Heliyon 10 (2024) e36678

Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

An improved nutcracker optimization algorithm for discrete and continuous optimization problems: Design, comprehensive analysis, and engineering applications

Mohamed Abdel-Basset [a], Reda Mohamed [a], Ibrahim M. Hezam [b], Karam M. Sallam [c,d], Ibrahim A. Hameed [e,*]

[a] Faculty of Computers and Informatics, Zagazig University, Zagazig, Sharqiyah, 44519, Egypt
[b] Department of Statistics & Operations Research, College of Sciences, King Saud University, Riyadh, Saudi Arabia
[c] Department of Computer Science, College of Computing and Informatics, University of Sharjah, Sharjah, United Arab Emirates
[d] Faculty of Science and Technology, School of IT and Systems, University of Canberra, ACT, 2601, Australia
[e] Department of ICT and Natural Sciences, Norwegian University of Science and Technology (NTNU), Ålesund, Norway

ARTICLE INFO

Keywords:
Nutcracker optimization algorithm
Exploitation improvement strategy
Proton exchange membrane fuel cell
Photovoltaic models
Image segmentation
Kapur's entropy

ABSTRACT

This study is presented to examine the performance of a newly proposed metaheuristic algorithm within discrete and continuous search spaces. Therefore, the multithresholding image segmentation problem and parameter estimation problem of both the proton exchange membrane fuel cell (PEMFC) and photovoltaic (PV) models, which have different search spaces, are used to test and verify this algorithm. The traditional techniques could not find approximate solutions for those problems in a reasonable amount of time, so researchers have used metaheuristic algorithms to overcome those shortcomings. However, the majority of metaheuristic algorithms still suffer from slow convergence speed and stagnation into local minima problems, which makes them unsuitable for tackling these optimization problems. Therefore, this study proposes an improved nutcracker optimization algorithm (INOA) for better solving those problems in an acceptable amount of time. INOA is based on improving the performance of the standard algorithm using a newly proposed convergence improvement strategy that aims to improve the convergence speed and prevent stagnation in local minima. This algorithm is first applied to estimating the unknown parameters of the single-diode, double-diode, and triple-diode models for a PV module and a solar cell. Second, four PEMFC modules are used to further observe INOA's performance for the continuous optimization challenge. Finally, the performance of INOA is investigated for solving the multi-thresholding image segmentation problem to test its effectiveness in a discrete search space. Several test images with different threshold levels were used to validate its effectiveness, stability, and scalability. Comparison to several rival optimizers using various performance indicators, such as convergence curve, standard deviation, average fitness value, and Wilcoxon rank-sum test, demonstrates that INOA is an effective alternative for solving both discrete and continuous optimization problems. Quantitively, INOA could solve those problems better than the other rival optimizers, with improvement rates for final results ranging between 0.8355 % and 3.34 % for discrete problems and 4.97 % and 99.9 % for continuous problems.

* Corresponding author.
E-mail addresses: mohamedbasset@ieee.org (M. Abdel-Basset), redamoh@zu.edu.eg (R. Mohamed), ialmishnanah@ksu.edu.sa (I.M. Hezam), KSallam@sharjah.ac.ae (K.M. Sallam), ibib@ntnu.no (I.A. Hameed).

https://doi.org/10.1016/j.heliyon.2024.e36678
Received 10 September 2023; Received in revised form 1 August 2024; Accepted 20 August 2024
Available online 22 August 2024

1. Introduction

The optimization problem (OP) involves determining the most favorable solution within a feasible range. Various industries encounter several optimization challenges that necessitate precise optimization to achieve enhanced performance. Generally, optimization problems can be classified into two categories: discrete and continuous. Discrete problems consist of decision variables with integer or binary values, which must be meticulously optimized to obtain a near-optimal solution. In the literature, there are numerous well-established discrete optimization problems, such as the multi-threshold image segmentation problem that is extensively addressed in the literature to extract the optimal threshold values for effectively separating different objects within an image [1,2]. On the other hand, continuous problems involve decision variables that are continuous in nature, as observed in the parameter estimation problems of photovoltaic models and fuel cells, which have been extensively investigated in the literature. It is worth noting that each optimization problem may have single, multiple, or multiple objectives [3].

Over the past two decades, meta-heuristic (MH) optimization algorithms have created a great deal of interest in handling optimization problems in a variety of sectors, particularly the engineering field. This interest is due to their efficient computational cost, high accuracy, and simplicity. According to the inspiration, the MH algorithms are classified into four major groups: physics-based algorithms (i.e. Archimedes optimization algorithm [4], and multiverse optimizer [5]), evolutionary algorithms (i.e. differential evolution (DE) algorithm, evolution strategy, genetic algorithm [6] and evolutionary programming [7]), swarm-based algorithms (i.e. artificial hummingbird algorithm [8] and artificial gorilla troops optimizer [9]), and human-based algorithms (i.e. brain storm optimization [10] and past present future [11]).

The existing metaheuristic algorithms continue to have several flaws, including a lack of population variety, early convergence, and slow convergence speed, which make them unable to achieve approximate solutions for different discrete and continuous optimization problems. Therefore, this paper presents a robust metaheuristic-based optimization technique with better exploitation and exploration operators to alleviate these challenges when applied to three well-known optimization problems: multilevel image segmentation, parameter estimation of fuel cells, and parameter estimation of photovoltaic (PV) models. This algorithm is based on improving the search capability of the recently published nutcracker optimization algorithm (NOA) using a newly proposed convergence improvement strategy that could assist in boosting the convergence speed and preventing stagnation into in local minima. This improved variant of NOA is abbreviated as INOA. Both NOA and INOA are first assessed using a PV module and a solar cell under the single-diode model (SDM), double-diode model (DDM), and third-diode model (TDM) to observe their performance for estimating the unknown parameters of the PV models. Then, they are applied to find the unknown parameters of the PEMFC model for four commercial PEMFC stacks. Finally, their performance is tested and verified using several test images with various threshold levels to investigate their robustness for the image segmentation problem. To show the effectiveness of INOA and NOA for these problems, they are compared to several rival optimizers in terms of several performance metrics, such as best fitness, average fitness, worst fitness, standard deviation, and convergence curve. In addition, the Wilcoxon rank-sum test is used in comparison to reveal the difference between the INOA's outcomes and each rival optimizer's outcomes. The experimental findings show that INOA is a strong alternative for tackling discrete and continuous optimization problems. The main contributions of this study are as follows.

- Proposing an improved variant of NOA, namely INOA, for solving the parameter estimation problems of both PEMFC and PV models as well as the multilevel thresholding image segmentation problem. This variant has outstanding performance in terms of convergence speed and final outcome accuracy.
- Assessing INOA using a PV module and a solar cell under the SDM, DDM, and TDM to observe its accuracy for determining the values of the unknown parameters of the PV models.
- Using four commercial PEMFC stacks to study the performance of the proposed INOA for estimating the parameters of the PEMFC model.
- Using several test images with various threshold levels to investigate its robustness for the image segmentation problem.
- In comparison to several rival optimizers, INOA is the best-performing optimizer for all optimization problems tackled in this study.

The remaining sections of the paper are consolidated as Section 3 presents the literature review. The mathematical model of the PV modules, PEMFC, and image segmentation problem are presented in section 4. Section 5 presents the details of the standard nutcracker optimization algorithm, while the detailed steps of the proposed algorithm are presented in Section 6. The results and discussions are presented in section 7. Finally, Section 8 presents the conclusion and future work of this study.

2. Literature review

2.1. Proton exchange membrane fuel cell (PEMFC)

In the last several decades, the proton exchange membrane fuel cell (PEMFC) has gained international attention as a significant renewable energy source due to its ability to constantly and immediately convert hydrogen fuel into electrical energy for the disposal of environmental pollution caused by conventional energy sources [12,13]. PEMFC modeling is necessary for simulating and designing PEMFC systems more accurately. In brief, PEMFC modeling requires correctly modeling the polarization curves, which illustrate the relationship between current and voltage. Several methods, including metaheuristic algorithms and empirical or semi-empirical-based approaches [12,14–18], have been utilized to build models of PEMFCs. For engineering applications, the semi-empirical model is

preferable because it is well-known and easier to solve. However, mathematical models based on semi-empirical equations include certain unknown parameters that require accurate estimation. Consequently, this issue has been formulated as an optimization problem, referred to as the parameter estimation problem of PEMFC in the literature, and has been addressed using various optimization techniques [19]. This problem is a complex nonlinear challenge with multiple local optima, and thereby the traditional optimization techniques are not able to well handle this challengeable problem [19]. Therefore, the metaheuristic algorithms that have robust characteristics for avoiding stagnation into local minima have been applied in the literature to overcome this challenge. Some of these algorithms are reviewed in the rest of this section.

