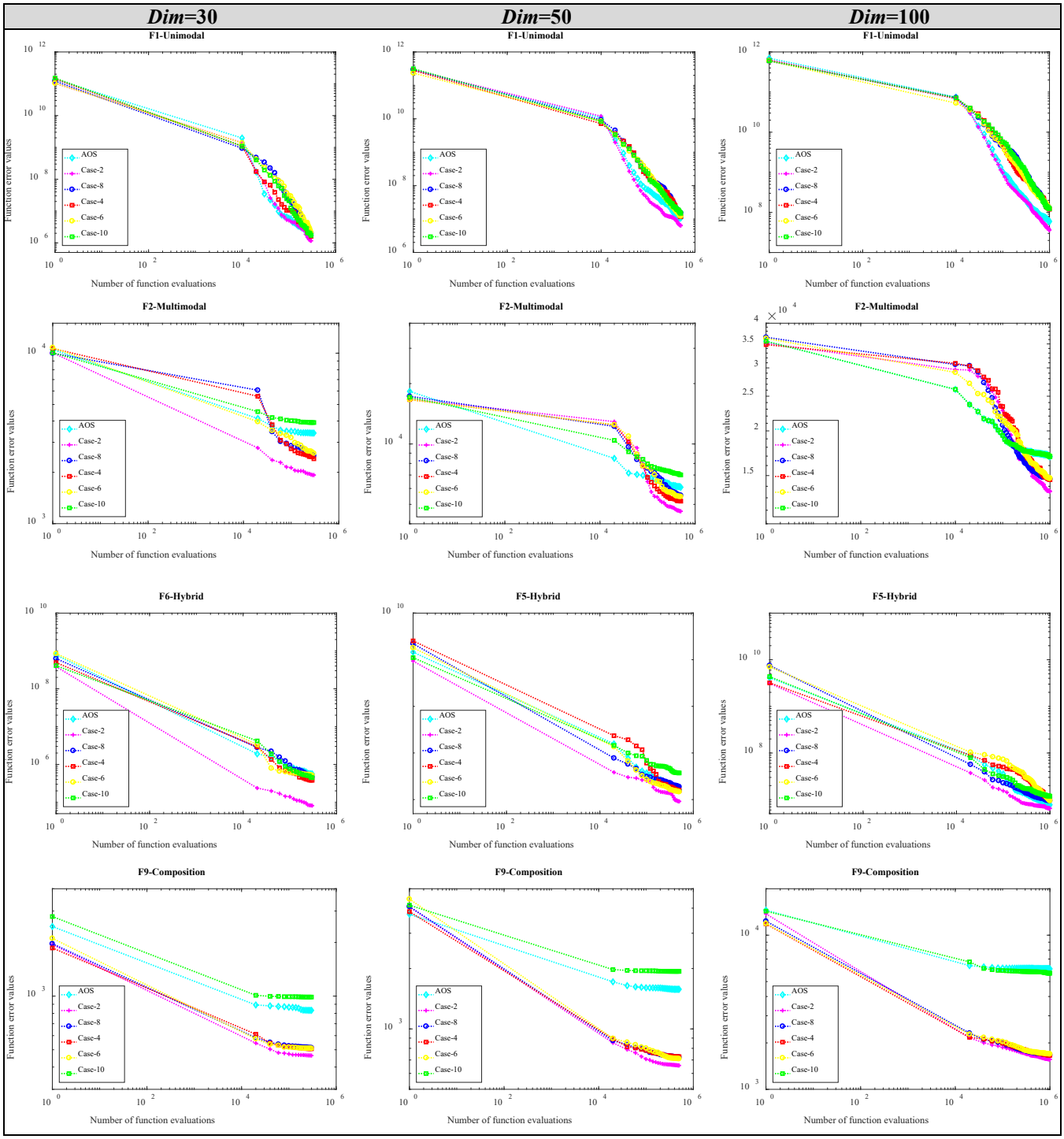

Figure 2. Convergence graphs of the first five variations according to Friedman ranking and AOS algorithm for CEC2020 benchmark problems.

Upon evaluation of all the analysis results on CLAOS variations presented in this section, it became clear that the obtained results were in agreement with each other. The search performance improved by redesigning the AOS algorithm using chaotic maps and Lévy flight. Furthermore, the utilization of the chaotic map significantly influenced the results. When all analysis results were evaluated, Case-2 emerged as the most successful variation. In the remainder of the study, Case-2 will be directly named CLAOS.