Sarhan [20] introduced an efficient adaptation of the atomic search optimizer (ASO) for this task. In a similar vein [19], integrated the Nelder-Mead simplex method (NMSM) and JAYA algorithm to boost the convergence and avoid getting stuck into local optima during parameter estimation in PEMFCs. Additionally, the enhanced cuckoo search (ECS) algorithm [21] was developed to mitigate the risk of getting trapped in local minima. This algorithm incorporated an explosive operator to enhance its performance. In another study [22], proposed a novel hybrid grey wolf optimizer (HGWO) approach for addressing the parameter estimation problem. The HGWO method combined the classical grey wolf optimizer (GWO) with crossover and mutation operators to enhance searchability and prevent the occurrence of local optima. More recently, the political optimizer (PO) and marine predators algorithm (MPA) have been applied to the parameter estimation problem, with MPA showing promising results [23]. Furthermore, Abbassi et al. [24] adapted the dandelion optimizer for estimating unknown parameters in PEMFCs and evaluated its performance under both dynamic and steady-state conditions.

Zhang et al. [25] hybridized the artificial bee colony algorithm using the crossover operator of genetic algorithms and the mutation operator of DE to enhance its exploration and exploitation capabilities for boosting the convergence speed and avoiding falling into local minima. This improved algorithm was called IABC and used in estimating the unknown parameters of the PEMFC model. Sultan et al. [26] proposed a modified variant of the bonobo optimizer, namely the quasi-opposite BO (QOBO). This variant was used to identify the unknown parameters of this problem. There are several other MH algorithms proposed recently for tackling the parameter estimation of PEMFCs, such as modified monarch butterfly optimization [27], improved fluid search optimization algorithm [28], improved barnacles mating optimization algorithm [29], ranking-based gorilla troops optimizer [30], repairable grey wolf optimization algorithm [31], improved gorilla troops technique [32], white shark optimizer [33], improved remora optimizer [34], red-tailed hawk algorithm [35], and gradient-based optimizer [36].

To prove our proposition presented in the introduction section, which says that the existing metaheuristic algorithms suffer from either falling into local minima, slow convergence speed, or time-consuming problems, in the experiments section, the performance of several metaheuristic algorithms, which have been proposed over the last few years for this problem, is analyzed in comparison to INOA. Those algorithms include the interior search algorithm (ISA) [37], gradient-based optimizer (GBO) [38], differential evolution (DE) [22], marine predators algorithm (MPA) [23], manta ray foraging optimizer (MRFO) [39], equilibrium optimizer (EO), moth-flame optimizer (MFO) [39], salp swarm algorithm (SSA) [40], and improved horse herd optimization algorithm (IHOA) [38]. This analysis shows that some algorithms, such as SSA, MPA, MFO, and DE, suffer from falling into local minima, thereby making them unsuitable for searching for the approximate solutions to this problem, and the other algorithms suffer from slow convergence speed, thereby consuming a huge number of function evaluations for achieving the desired outcomes.

2.2. Solar photovoltaic

Solar photovoltaic (PV) technology stands as a significant renewable energy source, harnessing the photovoltaic effect to generate electricity. By leveraging the natural electrical phenomenon of certain materials known as semiconductors, solar cells or diodes convert sunlight into useable energy. The absorption properties of these semiconductors dictate the extent to which they absorb light across the solar spectrum. Through this absorption process, ample energy is generated to liberate negatively charged electrons. The conversion of solar energy through photovoltaic (PV) devices is crucial in providing an economically viable and sustainable alternative to traditional power sources [41]. The successful implementation of PV systems relies heavily on the ability to design an optimal configuration that fulfills specific requirements while minimizing costs [42]. To achieve this, precise modeling, formulation, and simulation of PV systems are necessary. A PV model is employed to represent the current-voltage (I-V) and power-voltage (P-V) characteristics of the PV system under different operating conditions [43]. However, accurately estimating the parameters of a solar cell, which are not typically provided in the manufacturer's specifications, presents a challenging optimization problem. The chosen parameter values directly influence the amount of electricity generated [4]. In addressing this issue, various metaheuristic (MH) algorithms have been proposed, and the subsequent sections will discuss some of these approaches.

Qaraad et al. [44] introduced a method called QPSOL, which combines particle swarm optimization (PSO) with a local search strategy. This integration helps mitigate issues related to local minima while enhancing convergence speed. They applied QPSOL to determine the unknown parameters of three PV models. Belabbes et al. [45] proposed an improved variant of the snake optimizer (SO) specifically designed for determining the unknown parameters in both SDM and DDM. They applied this variant to effectively estimate the unknown parameters in these models. Lakshmanan [46] made modifications to the force-control parameters of the white shark optimizer (WSO) to enhance its exploration operator. Additionally, chaotic maps were employed to improve its exploitation operator. This enhanced variant, named IWSO, was utilized to identify unknown values of five and seven parameters in SDM and DDM. The black widow optimization algorithm (BWOA) was employed to tackle the parameter estimation problem in two PV models [47]. This algorithm aims to optimize the parameter values to achieve accurate estimation results. Abdel-Basset et al. [48] utilized the recently published Kepler optimization algorithm to estimate the unknown parameters of the SDM, DDM, and the three-diode model (TDM). This algorithm was specifically applied for parameter estimation in PV models.

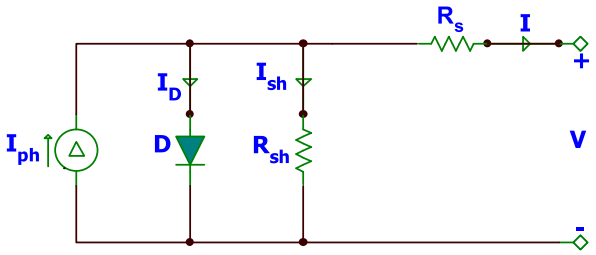


Fig. 1. SDM equivalent circuit.

There are several other metaheuristic algorithms proposed to tackle this problem, including the tree seed algorithm [49], enhanced leader particle swarm optimization [50], chaotic tunicate swarm algorithm [51], war strategy optimization algorithm [52], hybrid seagull optimization algorithm [53], artificial hummingbird optimization algorithm [54], grey wolf optimizer [55], spider wasp optimizer [56], Ali baba and forty thieves optimization technique [57], improved moth flame algorithms [58], and improved marine predators algorithm [59]. Each of these algorithms offers unique approaches and characteristics to address the parameter estimation challenge in PV models.

The metaheuristic algorithms proposed for this problem suffer from at least one of the following shortcomings: trap into local optima, slow convergence speed, and high computational cost. For example, in Ref. [60], Light Spectrum Optimizer was applied to estimate the unknown parameters of TDM, but it suffers from falling into local optima that prevents it from achieving better outcomes for this problem. This is deduced based on our experimental outcomes discussed later, where our proposed INOA manages to achieve superior outcomes in comparison to this algorithm. Likewise, in Ref. [61], the marine predators algorithm (MPA), which is inferior to the proposed algorithm according to the experimental results discussed later, was applied to tackle this problem. Also, the proposed INOA is compared to several other metaheuristic algorithms, such as the grey wolf optimizer [62], slime mould algorithm [63], and dandelion optimizer [64], which were adapted for solving this problem. The experimental outcomes show that those algorithms suffer from falling into local minima, which makes them unsuitable for this problem.

2.3. Image segmentation problem

Image segmentation is the extraction process of various objects from an image. It is required in image processing [65] and computer vision [66] to aid with the analysis and comprehension of the images. Threshold segmentation is one of the most common techniques used to segment the images, which can be classified into a bi-level threshold or a multi-level threshold. In the bi-level threshold, image objects can be divided into two categories: foreground (object) and background. The multi-level threshold divides the image objects into more than two classes. Threshold segmentation is straightforward, precise, quick, and space-efficient. Unfortunately, the multi-level threshold exponentially increases the time complexity. Utilized threshold strategies utilize parametric and non-parametric methodologies to determine the appropriate threshold values [67]. In the parametric method, the parameters for each image class are determined using a probability density function. The non-parametric method determines the threshold values by optimizing some functions such as fuzzy entropy [68], Kapur's entropy [69], and Otsu method [70,71] without utilizing statistical factors. Several MH techniques in the literature have been applied to optimize those functions for finding the near-optimal threshold values that could accurately segment the images. Some of these techniques are the crow search algorithm [72], improved firefly algorithm [73], moth swarm algorithm [74], multi-verse optimizer [5], water cycle algorithm [75], opposition-based Laplacian equilibrium optimizer [76], modified grasshopper algorithm [77], krill herd optimization algorithm [78], manta ray foraging optimizer [79], and growth optimizer [80].

In this study, we try to present a strong algorithm that could accurately find the near-optimal threshold values for various threshold levels, especially high threshold levels. The majority of the recently proposed metaheuristic algorithms were assessed and verified using low threshold levels, up to 7, which are unsuitable for evaluating the scalability and effectiveness of those algorithms. For example, in Ref. [81], the improved heap-based optimizer was proposed for solving this problem and assessed using low threshold levels, including 2, 3, 4, and 5. In Ref. [82], the modified reptile search algorithm was presented to segment the MRI images under threshold levels up to 10. Those levels are not enough to reveal its effectiveness for the images that require higher threshold levels. Yang et al. [83] proposed an improved variant of DE for accurately segmenting breast cancer images. This variant was evaluated using different threshold levels up to 6, which are too low to show its effectiveness and efficiency.