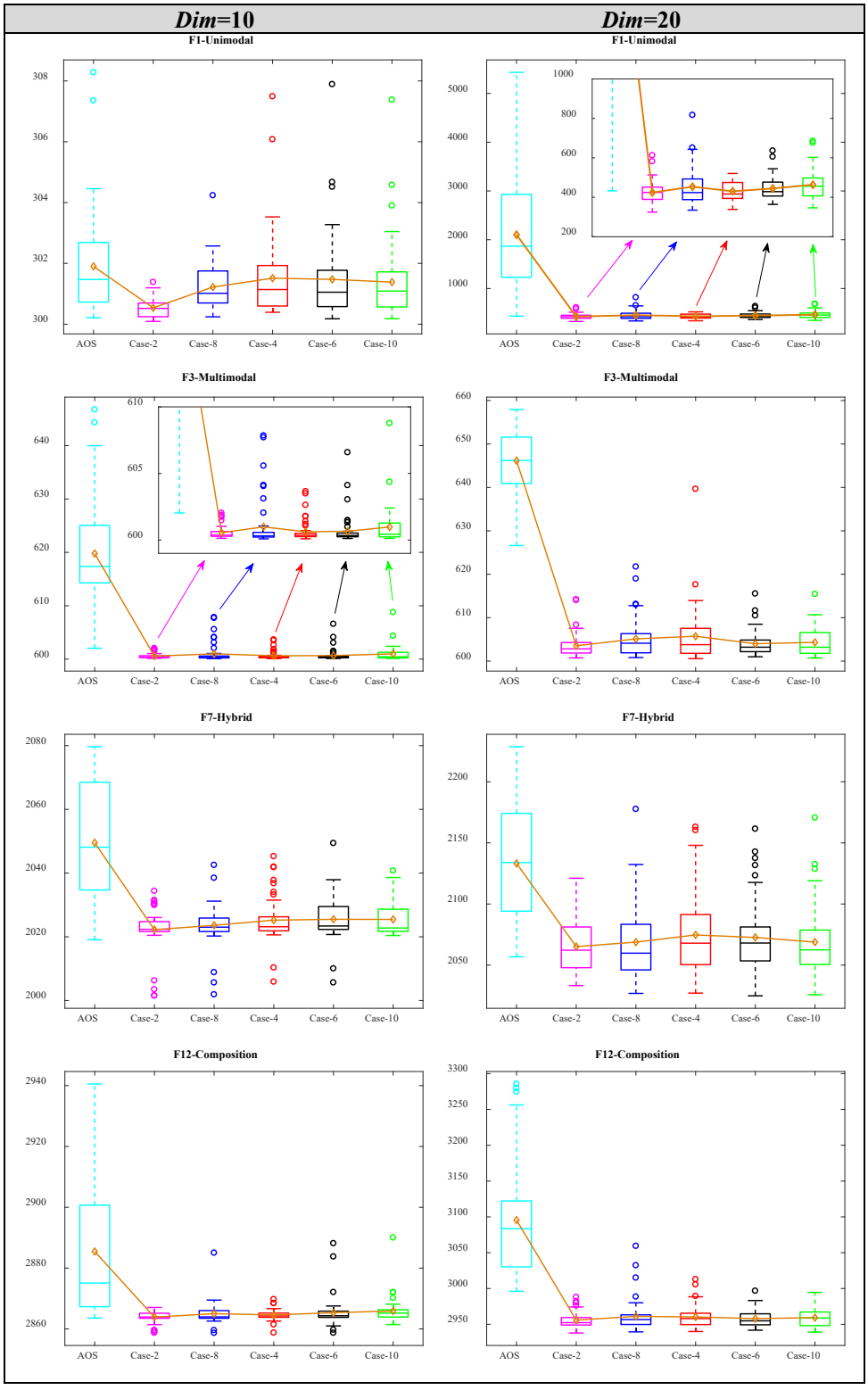

Figure 3. Box-plots of the first five variations according to Friedman ranking and AOS algorithm for CEC2022 benchmark problems.

4.2 Implementation of the CLAOS Algorithm for the PEMFC Parameter Estimation Problem

In this sub-section, the proposed CLAOS algorithm and six MHS algorithms, including AOS, AHA, GA, DE, GWO, and PSO algorithms, were applied to estimate the unknown parameter of the 250W PEMFC stack. The experimental data were taken from [4] in the simulation studies. As stated in Section 2.2, the decision variables for the parameter estimation of the PEMFC problem were ξ_1, ξ_2, ξ_3, ξ_4, λ, Rc, and b, with two search ranges for the parameters taken into consideration. On the other hand, two different operating ranges were considered, where two different values were used for P_c, P_a, and T values. Accordingly, two different case studies were considered, and their search ranges and design specifications are given in Table 5.

Table 5. Design specifications and search ranges for Case-1 and Case-2 [15].

Case-1					**Case-2**				
PEMFC specifications		Upper and lower limits (Range-1)			PEMFC specifications		Upper and lower limits (Range-2)		
Parameter	Value	Parameter	Lower	Upper	Parameter	Value	Parameter	Lower	Upper
A (cm^2)	27	ξ_1	-0.952	-0.944	A (cm^2)	27	ξ_1	-1.19969	-0.8532
N_{cell}	24	$\xi_2 \times 10^{-3}$	1	5	N_{cell}	24	$\xi_2 \times 10^{-3}$	1	5
J_{max} (mA/ cm^2)	860	$\xi_3 \times 10^{-5}$	7.4	7.8	J_{max} (mA/ cm^2)	860	$\xi_3 \times 10^{-5}$	3.6	9.8
l (μm)	127	$\xi_4 \times 10^{-5}$	-1.98	-1.88	l (μm)	127	$\xi_4 \times 10^{-5}$	-26	-9.54
RH_a	1	λ	14	23	RH_a	1	λ	10	24
RH_b	1	$R_c \times 10^{-4}$	1	8	RH_b	1	$R_c \times 10^{-4}$	1	8
P_a (bar)	1	b	0.016	0.5	P_a (bar)	3	b	0.0136	0.5
P_c (bar)	1				P_c (bar)	5			
T (K)	343.15				T (K)	353.15			

4.2.1. Results of Case-1

In order to verify the performance of the proposed CLAOS algorithm, a 250W PEMFC stack was used to search the unknown parameters of the model. The design specifications and search ranges for this case study can be found in Table 5. The proposed CLAOS and the comparison algorithms, including AOS, AHA, GA, DE, GWO, and PSO algorithms, were run separately 30 times. The maximum iteration number was set as 500. Table 6 displays the optimal parameter values and SSE values obtained by each algorithm. The SSE values for CLAOS, AOS, AHA, GA, DE, GWO, and PSO algorithms were **4.722489**, 4.944762, 4.944771, 4.944777, 4.944767, 4.944759, and 4.944831, respectively. Accordingly, the proposed CLAOS achieved a lower SSE value by 4.4951%, 4.4953%, 4.4954%, 4.4952%, 4.4951%, and 4.4964% than the AOS, AHA, GA, DE, GWO, and PSO algorithms, respectively.

The minimum, mean, maximum, median, and standard deviation values of the CLAOS, AOS, AHA, GA, DE, GWO, and PSO algorithms are presented in Table 6 to evaluate their performance. On the other hand, Table 7 presents the results of other reported methods found in the literature. The lowest SSE values for the JAYA-NM [4], RGA [5], HADE [15], MVO [7], and HABC [52] were 9.9010, 16.2746, 15.6669, 15.1316, and 15.6673, respectively. These values were higher than the SSE value of the CLAOS algorithm. Besides, the CLAOS algorithm obtained the best mean and median values among the methods presented in Table 6. To sum up, these findings demonstrate the clear superiority of the CLAOS algorithm over its competitors.

Table 6. Optimal parameter values obtained by the proposed CLAOS and AOS algorithms.

Method	ξ_1	ξ_2	ξ_3	ξ_4	λ	R_c	b	**SSE**
CLAOS	-0.95	0.0029757	0.000078	-0.000188	23	0.0001	0.04614805	**4.722489**
AOS	-0.944	0.00290381	0.000074	-0.000188	23	0.0003	0.04470466	4.944762
AHA	-0.94957428	0.00295549	7.65E-05	-1.88E-04	23	3.02E-04	0.04470604	4.944771
GA	-0.944	0.00291329	7.47E-05	-1.88E-04	23	3.00E-04	0.04471088	4.944777
DE	-0.952	0.00292723	7.40E-05	-1.88E-04	23	3.00E-04	0.04472589	4.944767
GWO	-0.952	0.00293398	7.45E-05	-1.88E-04	23	3.00E-04	0.04469022	4.944759
PSO	-0.952	0.00293068	7.42E-05	-1.88E-04	23	3.00E-04	0.0447131	4.944831

Table 7. Statistical results obtained from CLAOS, comparison algorithms, and other reported algorithms for Case-1.

Method	**Min**	**Max**	**Mean**	**Median**	**Std**
CLAOS	**4.722489**	**4.984965**	**4.748789**	**4.73458**	0.0503872
AOS	4.944762	5.098928	5.016907	5.02508	0.048963
AHA	4.944771	4.961266	4.946396	4.945488	0.003045
GA	4.944777	4.999371	4.948077	4.945577	0.009888
DE	4.944767	4.993800	4.947871	4.945588	0.009022
GWO	4.944759	4.948339	4.945479	4.945208	0.000787
PSO	4.944831	4.963357	4.947394	4.945523	0.004786
JAYA-NM [4]	9.901	9.901	9.901	9.901	5.79E-06
RGA [5]	16.2746	N/A	N/A	N/A	N/A
HADE [15]	15.6669	N/A	N/A	N/A	N/A
MVO [7]	15.1316	3145	107.1	29.84	169.9
HABC [52]	15.6673	N/A	N/A	N/A	N/A
N/A: Not available in the literature					

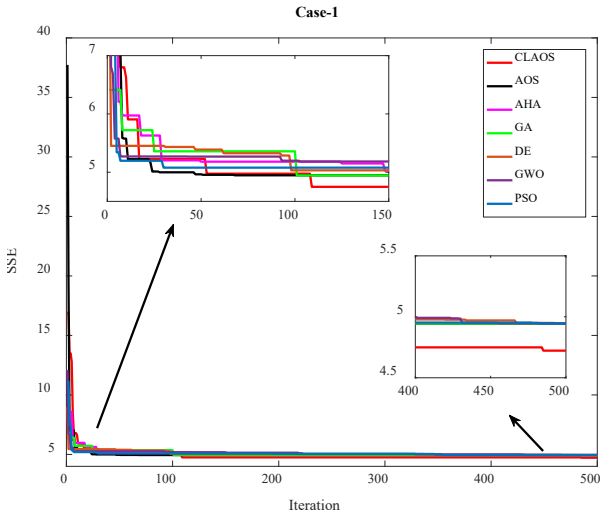

Figure 4. The convergence graphs of all algorithms for Case-1.

Besides, the convergence analysis was used to evaluate CLAOS and its competitors. From Figure 4, CLAOS was the best-performing algorithm in this comparison, achieving low SSE values quickly. On the other hand, PSO converged to a higher SSE at the end of the iteration and showed the worst convergence performance.