3. Mathematical formulation

In this paper, two continuous optimization problems, namely parameter estimation of the PEMFC and PV model, and another discrete optimization problem, namely multilevel thresholding image segmentation, will be solved by the proposed algorithm to observe its effectiveness.

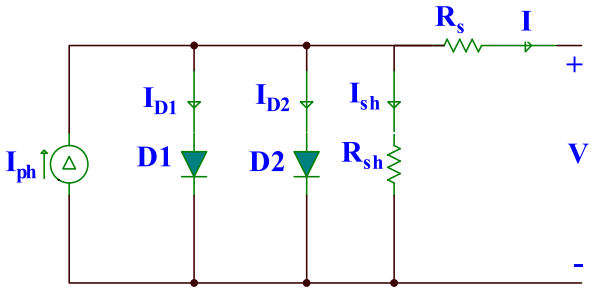


Fig. 2. DDM electrical circuit.

3.1. Photovoltaic models and modules

Firstly, in this section, we introduce the mathematical model of the photovoltaic SDM, DDM, and TDM. Following that, we discuss the objective function utilized in conjunction with the proposed algorithm to direct the search during the entire optimization process.

3.1.1. Mathematical model of single-diode model

The SDM's electrical circuit is shown in Fig. 1. Eq. (1) describes how to derive the SDM output symbolized as I.

$$I = I_{ph} - I_D - I_{sh} \tag{1}$$

where I_{ph} stands for photo-generated current [84]. The current through a diode, denoted by the symbol I_D, can be calculated using Eq. (2).

$$I_D = I_{sd}\left(exp\left(\frac{V + I * R_s}{n * V_t}\right) - 1\right) \tag{2}$$

where I_{sd} is the diode's reverse saturation current; V is the diode's output voltage; R s is the series resistance; n is the diode's ideality factor; and V_t is calculated by Eq. (3).

$$V_t = \frac{k * T}{q} \tag{3}$$

where T represents the junction temperature in Kelvin, k represents the Boltzmann constant ($1.3806503 \times 10^{-23}$ J/K), and q represents the electron charge ($1.60217646 \times 10^{-19}$ C). Eq. (4) can be used to determine the shunt resistor current, I_{sh}.

$$I_{sh} = \frac{V + I * R_s}{R_{sh}} \tag{4}$$

In which R_{sh} is the shunt resistance. By Substitution, I can be computed according to Eq. (5).

$$I = I_{ph} - I_{sd}\left(exp\left(\frac{q * (V + I * R_s)}{n * k * T}\right) - 1\right) - \frac{V + I * R_s}{R_{sh}} \tag{5}$$

The above discussion suggests that five unknown parameters ($I_{ph}, I_{sd}, n, R_s, and\ R_{sh}$) must be precisely estimated to run an accurate simulation of the SDM. Stochastic methods, such as the proposed algorithm, can be used to solve this optimization problem.

3.1.2. Mathematical model of double-diode model

The double diode (DD) model is meant to be a powerful alternative to the single diode (SDM) model in situations where the SDM is not optimal, such as when the irradiance level is low [85]. Fig. 2 depicts the DDM, which consists of two diodes, one of which acts as a rectifier and the other of which is used to compensate for the influence of non-idealities in the SC and the effect of current resulting from recombination. Eq. (6) is the formula used to determine DDM output.

$$I = I_{ph} - I_{D1} - I_{D2} - I_{sh} \tag{6}$$

I can be substituted with Eq. (7).

$$I = I_{ph} - I_{sd1}\left(exp\left(\frac{V + I * R_s}{n_1 * V_t}\right) - 1\right) - I_{sd2}\left(exp\left(\frac{V + I * R_s}{n_2 * V_t}\right) - 1\right) - \frac{V + I * R_s}{R_{sh}} \tag{7}$$

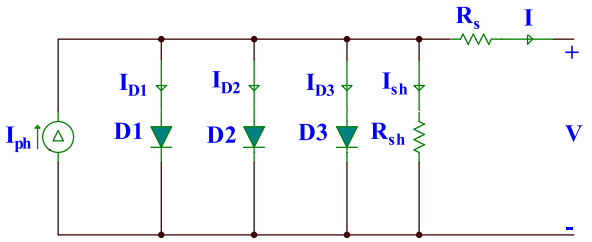


Fig. 3. TDM electrical circuit.

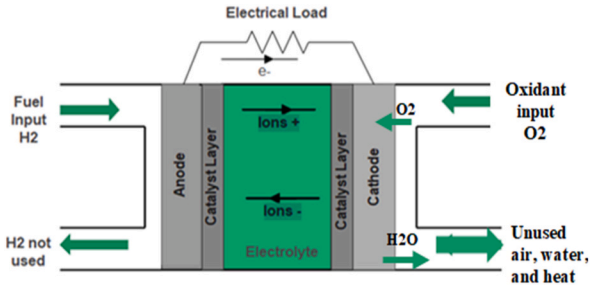


Fig. 4. The structure of the PEMFC.

where I_{sd1} is the first diode's current and I_{sd2} is the second diode's current. n_1 and n_2 are the diodes' ideality factors. In this formula, seven parameters, I_{ph}, I_{sd1}, I_{sd2}, R_s, R_{sh}, n_1, *and* n_2 are absent from the manufacturing sheet. Without these variables, it is impossible to precisely recreate the DDM. Consequently, this paper attempts to estimate the near-optimal values for these parameters using the proposed technique.

3.1.3. Mathematical model of triple diode model (TDM)

The TDM depicted in Fig. 3 consists of a photocurrent source, I_{ph}, three diodes in parallel, a shunt resistor R_{sh}, and a series of resistance R_s. According to Eq. (8), the TDM's output current could be computed.

$$I = I_{ph} - \sum_{i}^{3} I_{D,i} - I_{sh} \tag{8}$$

Where $I_{D,i}$ represents the current in the *i-th* diode, I_{sh} represents the shunt current in R_{sh}, and I represent the current in R_s. $I_{D,i}$ and I_{sh} can be determined using Eq. (9).

$$I_{D,i} = I_{sdi}\left(e^{\frac{q(V+IR_s)}{N_s a_i KT}} - 1\right) \forall\, i \in 1:3, I_{sh} = \frac{V + IR_s}{R_{sh}} \tag{9}$$

where I_{sdi} is the reverse saturation current of the ith diode, a_i is the ideality factor of the ith diode, and V is the output voltage of the solar cell. As a substitute, I can be represented according to Eq. (10).

$$I = I_{ph} - I_{sd1}\left(e^{\frac{q(V+IR_s)}{N_s a_1 KT}} - 1\right) - I_{sd2}\left(e^{\frac{q(V+IR_s)}{N_s a_2 KT}} - 1\right) - I_{sd3}\left(e^{\frac{q(V+IR_s)}{N_s a_3 KT}} - 1\right) - \frac{(V + IR_s)}{R_{sh}} \tag{10}$$

On the contrary, this equation involves nine unknown parameters: $I_{ph}, I_{sd1}, I_{sd2}, I_{sd3}, R_s, R_{sh}, a_1, a_2,$ and a_3. Accurately and efficiently estimating these parameters is crucial to enhance the performance of the PV model. The main goal of this study is to identify and extract these parameters using the proposed algorithm, guided by the objective function (OF) which will be described in the subsequent section.

3.1.4. Objective function (OF)

The fundamental objective of addressing the parameter extraction problem is to identify the unknown parameter values that result in the smallest amount of discrepancy between the measured and simulated current data. Consequently, the root mean squared error (RMSE), referred to as the objective function (OF), is used to compute the difference between observed and estimated data based on Eq. (11).

$$RMSE = f\left(\vec{X}_i\right) = \sqrt{\frac{1}{M} * \sum_{k=1}^{M} \left(I_m - I_e\left(V_e, \vec{X}_i\right)\right)^2} \tag{11}$$

where I_e denotes the estimated current, and I_m denotes the measured current. The measured data's length is indicated by M. The parameters of the ith solution are shown in $\vec{X}_i$. I_e can be estimated by solving Eqs. (4), (6) and (9) using the Newton-Raphson method and the extracted parameters shown in X_i, as defined in Eq. (12) [86].

$$I_e = I - \frac{I}{I'} \tag{12}$$

where I' refers to the first derivative of I.

3.2. Proton exchange membrane fuel cell (PEMFC)

A PEMFC, as shown in Fig. 4, consists of a cathode and an anode separated by an electrolyte [87]. At the anode, H_2 is split into H^+ ions and e^- by a catalyst layer, as described in eq. (13). Then, a circuit external to the cell supplies the cathode with e^-, and the electrolyte supplies it with H^+. After that, the oxygen in the cathode's catalyst layer combines with the electrons and protons to produce water and energy, as shown in eq. (14). The final formula representing all of the reactions is shown in Eq. (15) [88,89]:

$$\text{Anode} : H_2 \rightarrow 2H^+ + 2e^- \tag{13}$$

$$\text{Cathode} : \frac{1}{2}O_2 + 2H^+ + 2e^- \rightarrow H_2O \tag{14}$$

$$\text{Overall reaction} : 2H_2 + O_2 \rightarrow 2H_2O + Heat + Electricity \tag{15}$$

To generate a substantial amount of energy, a single fuel cell's terminal voltage is typically insufficient. Therefore, N_{cells} of fuel cells need to be connected in series to achieve a higher voltage. Eq. (15) illustrates this relationship, which determines the total voltage produced by PEMFCs [90–93].

$$V_s = N_{cells}(E - \nu_{act} - \nu_R - \nu_c) \tag{16}$$

Eq. (17) provides a means to calculate the open-circuit voltage (E) for each cell. This voltage represents the potential difference across the terminals of the cell when no current is flowing. Eq. (21) can be utilized to determine the activation over-potential (ν_{act}). This overpotential arises from the energy barriers involved in initiating an electrochemical reaction within the cell. The voltage drop that occurs when protons move across the electrolyte barrier and electrons traverse from the anode to the cathode through the external circuit is known as the ohmic voltage (ν_R). This voltage can be calculated using eq. (23). Furthermore, Eq. (25) can be employed to calculate the concentration over-potential (ν_c). This overpotential arises from the concentration gradient of reactants or products involved in the electrochemical reaction.

$$E = 1.229 - 0.85 * 10^{-3}\left(T_{fc} - 298.15\right) + 4.3085 \times 10^{-5} T_{fc}\, ln\left(P_{H_2}\sqrt{Po_2}\right) \tag{17}$$

$$P_{H_2} = \frac{(RH_a * P_{H2O})}{2}\left[1 \Big/ \left(\frac{(RH_a \times P_{H2O})}{P_a} e^{\frac{\left(4.192 I_{fc}/A\right)}{T_{fc}^{1.334}}}\right) - 1\right] \tag{18}$$

$$P_{O_2} = (RH_c * P_{H2O})\left[1 \Big/ \left(\frac{(RH_a \times P_{H2O})}{P_c} e^{\frac{\left(4.192 I_{fc}/A\right)}{T_{fc}^{1.334}}}\right) - 1\right] \tag{19}$$

$$P_{H2O} = 2.95 * 10^{-2} T_c - 9.18 * 10^{-5} T_c^2 + 1.44 * 10^{-7} T_c^3 - 2.18 \;\Big|\; T_c = T_{fc} - 273.15 \tag{20}$$

$$\nu_{act} = -\left[\xi_1 + \xi_2 T_{fc} + \xi_3 T_{fc}\, ln\left(C_{O_2}\right) + \xi_4 T_{fc}\, ln\left(I_{fc}\right)\right] \tag{21}$$

$$C_{O_2} = \frac{P_{O_2}}{5.08 \times 10^6} e^{\frac{498}{T_{fc}}} \tag{22}$$

$$V_R = I_{fc}(R_m + R_c) \;\Big|\; R_m = \frac{\rho_m^l}{A} \tag{23}$$

$$\rho_m = \frac{181.6\left[1 + 0.03\left(\frac{I_{fc}}{A}\right) + 0.062\left(\frac{I_{fc}}{303}\right)^2\left(\frac{I_{fc}}{A}\right)^{2.5}\right]}{\left[\lambda - 0.634 - 3\left(\frac{I_{fc}}{A}\right)\right]e^{4.18\left(\frac{T_{fc}-303}{T_{fc}}\right)}} \tag{24}$$

$$v_c = -\beta \ln\left(\frac{J_{max} - J}{J_{max}}\right) \tag{25}$$

where Po_2, P_{H_2} and P_{H2O} are, respectively, the partial pressures of oxygen (O_2), hydrogen (H_2), and water ($H2O$). The current gas's temperature, T_{fc}, is expressed in Kelvin. RH_a and RH_c stands for the relative humidity of vapor at the anode and cathode, respectively. P_a and P_c, respectively, stand for the inputs for the anode and cathode pressures (measured in atm). The oxygen concentration is represented by the symbol C_{O_2} and is measured in mol/cm3. R_m, and R_c, respectively, stand for the membrane resistance and the connection resistance. I_{fc} stands for the operating current of the FC. A parametric coefficient is β. The membrane is designated as having a thickness of l (CM). ρ_m stands for the membrane's resistance. J_{max} stands for maximum current density, and J stands for real current density. Following a description of the mathematical model of the PEMFC, it was evident that the model would require accurate calculation of seven unknown parameters (ξ_1, ξ_2, ξ_3, ξ_4, λ, R_c, and β). Since this topic is framed as an optimization problem, these parameters can be calculated using an optimization technique.

3.2.1. PEMFC's objective function

The quality of each solution provided by the proposed technique is determined by comparing the optimal voltages estimated experimentally with the estimated voltages using the extracted parameters. Based on that, the objective function is represented by the sum of the squared error (SSE) rates between the measured and estimated data [75,94,95]. The mathematical model of this function is shown in Eq. (26) [91].

$$SSE = f\left(\overrightarrow{X_i}\right) = \sum_{c=1}^{LD} \left|V_{S,measured}(c) - V_{S,estimated}(c)\right|^2 \tag{26}$$

where LD is the number of the measured points.

3.3. Image segmentation problem: Kapur's entropy

This section will describe Kapur's entropy, which is utilized to obtain appropriate threshold values by maximizing the entropy of segmented regions [69]. Assuming that the threshold values that segment an image into k comparable regions are $t_0, t_1, t_2, \ldots\ldots\ldots$,and t_k, then Kapur's entropy will maximize Eq. (27) until the optimal values for these thresholds are reached.

$$T\,(t_0, t_1, t_2, \ldots\ldots\ldots, t_k) = T_0 + T_1 + T_2 + \ldots\ldots\ldots + T_k \tag{27}$$

where:

$$T_0 = -\sum_{i=0}^{t_0-1} \frac{X_i}{W_0} * \ln\frac{X_i}{W_0},\; X_i = \frac{N_i}{W}, W_0 = \sum_{i=0}^{t_1-1} X_i \tag{28}$$

$$T_1 = -\sum_{i=t_0}^{t_1-1} \frac{X_i}{W_1} * \ln\frac{X_i}{W_1}, X_i = \frac{N_i}{W}, W_1 = \sum_{i=t_0}^{t_1-1} X_i \tag{29}$$

$$T_2 = -\sum_{i=t_1}^{t_2-1} \frac{X_i}{W_2} * \ln\frac{X_i}{W_2},\; X_i = \frac{N_i}{W}, W_2 = \sum_{i=t_1}^{t_2-1} X_i \tag{30}$$

$$T_k = -\sum_{i=t_k}^{L-1} \frac{X_i}{W_k} * \ln\frac{X_i}{W_k},\; X_i = \frac{N_i}{W}, W_k = \sum_{i=t_k}^{L-1} X_i \tag{31}$$

$T_0, T_1, T_2, \ldots\ldots\ldots$, *and* T_k represent the entropies of similar regions, while N_i is the number of pixels with the same value as i. W_0, W_1, $W_2, \ldots$, *and* W_k describe the probability of the various regions relative to the whole pixel W in an image. In our proposal, Eq. (26) is employed as a maximization fitness function to determine the ideal threshold values.

4. Nutcracker optimization algorithm (NOA)

Recently, the Nutcracker Optimization Algorithm (NOA) was introduced to address continuous optimization problems [96]. NOA mimics nutcrackers' foraging, storing, and retrieving of pine seeds. There are two basic types of nutcracker behavior, and they both happen at distinct times. First, during the warmer months, the nutcracker is likely collecting seeds to preserve for the colder months. Searching for hidden caches at different angles utilizing a variety of objects or markings as guides is another strategy that can be employed during the colder months of winter and spring. In the absence of their usual food source—the stored seeds—the nutcrackers will scavenge for food by randomly probing the search area. To formalize NOA's foraging and storage strategy into a mathematical model, we can define the following steps for the foraging stage.

4.1. Foraging and storage strategy

This strategy can be divided into its two basic phases, which are referred to as foraging and storage, and are described in greater detail below.