4.2.2. Results of Case-2

Similar to the Case-1, the 250W PEMFC stack was used in this case. However, the search range and the design specifications given in Table 5 were different from those in Case 1. The proposed CLAOS and the comparison algorithms, including AOS, AHA, GA, DE, GWO, and PSO algorithms, were run separately 30 times. The maximum iteration number was set as 500. Table 8 presents the optimal parameter values and the best objective function values obtained from CLAOS and six competitor algorithms. Accordingly, the SSE values for CLAOS, AOS, AHA, GA, DE, GWO, and PSO algorithms were **0.152027**, 1.612661, 1.61281406, 1.6131974, 1.61367413, 1.61260619, and 1.61247391, respectively.

On the other hand, the results of the CLAOS and the comparison algorithms were evaluated statistically. Table 9 presents the minimum, mean, maximum, median, and standard deviation values. Table 9 reveals that CLAOS achieved the lowest error value across all indicators. The results of the other methods reported in the literature are given in Table 9. The results of the CLAOS algorithm for Case-2 were compared with those of JAYA-NM [4], RGA [5], HADE [7], MVO [9], SSO [64], ALO [64], and HABC [63], where they obtained the objective function values as 5.2513, 8.4854, 7.9908, 3.5846, 1.1508, 1.1513, and 8.0047, respectively. When comparing the CLAOS algorithm with six comparison algorithms and the results of the reported methods in the literature, it was demonstrated that the CLAOS achieved the best SSE value and outperformed all rivals.

Figure 5 presents the convergence curves of both algorithms. Therefore, the search performance of the CLAOS algorithm was significantly superior to that of the other algorithms. In summary, all simulation results and convergence analyses confirmed the performance of the proposed CLAOS algorithm in determining the parameters of the PEMFC stack for Case-2.

Table 8. Optimal parameter values obtained by the proposed CLAOS and AOS algorithms.

Method	ξ_1	ξ_2	ξ_3	ξ_4	λ	R_c	b	**SSE**
CLAOS	-0.86	0.00208784	0.000036	-0.000136	12.1649409321	0.0001	0.014032	**0.152027**
AOS	-1.19969	0.0032006	0.0000489	-0.000135	23	0.0001	0.04	1.612661
AHA	-0.98820632	0.00257325	4.65E-05	-1.34E-04	23	0.00016	0.04	1.61281406
GA	-0.93880752	0.00230708	3.65E-05	-1.35E-04	23	0.00010	0.04	1.6131974
DE	-1.18685647	0.00317664	4.99E-05	-1.34E-04	23	0.00010	0.04	1.61367413
GWO	-1.19690734	0.00303175	3.60E-05	-1.35E-04	23	0.00010	0.04	1.61260619
PSO	-0.99425691	0.00246394	3.64E-05	-1.35E-04	23	0.00010	0.04	1.61247391

Table 9. Statistical results obtained from CLAOS, comparison algorithms, and other reported algorithms for Case-1.

Method	**Min**	**Max**	**Mean**	**Median**	**Std**
CLAOS	**0.152027**	**0.169469**	**0.160196**	**0.15967**	**0.0049355**
AOS	1.612661	1.924455	1.717473	1.6511	0.1218666
AHA	1.61281406	1.7328198	1.65069165	1.63620619	0.03537587
GA	1.6131974	1.74798106	1.65840573	1.65321379	0.03788932
DE	1.61367413	1.76475875	1.65893604	1.63986715	0.04490026
GWO	1.61260619	1.84854479	1.65168312	1.6362934	0.04817167

PSO	1.61247391	1.79248506	1.66050282	1.65832658	0.04578022
JAYA-NM [4]	5.2513	5.2538	5.2514	5.2764	6.80E-03
RGA [5]	8.4854	N/A	N/A	N/A	N/A
HADE [15]	7.9908	N/A	N/A	N/A	N/A
MVO [7]	3.5846	1.14E+04	56.28	5.08	200.9
SSO [53]	1.1508	1.4388	1.2429	N/A	0.0941
ALO [53]	1.1513	1.4481	1.3352	N/A	0.0982
HABC [52]	8.0047	N/A	N/A	N/A	N/A

N/A: Not available in the literature

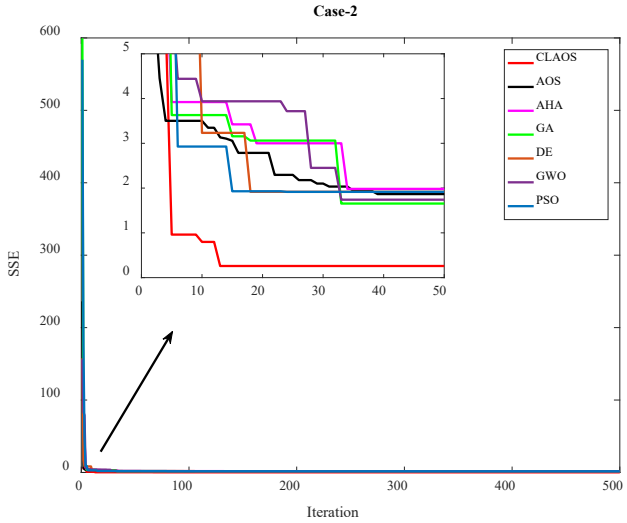

Figure 5. The convergence graphs of all algorithms for Case-2.