- **Foraging stage: Exploration phase 1**

In this phase, the nutcrackers exhibit a randomized placement behavior within the search space. Each nutcracker initiates by evaluating the initial position of a cone filled with seeds. If viable seeds are detected, the nutcracker proceeds to relocate and bury them within a designated storage area. Conversely, if no seeds are found, the nutcracker engages in a search for an alternative cone positioned among various trees, including pine trees. To capture this behavior mathematically, Eq. (32) can be employed.

$$\overrightarrow{X}_i^{t+1} = \begin{cases} X_{i,j}^t \quad if\ \tau_1 < \tau_2 \\ \begin{cases} X_m^t + \gamma \bullet \left(X_{A,j}^t - X_{B,j}^t\right) + \mu.\left(r^2.U_j - L_j\right), if\ t \le T_{max} \Big/ 2.0 \\ X_{C,j}^t + \mu \bullet \left(X_{A,j}^t - X_{B,j}^t\right) + \mu.(r_1 < \delta).\left(r^2.U_j - L_j\right), Otherwise \end{cases} otherwise \end{cases} \tag{32}$$

where $\overrightarrow{X}_i^{t+1}$ represents the new position of the ith nutcracker at iteration $t+1$; $X_{i,j}^t$ is the *jth* variable of the ith nutcracker at iteration t; U_j and L_j are the upper bound and lower bound of the *jth* dimension, respectively; γ is a random number based on the levy-flight; $X_{best,j}^t$ is the *jth* dimension of the best solution obtained so far; *A, C,* and *B* represent the indices of three nutcrackers chosen randomly from the current population; τ_1, τ_2, r, and r_1 are numerical values generated randomly between 0 and 1; $X_{m,j}^t$ represents the mean of the *jth* dimensions for all solutions at iteration t; and μ is a random number generated according to Eq. (33).

$$\mu = \begin{cases} \tau_3 \quad if\ r_1 < r_2 \\ \tau_4 \quad if\ r_2 < r_3 \\ \tau_5 \quad if\ r_1 < r_3 \end{cases} \tag{33}$$

where r_2, r_3, and τ_4 are numbers generated randomly between 0 and 1. τ_4 is a number generated randomly using normal distribution, and τ_5 is a random number generated using the levy-flight (τ_5).

- **Storage stage: Exploitation phase 1**

Nutcrackers begin by transporting the food gathered in the previous exploration phase to temporary storage locations. During this initial phase of exploitation, the nutcrackers collect and store pine seed crops. Eq. (34) presents the mathematical model for this behavior.

$$\overrightarrow{X}_i^{t+1(new)} = \begin{cases} \overrightarrow{X}_i^t + \mu \bullet \left(\overrightarrow{X}_{best}^t - \overrightarrow{X}_i^t\right) \bullet |\lambda| + r_1 \bullet \left(\overrightarrow{X}_A^t - \overrightarrow{X}_B^t\right)\ if\ \tau_1 < \tau_2 \\ \overrightarrow{X}_{best}^t + \mu \bullet \left(\overrightarrow{X}_A^t - \overrightarrow{X}_B^t\right)\ if\ \tau_1 < \tau_3 \\ \overrightarrow{X}_{best}^t \bullet l \quad Otherwise \end{cases} \tag{34}$$

where λ is a number created randomly according to the lévy flight. *l* is a factor decreasing linearly from 1 to 0. During the optimization process, Eq. (35) shows how to tradeoff between the foraging phase and the cache to maintain the balance between exploration and exploitation operators.

$$\overrightarrow{X}_i^{t+1} = \begin{cases} \text{Eq.}(32), if \ \ \varphi < P_{a_1} \\ \text{Eq.}(34), otherwise \end{cases} \tag{35}$$

where φ is a random number between 0 and 1, and P_{a_1} is a linearly decreasing probability from 1 to 0.

4.2. Cache-search and recovery strategy

This mechanism can be divided into two phases: The cache-search and recovery phases are covered in detail in the next two sections.

- **Cache-search stage: Exploration phase 2**

In NOA, every nutcracker employs two reference points (RPs) for each cache as marks, as defined by Eq. (36).

$$\text{RPs} = \begin{bmatrix} \overrightarrow{RP}_{1,1}^{t} & \overrightarrow{RP}_{1,2}^{t} \\ \vdots & \vdots \\ \overrightarrow{RP}_{i,1}^{t} & \overrightarrow{RP}_{i,2}^{t} \\ \vdots & \vdots \\ \overrightarrow{RP}_{N,1}^{t} & \overrightarrow{RP}_{N,1}^{t} \end{bmatrix} \tag{36}$$

where $\overrightarrow{RP}_{i,1}^{t}$ represents the first RPs for the ith nutcracker at iteration t. Two distinct equations were designed to produce the first and second RPs, respectively, to improve the nutcracker exploring process when searching for hidden caches. First RP can be created using Eq. (37), and second RP may be calculated using Eq. (38).

$$\overrightarrow{RP}_{i,1}^{t} = \begin{cases} \overrightarrow{X}_i^t + \alpha \bullet \cos(\theta) \bullet \left(\left(\overrightarrow{X}_A^t - \overrightarrow{X}_B^t \right) \right) + \alpha \bullet RP, if \ \theta = \pi/2 \\ \overrightarrow{X}_i^t + \alpha \bullet \cos(\theta) \bullet \left(\left(\overrightarrow{X}_A^t - \overrightarrow{X}_B^t \right) \right), otherwise \end{cases} \tag{37}$$

$$\overrightarrow{RP}_{i,2}^{t} = \begin{cases} \overrightarrow{X}_i^t + \left(\alpha \bullet cos(\theta) \bullet \left(\left(\overrightarrow{U} - \overrightarrow{L} \right) \bullet \tau_3 + \overrightarrow{L} \right) + \alpha \bullet RP \right) \bullet \overrightarrow{U_2}, if \ \theta = \pi/2 \\ \overrightarrow{X}_i^t + \alpha \bullet cos\ (\theta) \bullet \left(\left(\overrightarrow{U} - \overrightarrow{L} \right) \bullet \tau_3 + \overrightarrow{L} \right) \bullet \overrightarrow{U_2}, otherwise \end{cases} \tag{38}$$

$$\overrightarrow{U_1} = \begin{cases} 1 \ \overrightarrow{r_2} < P_{rp} \\ 0 \quad otherwise \end{cases} \tag{39}$$

where $\overrightarrow{r_2}$ is a vector generated randomly between 0 and 1; $\overrightarrow{X}_A^t$ is a randomly selected solution from the current population; θ is a numerical value generated randomly at the interval of 0 and π; and P_{rp} is a probability to determine the exploration rate; and α is computed using Eq. (40).

$$\alpha = \begin{cases} \left(1 - \frac{t}{T_{max}} \right)^{2\frac{t}{T_{max}}}, if \ r_1 > r_2 \\ \left(\frac{t}{T_{max}} \right)^{\frac{2}{t}}, otherwise \end{cases} \tag{40}$$

where T_{max} represents the maximum number of generations and t is the current generation. Eq. (41) is utilized in NOA to activate the spatial memory for the first RP.

$$\overrightarrow{X}_i^{t+1} = \begin{cases} \overrightarrow{X}_i^t, if\, f\left(\overrightarrow{X}_i^t\right) < if\left(\overrightarrow{RP}_{i,1}^t\right) \\ \overrightarrow{RP}_{i,1}^t, otherwise \end{cases} \tag{41}$$

- **Recovery stage: Exploitation phase 2**

The first scenario is that a nutcracker can recall the cache's location using the first RP. This behavior is represented by Eq. (42).

$$X_{ij}^{t+1} = \begin{cases} X_{ij}^t, if\ \tau_3 < \tau_4 \\ X_{ij}^t + r_1 \bullet \left(X_{best,j}^t - X_{ij}^t\right) + r_2 \bullet \left(\overrightarrow{RP}_{i,1}^t - X_{Cj}^t\right), otherwise \end{cases} \tag{42}$$

where r_1, r_2, τ_3 and τ_4 are numbers generated randomly between 0 and 1, and C represents the index of a solution selected randomly from the current population. The second scenario is that the nutcracker forgets where he concealed his food with the first RP, so he will use the second RP to look for it. During early storage, a nutcracker memorizes many of the RPs it will take. Nutcrackers recover caches on the first attempt (with the first reference), whereas the proposed approach takes into consideration the possibility of failing on the first try. According to Eq. (43), the nutcrackers that are unable to reach their cache using the first RP will activate their spatial memory to the second RP.

$$\overrightarrow{X}_i^{t+1} = \begin{cases} \overrightarrow{X}_i^t, if\, f\left(\overrightarrow{X}_i^t\right) < if\left(\overrightarrow{RP}_{i,2}^t\right) \\ \overrightarrow{RP}_{i,2}^t, otherwise \end{cases} \tag{43}$$

Eq. (42) is modified in NOA to account for the second RP on the assumption that a nutcracker will utilize it to find its cache, as defined in Eq. (44).

$$X_{ij}^{t+1} = \begin{cases} X_{ij}^t, if\ \tau_5 < \tau_6 \\ X_{ij}^t + r_1 \bullet \left(X_{best,j}^t - X_{ij}^t\right) + r_2 \bullet \left(\overrightarrow{RP}_{i,2}^t - X_{Cj}^t\right), otherwise \end{cases} \tag{44}$$

where τ_5, and τ_6 are randomly generated numbers between 0 and 1. Eq. (45) represents the tradeoff between the first and second RPs within the recovery behavior.

$$\overrightarrow{X}_i^{t+1} = \begin{cases} Eq.(42), if\ \tau_7 < \tau_8 \\ Eq.(44), otherwise \end{cases} \tag{45}$$

where τ_7 and τ_8 are random numbers created between 0 and 1. The first state in the preceding equation represents a nutcracker who recalls the secret store, whereas the second case represents a nutcracker who forgets about the hidden store. According to Eq. (46), the spatial memory between the first and second RPs, as well as the current position, is activated to search for the cache:

$$\overrightarrow{X}_i^{t+1} = \begin{cases} Eq.(41), if\, f(Eq.(41)) < f(Eq.(43)) \\ Eq.(43), otherwise \end{cases} \tag{46}$$

In order to achieve a balance between exploitation and exploration, the recovery and cache-search phases are then randomly swapped as defined in Eq. (47).

$$\overrightarrow{X}_i^{t+1} = \begin{cases} Eq.(44), if\ \phi < P_{a_2} \\ Eq.(45), otherwise \end{cases} \tag{47}$$

where P_{a_2} is a preset value between 0 and 1 indicating the probability of the exploitation stage in the optimization process. However, in the NOA algorithm, the nutcrackers remain in their current location if the quality of their solution is superior to that of the new solution. In conclusion, Eq. (48) expresses the preceding concept.

$$\overrightarrow{X}_i^{t+1} = \begin{cases} \overrightarrow{X}_i^{t+1}, if\, f\left(\overrightarrow{X}_i^{t+1}\right) < f\left(\overrightarrow{X}_i^t\right) \\ \overrightarrow{X}_i^t, otherwise \end{cases} \tag{48}$$

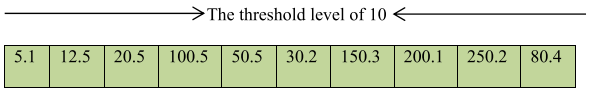


Fig. 5. A solution representation to multilevel thresholding image segmentation.

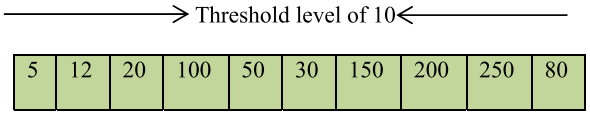


Fig. 6. Unsorted integer values.

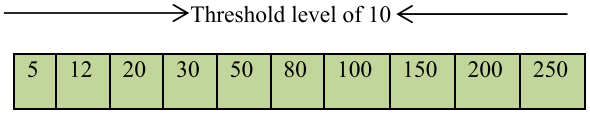


Fig. 7. Sorted integer values.

5. Proposed algorithm

In this section, the NOA is integrated with an effective improvement strategy to strengthen its exploration and exploitation operator for solving some discrete and continuous optimization problems such as the parameter estimation of PV models and PEMFC, and multilevel thresholding image segmentation problem. In detail, the main steps of the proposed algorithm, namely improved NOA (INOA), will be discussed in the next subsections.

5.1. Initialization and evaluation

Similar to population-based algorithms, the proposed algorithm randomly initializes N solutions $\overrightarrow{X}_i(i \in N)$ within the search space of the optimization problem by Eq. (49), where each solution has d dimensions to be optimized.

$$\overrightarrow{X_i} = \overrightarrow{L} + \left(\overrightarrow{U} - \overrightarrow{L}\right).\overrightarrow{r} \tag{49}$$

where $\overrightarrow{r}$ is a randomly generated vector between 0 and 1. These initialized solutions are not applicable to the multithresholding image segmentation problem because it is discrete and these solutions are continuous. Therefore, these solutions will be converted into discrete solutions, as clarified next, to be relevant to this problem. The grey-scale level is represented in 8-bit, where the minimum values $(\overrightarrow{L})$ are 0 and the maximum values $(\overrightarrow{U})$ are $2^8 - 1 = 255$. Assuming that we have an image with 10 various objects (K), which need to be separated by finding the near-optimal threshold values. Therefore, the proposed algorithm will first distribute its solutions randomly throughout the search space, as exemplified in Fig. 5. Afterwards, these solutions will be converted into integers by dropping all decimal places past the dot, as shown in Fig. 6. As indicated in Fig. 7, these converted numbers will be sorted ascendingly, and Eq. (26) will be utilized to determine the quality of these threshold values according to Kapur's entropy.

5.2. Convergence improvement strategy (CIS)

This method is designed to expedite the convergence speed of the NOA while mitigating the issue of local minima. The approach consists of three key components. The first component involves comparing the fitness value of a randomly selected solution with that of the current solution. If the current solution is inferior, a new solution is generated by updating the current solution in the direction of the randomly selected solution. In addition, two new solutions are chosen randomly from the population, and the current solution is updated by moving towards the solution with the highest fitness value. It is plausible that the randomly selected solution may be closer to the near-optimal solution compared to the best-so-far solution. This strategy leverages the regions within the randomly selected solution that exhibit higher fitness values than the other two randomly selected solutions from the current set of solutions. The purpose is to accelerate convergence and prevent getting trapped in local minima. This particular aspect can be mathematically formulated as defined in Eq. (50).

$$\overrightarrow{G}_i(t+1) = \overrightarrow{X}_i(t) + (r * (\overrightarrow{v_1}) + (1-r) * (\overrightarrow{v_2})) \tag{50}$$

$$\overrightarrow{v_1} = \begin{cases} \overrightarrow{X_i^t} - \overrightarrow{X_a^t}, if\ f\left(\overrightarrow{X_i^t}\right) \le f\left(\overrightarrow{X_a^t}\right) \\ \overrightarrow{X_a^t} - \overrightarrow{X_i^t}, otherwise \end{cases} \tag{51}$$

$$\overrightarrow{v_2} = \begin{cases} \overrightarrow{X_b^t} - \overrightarrow{X_c^t}, if\ f\left(\overrightarrow{X_b^t}\right) \le f\left(\overrightarrow{X_c^t}\right) \\ \overrightarrow{X_c^t} - \overrightarrow{X_b^t}, otherwise \end{cases} \tag{52}$$

where $a, b,$ and c are three numbers randomly selected from the solutions such that $a \neq b \neq c \neq i$. r is a number generated randomly between 0 and 1. The second and third folds are formulated in Eqs. (52) and (53), respectively, as an attempt to accelerate the convergence speed in the direction of the best-so-far solution while avoiding getting stuck in local minima.

$$\overrightarrow{X}_i(t+1) = \frac{\left(\overrightarrow{X}_i(t) + \overrightarrow{X}_a(t) + \overrightarrow{X}^t_{best}\right)}{3} + \beta * \left(- \overrightarrow{X}^t_{best} + \overrightarrow{X}_i(t)\right) + r_1 * \left(\overrightarrow{X_b^t} - \overrightarrow{X_c^t}\right) \tag{53}$$

$$\overrightarrow{X}_i(t+1) = \frac{\left(\overrightarrow{X}_i(t) + \overrightarrow{X}_a(t) + \overrightarrow{X}^t_{best}\right)}{3} + r * \left(\overrightarrow{X}^t_{best} - \overrightarrow{X}_i(t)\right) * \overrightarrow{U} + r_1 * \left(\overrightarrow{X_b^t} - \overrightarrow{X_c^t}\right) * \overrightarrow{U_1} \tag{54}$$

$$\beta = \frac{1}{1+e^{\sigma l}} \tag{55}$$

where σ is a constant value determined later within the experiment section; l is a variable including values linearly decreasing from -1 to -2. $\overrightarrow{U}$ and $\overrightarrow{U_1}$ are binary vectors generated according to Eqs. (56) and (57), respectively.

$$\overrightarrow{U} = \begin{cases} 1 & \overrightarrow{r_2} \ge \overrightarrow{r}_3 \\ 0 & otherwise \end{cases} \tag{56}$$

$$\overrightarrow{U_1} = \begin{cases} 1 & \overrightarrow{r_2} \ge r \\ 0 & otherwise \end{cases} \tag{57}$$

Eq. (58) is used to calculate the tradeoff between these three folds.

$$\overrightarrow{X}_i(t+1) \begin{cases} Eq.\ (52) & if\ r < \delta \\ Eq.\ (53) & if\ r_1 < \delta_1 \\ Eq.\ (49) & Else \end{cases} \tag{58}$$

where δ and δ_1 are controlling parameters defined within the experiments to determine the probability of each fold.