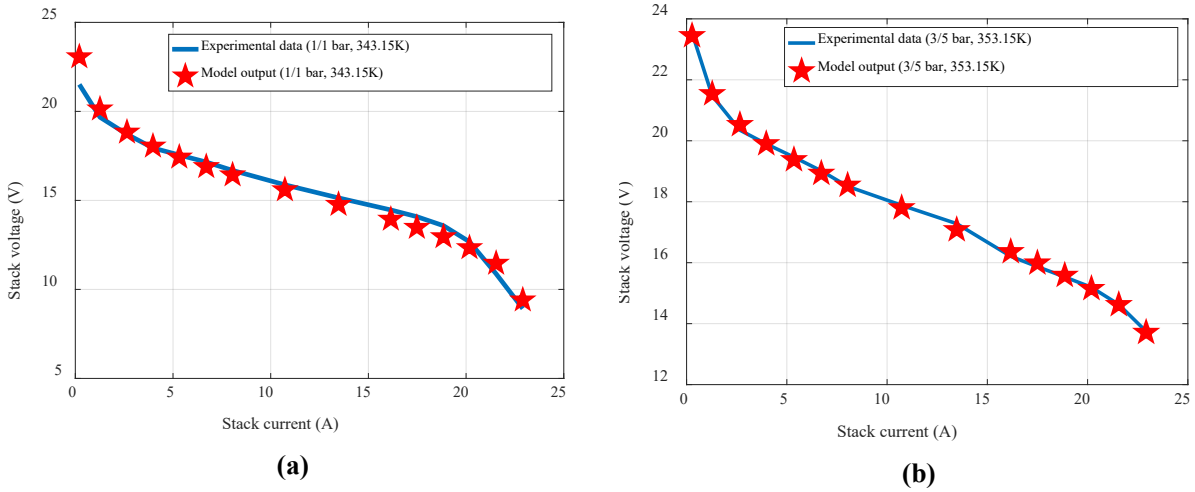

Figure 6. The current-voltage curve obtained from the CLAOS for parameters evaluation of PEMFC stack for both cases.

4.2.3. 250W PEMFC Polarization Characteristics

In order to validate the performance of the proposed algorithm in solving the PEMFC parameter estimation problem, the current-voltage (I-V) polarization curves were drawn for both cases. Figure 6 presents the I-V curves derived from the CLAOS for both cases. This graphic compares the polarization curves produced using the measured voltage/current data sets and those derived from the optimized model. Figure 6(a) shows the I-V curve for Case-1. It matched the output model data from the proposed algorithm and the experimental data. On the other hand, the I-V curve for Case-2, given in Figure 6(b), matched the output model data and experimental data obtained by the proposed algorithm quite well. The results showed that the proposed method was excellent at finding the unknown parameters of the PEMFC stack. This was clear from the fact that the experimental data and model output curves fit together so well.

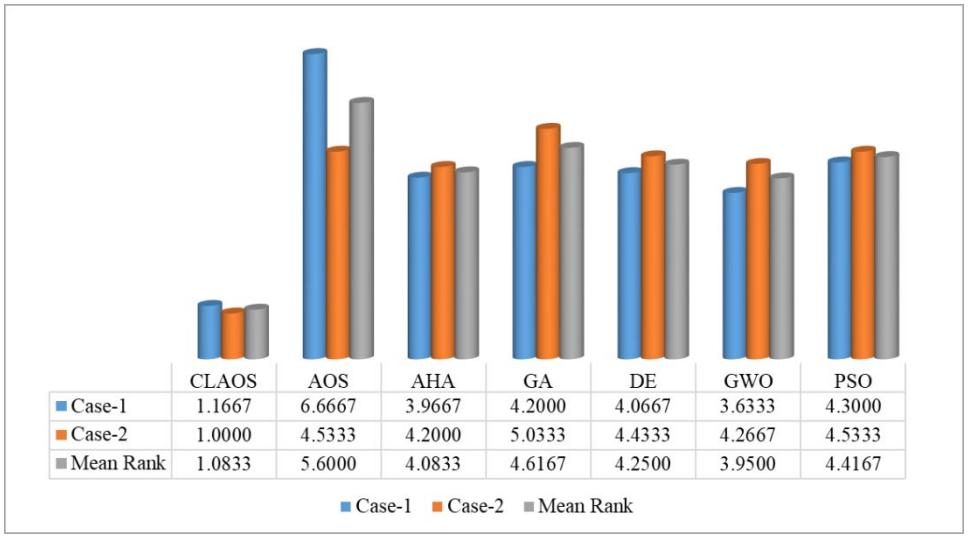

Figure 7. Friedman test results for all algorithms for Case-1 and Case-2.

Table 10. Wilcoxon test results for Case-1 and Case-2 between CLAOS and its rivals.

CLAOS vs.	**Case-1**			**Case-2**		
	X+	**X-**	***p***	**X+**	**X-**	***p***
AOS	465	0	1.73E-06	465	0	1.73E-06
AHA	464	1	1.92E-06	465	0	1.73E-06
GA	464	1	1.92E-06	465	0	1.73E-06
DE	464	1	1.92E-06	465	0	1.73E-06
GWO	464	1	1.92E-06	465	0	1.73E-06
PSO	464	1	1.92E-06	465	0	1.73E-06

4.2.4. Statistical Analysis

The Friedman and Wilcoxon tests have been widely used in the literature to evaluate the performance of meta-heuristic search algorithms. The Friedman test compares algorithms according to their performance in experimental studies with more than two competitors. On the other hand, the Wilcoxon test makes pairwise comparisons between two algorithms.