5.3. Pseudo-code of the proposed INOA

The conventional NOA is effectively combined with CIS to present a new metaheuristic algorithm for both continuous and discrete optimization problems. This improved variant is known as improved NOA (INOA). In addition, the conventional NOA is used with a probability P_{se} in order to optimize the utilization of the number of function evaluations for accelerating convergence towards the near-optimal solution. In the experiments, the ideal value for this probability is examined. Finally, Algorithm 1 describes the steps of the proposed INOA. The main advantages of INOA are:

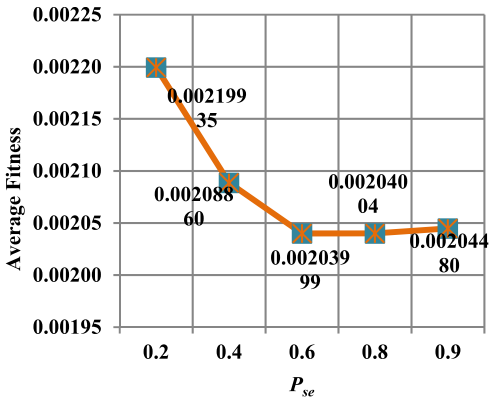
- Easy to implement
- Having a better convergence speed
- Having a high ability to alleviate falling into local minima

Table 1
Properties of RTC France solar cell and PV models.

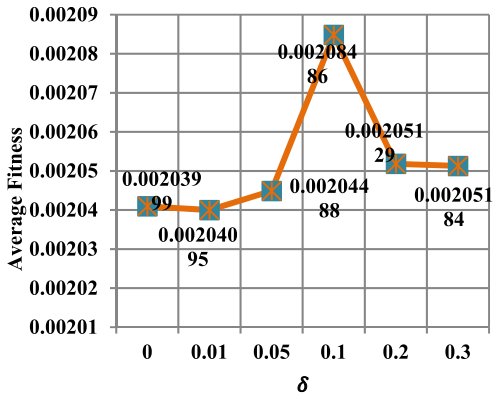
characteristics	RTC	PWP
$P_m[W]$	0.31	11.5
$V_m[V]$	0.459	12.649
$I_m[A]$	0.6755	0.912
$V_{oc}[V]$	0.5736	16.7785
$I_{SC}[A]$	0.7605	1.0317
N_s	1	36
K_i	0.000387	0.0008
K_v	−0.003739	−0.0725

Table 2
The search boundary of unknown parameters.

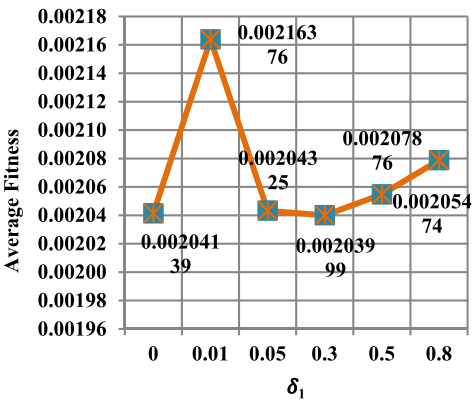
	$I_{ph}(A)$	$I_{sdi}(A), i \in 1:3$	$R_s(\Omega)$	$R_{sh}(\Omega)$	$a1$	$a2$	$a3$
U	$1.1I_{SC}$	$10\ \mu A$	0.5	500	2	2	2
L	$0.9I_{SC}$	$1\ nA$	0	0	1	1.2	1.4



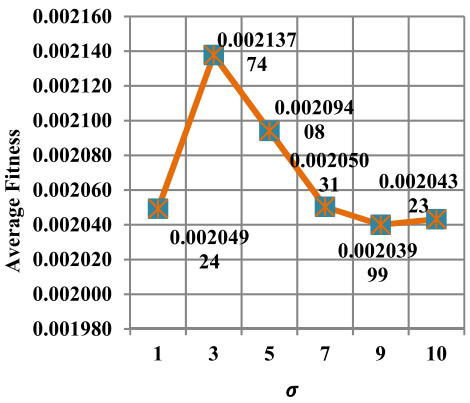
a) Adjusting the parameter P_{se}



b) Adjusting the parameter δ



c) Adjusting the parameter δ_1



d) Adjusting the parameter σ

Fig. 8. Adjusting the parameters of the proposed INOA for the parameter estimation of the SDM-based PWP module.

Table 3
Performance evaluation using SDM-based RTC France.

	Best-obtained parameters					RMSE					
	$I_{ph}(A)$	$I_{sd}(\mu A)$	$R_s(\Omega)$	$R_{sh}(\Omega)$	n	*Best*	*Worst*	*Avg*	*SD*	*Rank*	*p-value*
INOA	**0.7608**	**3.11E-07**	**0.037**	**52.890**	**1.477**	**7.7301E-04**	**7.7301E-04**	**7.7301E-04**	**1.2040E-17**	**1**	
NOA	0.7607	3.63E-07	0.036	57.879	1.493	8.1925E-04	1.7862E-03	1.2592E-03	2.5752E-04	2	**3.01E-11**
FLA	0.7635	3.13E-07	0.036	47.674	1.477	3.1516E-03	1.1387E-01	1.8722E-02	2.7635E-02	10	**3.01E-11**
EO	0.7604	4.05E-07	0.035	65.354	1.504	9.0687E-04	7.3694E-02	6.5035E-03	1.8271E-02	9	**3.01E-11**
ISA	0.7608	3.11E-07	0.037	52.899	1.477	7.7301E-04	7.3694E-02	3.0007E-02	3.6281E-02	11	**2.94E-11**
RUN	0.7601	5.40E-07	0.034	82.564	1.535	1.2236E-03	1.9544E-02	7.7874E-03	2.9242E-03	8	**3.01E-11**
GWO	0.7653	7.84E-09	0.049	18.099	1.182	5.3408E-03	7.3694E-02	1.4019E-02	1.2584E-02	4	**3.01E-11**
MPA	0.7599	6.16E-07	0.034	99.606	1.549	1.4290E-03	6.3233E-03	4.8931E-03	1.2239E-03	3	**3.01E-11**
SMA	0.7620	1.02E-07	0.041	32.991	1.374	1.8856E-03	8.6503E-03	6.9849E-03	1.7972E-03	7	**3.01E-11**
DO	0.7605	2.76E-07	0.037	55.921	1.465	8.4908E-04	8.0755E-03	5.2948E-03	2.5368E-03	4	**3.01E-11**
POA	0.7580	1.57E-07	0.040	96.135	1.411	2.4347E-03	8.6882E-03	5.3957E-03	1.8125E-03	5	**3.01E-11**

Bold values indicate the best results.

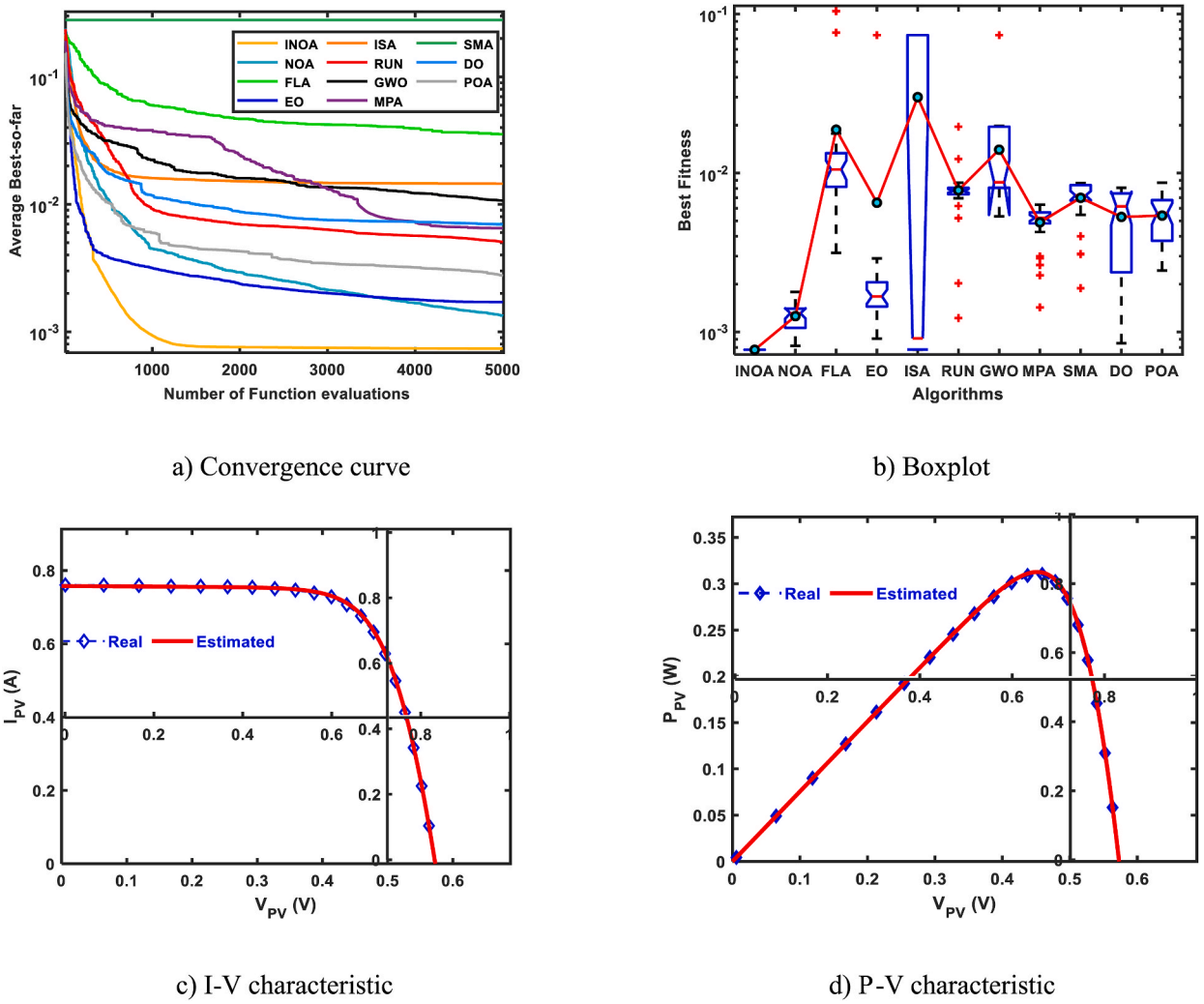
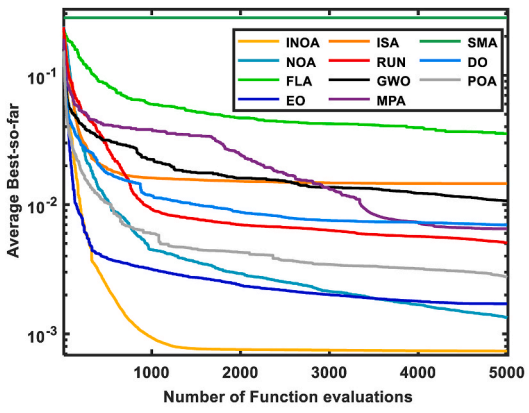