In this study, to compare the performance of the proposed CLAOS, AOS, GA, DE, GWO, and PSO algorithms in solving the PEMFC parameter estimation problem, the Friedman test was applied to the results obtained from the 30 independent runs. Figure 7 shows the Friedman scores of all algorithms for Case-1 and Case-2. Besides, the mean score of the two case studies is presented. Accordingly, the CLAOS algorithm obtained the **1.1667** and **1.0000** scores for Case-1 and Case-2, respectively. With these scores, the CLAOS ranked first among all algorithms for two cases. Since CLAOS ranked first in both cases, it naturally ranked first among its competitors with a mean score of **1.0833**. Conversely, the mean score value revealed that the AOS algorithm came in last. These results clearly showed the superior performance of CLAOS over AOS.

Besides the Friedman test, the Wilcoxon test was applied between the CLAOS and its rivals. Table 10 presents the Wilcoxon results for Case-1 and Case-2. Here, a pairwise comparison of results from 30 independent runs yields the X+ value, which represents the total number of runs won by the proposed algorithm. On the other hand, the X- value represents the score that the proposed algorithm lost in the same comparison. According to Table 10, the CLAOS outperformed its competitors for both case studies. Consequently, the statistical analysis results demonstrated the success of the CLAOS algorithm in solving the PEMFC parameter estimation problem.

5. CONCLUSION

This study proposed an improved version of the atomic orbital search algorithm to solve the PEMFC parameter estimation problem. In order to improve the search performance of the AOS algorithm, Lévy flight, and chaotic map functions were integrated into the AOS algorithm. Ten AOS variations were created using the ten chaotic map functions, where the Lévy flight was considered in all variations. A comprehensive simulation study was carried out to test and validate the performance of the proposed algorithm. In the first simulation study, CEC2020 and CEC2022 benchmark problem suites were solved by ten AOS variations and the base AOS algorithm. Their results were analyzed using Friedman and Wilcoxon tests. Based on these results, Case-2, which had the highest Friedman score of 5.3823 among all variations, was identified as the remaining CLAOS algorithm in the study. In the second simulation study, the proposed CLAOS and six comparison algorithms, including AOS, AHA, GA, DE, GWO, and PSO, were applied to solve the PEMFC parameter estimation problem, where a 250W

PEMFC stack was considered. Here, two different search ranges were used for the unknown parameters of the stack. Moreover, in these search ranges, the temperature and anode and cathode pressure values differed in PEMFC specifications. Accordingly, two case studies were created, and their results were compared with the six MHS algorithms and the reported results in the literature. According to the results, the proposed algorithm obtained the smallest SSE values for both cases compared to its competitors. In other words, the proposed algorithm's output models matched the experimental data quite well. On the other hand, the results obtained by the proposed algorithm and its rivals were analyzed using Friedman and Wilcoxon tests. According to the Friedman test results, the proposed CLAOS algorithm ranked first and obtained 1.1667 and 1.0000 scores for Case-1 and Case-2, respectively. Consequently, the proposed algorithm outperformed its rivals in solving benchmark and PEMFC parameter estimation problems. In future work, the AOS variations presented in the study will be applied to solve different real-world optimization problems.

ACKNOWLEDGMENT AND FUNDING

The authors receive no financial support for the research, authorship, and publication of this article.

DECLARATION OF CONFLICTING INTERESTS

The authors declare no potential conflicts of interest with respect to the research and publication of this article

REFERENCES

[1] W. Gong and Z. Cai, Parameter optimization of PEMFC model with improved multi-strategy adaptive differential evolution, *Engineering Applications of Artificial Intelligence*, 27, 2014, 28-40.

[2] H. Rezk, A. G. Olabi, S. Ferahtia and T. E. Sayed, Accurate parameter estimation methodology applied to model proton exchange membrane fuel cell, *Energy*, 255, 2022, 124454.

[3] Y. Chen, D. Pi, B. Wang, J. Chen and Y. Xu, Bi-subgroup optimization algorithm for parameter estimation of a PEMFC model, *Expert Systems with Applications*, 196, 2022, 116646.

[4] S. Xu, Y. Wang and Z. Wang, Parameter estimation of proton exchange membrane fuel cells using eagle strategy based on JAYA algorithm and Nelder-Mead simplex method, *Energy*, 173, 2019, 457-467.

[5] M. Ohenoja and K. Leiviskä, Validation of genetic algorithm results in a fuel cell model, *International Journal of Hydrogen Energy*, 35(22), 2010, 12618-12625.

[6] M. Ali, M. A. El-Hameed and M. A. Farahat, Effective parameters' identification for polymer electrolyte membrane fuel cell models using grey wolf optimizer, *Renewable Energy*, 111, 2017, 455-462.

[7] Fathy and H. Rezk, Multi-verse optimizer for identifying the optimal parameters of PEMFC model, *Energy*, 143, 2018, 634-644.

[8] J. Gupta, P. Nijhawan and S. Ganguli, Optimal parameter estimation of PEM fuel cell using slime mould algorithm, *International Journal of Energy Research*, 45(10), 2021, 14732-14744.

[9] H. Rezk, S. Ferahtia, A. Djeroui, A. Chouder, A. Houari, M. Machmoum and M. A. Abdelkareem, Optimal parameter estimation strategy of PEM fuel cell using gradient-based optimizer, *Energy*, 239, 2022, 122096.

[10] H. Ashraf, S. O. Abdellatif, M. M. Elkholy and A. A. El-Fergany, Honey badger optimizer for extracting the ungiven parameters of PEMFC model: Steady-state assessment, *Energy Conversion and Management*, 258, 2022, 115521.

[11] H. M. Sultan, A. S. Menesy, M. Alqahtani, M. Khalid and A. A. Z. Diab, Accurate parameter identification of proton exchange membrane fuel cell models using different metaheuristic optimization algorithms. *Energy Reports*, 10, 2023, 4824-4848.

[12] L. Xuebin, J. Zhao, Y. Daiwei, Z. Jun and Z. Wenjin, Parameter estimation of PEM fuel cells using metaheuristic algorithms, *Measurement*, 237, 2024, 115302.

[13] Y. Yuan, Q. Yang, J. Ren, X. Mu, Z. Wang, Q. Shen and W. Zhao, Attack-defense strategy assisted osprey optimization algorithm for PEMFC parameters identification, *Renewable Energy*, 225, 2024, 120211.

[14] T. S. Ayyarao, N. Polumahanthi and B. Khan, An accurate parameter estimation of PEM fuel cell using war strategy optimization, *Energy*, 290, 2024, 130235.

[15] Z. Sun, N. Wang, Y. Bi and D. Srinivasan, Parameter identification of PEMFC model based on hybrid adaptive differential evolution algorithm, *Energy*, 90, 2015, 1334-1341.

[16] Z. Yuan, W. Wang and H. Wang, Optimal parameter estimation for PEMFC using modified monarch butterfly optimization, *International Journal of Energy Research*, 44(11), 2020, 8427-8441.

[17] Z. Yang, Q. Liu, L. Zhang, J. Dai and N. Razmjooy, Model parameter estimation of the PEMFCs using improved barnacles mating optimization algorithm, *Energy*, 212, 2020, 118738.

[18] R. M. Rizk-Allah and A. A. El-Fergany, Artificial ecosystem optimizer for parameters identification of proton exchange membrane fuel cells model, *International Journal of Hydrogen Energy*, 46(75), 2021, 37612-37627.

[19] M. Alizadeh and F. Torabi, Precise PEM fuel cell parameter extraction based on a self-consistent model and SCCSA optimization algorithm, *Energy Conversion and Management*, 229, 2021, 113777.

[20] H. M. Hasanien, M. A. Shaheen, R. A. Turky, M. H. Qais, S. Alghuwainem, S. Kamel, M. Tostado-Véliz and F. Jurado, Precise modeling of PEM fuel cell using a novel Enhanced Transient Search Optimization algorithm, *Energy*, 247, 2022, 123530.

[21] H. M. Sultan, A. S. Menesy, M. S. Hassan, F. Jurado and S. Kamel, Standard and quasi oppositional bonobo optimizers for parameter extraction of PEM fuel cell stacks, *Fuel*, 340, 2023, 127586.
[22] F. Duan, F. Song, S. Chen, M. Khayatnezhad and N. Ghadimi, Model parameters identification of the PEMFCs using an improved design of crow search algorithm, *International Journal of Hydrogen Energy*, 47(79), 2022, 33839-33849.
[23] B. Deepanraj, S. K. Gugulothu, R. Ramaraj, M. Arthi and R. Saravanan, Optimal parameter estimation of proton exchange membrane fuel cell using improved red fox optimizer for sustainable energy management, *Journal of Cleaner Production*, 369, 2022, 133385.
[24] S. M. Ebrahimi, S. Hasanzadeh and S. Khatibi, Parameter identification of fuel cell using repairable grey wolf optimization algorithm, *Applied Soft Computing*, 147, 2023, 110791.
[25] J. Zhou, M. A. Ali, K. Sharma, et al., Improved fish migration optimization method to identify PEMFC parameters, *International Journal of Hydrogen Energy*, 48(52), 2023, 20028-20040.
[26] B. Zhang, R. Wang, D. Jiang, Y. Wang, J. Wang and B. Ruan, Parameter identification of proton exchange membrane fuel cell based on swarm intelligence algorithm, *Energy*, 283, 2023, 128935.
[27] M. Ćalasan, M. Micev, H. M. Hasanien and S. H. A. Aleem, PEM fuel cells: Two novel approaches for mathematical modeling and parameter estimation, *Energy*, 290, 2024, 130130.
[28] S. Menesy, H. M. Sultan, M. E. Zayed, I. O. Habiballah, S. Dmitriev, M. Safaraliev and S. Kamel, A modified slime mold algorithm for parameter identification of hydrogen-powered proton exchange membrane fuel cells, *International Journal of Hydrogen Energy*, 86, 2024, 853-874.
[29] K. Priya, V. Selvaraj, N. Ramachandra and N. Rajasekar, Modelling of PEM fuel cell for parameter estimation utilizing clan co-operative based spotted hyena optimizer, *Energy Conversion and Management*, 309, 2024, 118371.
[30] M. H. Elfar, M. Fawzi, A. S. Serry, M. Elsakka, M. Elgamal and A. Refaat, Optimal parameters identification for PEMFC using autonomous groups particle swarm optimization algorithm, *International Journal of Hydrogen Energy*, 69, 2024, 1113-1128.
[31] S. Saidi, S. Marrouchi, B. N. Alhasnawi, P. K. Pathak, O. Alshammari, A. Albaker and R. Abbassi, Precise parameter identification of a PEMFC model using a robust enhanced salp swarm algorithm, *International Journal of Hydrogen Energy*, 71, 2024, 937-951.
[32] H. Alqahtani, H. M. Hasanien, M. Alharbi and S. Chuanyu, Parameters estimation of proton exchange membrane fuel cell model based on an improved Walrus optimization algorithm, *IEEE Access*, 12, 2024, 74979-74992.
[33] Askarzadeh and L. dos Santos Coelho, A backtracking search algorithm combined with Burger's chaotic map for parameter estimation of PEMFC electrochemical model, *International Journal of Hydrogen Energy*, 39(21), 2014, 11165-11174.
[34] Z. Yuan, W. Wang, H. Wang and A. Yildizbasi, Developed coyote optimization algorithm and its application to optimal parameters estimation of PEMFC model, *Energy Reports*, 6, 2020, 1106-1117.
[35] M. T. Özdemir, Optimal parameter estimation of polymer electrolyte membrane fuel cells model with chaos embedded particle swarm optimization, *International Journal of Hydrogen Energy*, 46(30), 2021, 16465-16480.
[36] F. Qin, P. Liu, H. Niu, H. Song and N. Yousefi, Parameter estimation of PEMFC based on improved fluid search optimization algorithm, *Energy Reports*, 6, 2020, 1224-1232.
[37] M. Ćalasan, S. H. A. Aleem, H. M. Hasanien, Z. M. Alaas,and Z. M. Ali, An innovative approach for mathematical modeling and parameter estimation of PEM fuel cells based on iterative Lambert W function, *Energy*, 264, 2023, 126165.
[38] M. Azizi, Atomic orbital search: A novel metaheuristic algorithm, *Applied Mathematical Modelling*, 93, 2021, 657-683.
[39] F. Ali, A. Sarwar, F. I. Bakhsh, S. Ahmad, A. A. Shah and H. Ahmed, Parameter extraction of photovoltaic models using atomic orbital search algorithm on a decent basis for novel accurate RMSE calculation, *Energy Conversion and Management*, 277, 2023, 116613.
[40] M. Azizi, S. Talatahari and A. Giaralis, Optimization of engineering design problems using atomic orbital search algorithm, *IEEE Access*, 9, 2021, 102497-102519.
[41] S. Duman, H. T. Kahraman, U. Guvenc and S. Aras, Development of a Lévy flight and FDB-based coyote optimization algorithm for global optimization and real-world ACOPF problems, *Soft Computing*, 25, 2021, 6577-6617.
[42] S. Saremi, S. Mirjalili and A. Lewis, Biogeography-based optimisation with chaos, *Neural Computing and Applications*, 25, 2014, 1077-1097.
[43] M. Abd Elaziz, S. Ouadfel, A. A. Abd El-Latif and R. Ali Ibrahim, Feature selection based on modified bio-inspired atomic orbital search using arithmetic optimization and opposite-based learning, *Cognitive Computation*, 14(6), 2022, 2274-2295.
[44] G. Manita, A. Chhabra and O. Korbaa, Efficient e-mail spam filtering approach combining logistic regression model and Orthogonal Atomic Orbital Search algorithm, *Applied Soft Computing*, 144, 2023, 110478.
[45] C. T. Yue, K. V. Price, P. N. Suganthan, et al., Problem definitions and evaluation criteria for the CEC 2020 special session and competition on single objective bound constrained numerical optimization, *Zhengzhou Univ. Zhengzhou Technical Report*, 2019, 201911.

[46] Kumar, K. V. Price, A. W. Mohamed, A. A. Hadi and P. N. Suganthan, Problem definitions and evaluation criteria for the CEC 2022 special session and competition on single objective bound constrained numerical optimization, *Nanyang Technol. Univ. Singapore Technical Report*, 2022.

[47] J. H. Holland, Genetic algorithms. *Scientific American*, 267(1), 1992, 66-73.

[48] J. Kennedy and R. Eberhart, Particle swarm optimization, *Proceedings of ICNN'95-International Conference on Neural Networks*, Australia, 1995, 1942-1948.

[49] R. Storn and K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization*, 11, 1997, 341-359.

[50] S. Mirjalili, S. M. Mirjalili and A. Lewis, Grey wolf optimizer, *Advances in Engineering Software*, 69, 2014, 46-61.

[51] W. Zhao, L. Wang and S. Mirjalili, Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications, *Computer Methods in Applied Mechanics and Engineering*, 388, 2022, 114194.

[52] W. Zhang, N. Wang and S. Yang, Hybrid artificial bee colony algorithm for parameter estimation of proton exchange membrane fuel cell, *International Journal of Hydrogen Energy*, 38(14), 2013, 5796-5806.

[53] Y. Rao, Z. Shao, A. H. Ahangarnejad, E. Gholamalizadeh and B. Sobhani, Shark smell optimizer applied to identify the optimal parameters of the proton exchange membrane fuel cell model, *Energy Conversion and Management*, 182, 2019, 1-8.