Fig. 9. Various comparisons among algorithms using SDM.

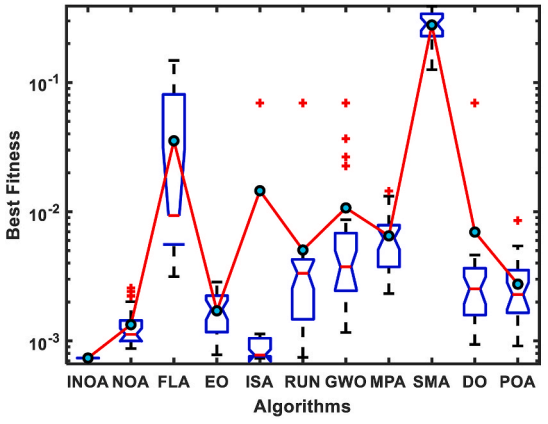
Table 4
Performance evaluation using DDM-based RTC France.

	Best-obtained parameters							RMSE					
	$I_{ph}(A)$	$I_{sd1}(A)$	$I_{sd2}(A)$	$R_s(\Omega)$	$R_{sh}(\Omega)$	n_1	n_2	Best	Worst	Avg	SD	Rank	p-value
INOA	**0.7608**	**9.17E-08**	**2.08E-06**	**0.038**	**58.065**	**1.377**	**2.000**	**7.3274E-04**	**7.4201E-04**	**7.3703E-04**	**2.1080E-06**	**1**	
NOA	0.7606	2.03E-07	8.91E-07	0.037	61.952	1.444	1.942	8.7247E-04	2.5633E-03	1.3364E-03	4.7787E-04	2	**3.02E-11**
FLA	0.7558	1.80E-08	2.92E-06	0.043	483.690	1.253	2.000	3.1454E-03	1.4847E-01	3.5400E-02	4.6868E-02	10	**3.02E-11**
EO	0.7607	1.99E-07	6.63E-07	0.036	58.429	1.444	1.864	7.8147E-04	2.8516E-03	1.7097E-03	5.9515E-04	3	**3.02E-11**
ISA	0.7608	8.08E-08	2.24E-06	0.038	58.599	1.367	2.000	7.3277E-04	6.9442E-02	1.4534E-02	2.7924E-02	9	**8.33E-08**
RUN	0.7609	6.92E-08	1.52E-06	0.038	55.291	1.357	1.901	7.4639E-04	6.9443E-02	5.0538E-03	1.2247E-02	5	**3.02E-11**
GWO	0.7615	2.29E-07	1.24E-06	0.035	53.269	1.459	1.954	1.1617E-03	6.9479E-02	1.0702E-02	1.7859E-02	8	**3.02E-11**
MPA	0.7608	2.53E-09	5.00E-06	0.042	490.048	1.142	1.970	2.3234E-03	1.4455E-02	6.5017E-03	3.0484E-03	6	**3.02E-11**
SMA	0.7456	2.23E-07	2.80E-06	0.209	39.133	1.604	1.941	1.2560E-01	3.8795E-01	2.7893E-01	6.6896E-02	11	**3.02E-11**
DO	0.7608	2.30E-08	1.01E-06	0.040	52.288	1.281	1.764	9.3909E-04	6.9442E-02	6.9604E-03	1.7019E-02	**7**	**3.02E-11**
POA	0.7602	1.75E-07	1.52E-06	0.037	74.511	1.433	1.981	9.1404E-04	8.5431E-03	2.7510E-03	1.5974E-03	**4**	**3.02E-11**

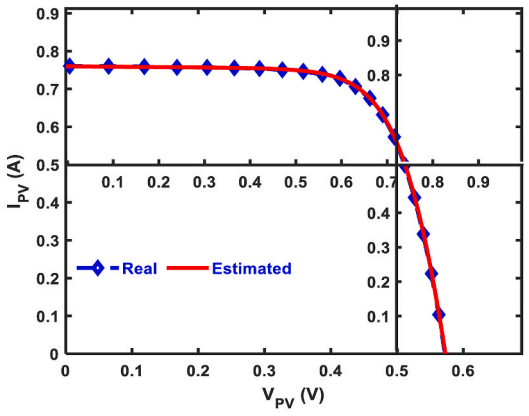
Bold values indicate the best results.



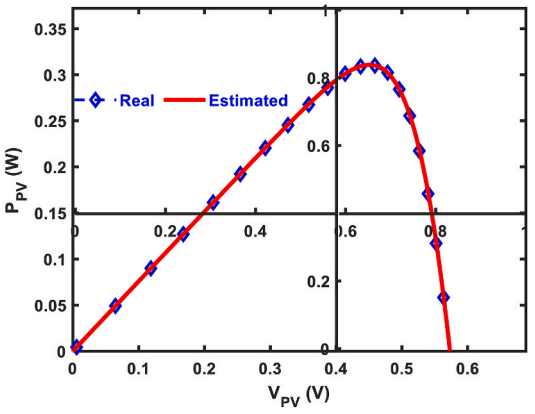
a) Convergence curve



b) Boxplot



c) I-V characteristic



d) P-V characteristic

Fig. 10. Various comparisons among algorithms using DDM-based RTC France.

Table 5
Performance evaluation using TDM-based RTC France.

	Best-obtained parameters									RMSE					
	$I_{ph}(A)$	$I_{sd1}(A)$	$I_{sd2}(A)$	$I_{sd3}(A)$	$R_s(\Omega)$	$R_{sh}(\Omega)$	n_1	n_2	n_3	*Best*	Worst	Avg	SD	Rank	p-value
INOA	**0.7605**	**8.9E-08**	**1.8E-09**	**2.1E-06**	**0.038**	**61.327**	**1.376**	**1.408**	**2.000**	**7.51E-04**	**7.71E-04**	**7.56E-04**	**5.5E-06**	**1**	
NOA	0.7601	5.2E-07	7.1E-08	4.0E-07	0.033	107.015	1.536	1.828	1.960	1.56E-03	3.52E-03	2.48E-03	5.4E-04	7	**3.0E-11**
FLA	0.7605	2.4E-07	3.5E-07	2.0E-06	0.036	500.000	1.466	2.000	2.000	3.14E-03	5.25E-02	2.18E-02	1.3E-02	11	**3.0E-11**
EO	0.7594	1.4E-07	1.2E-06	2.7E-07	0.035	134.779	1.753	1.975	1.477	1.50E-03	3.51E-03	2.45E-03	4.2E-04	6	**3.0E-11**
ISA	0.7605	1.6E-06	1.2E-09	1.3E-07	0.037	59.509	2.000	1.652	1.406	7.54E-04	2.06E-03	1.06E-03	4.4E-04	2	**3.8E-10**
RUN	0.7605	1.7E-06	1.2E-08	1.2E-07	0.037	60.841	1.997	1.730	1.400	7.55E-04	8.35E-03	4.47E-03	2.6E-03	10	**9.0E-11**
GWO	0.7605	1.0E-06	1.1E-06	1.2E-07	0.037	70.005	1.998	1.981	1.400	8.36E-04	4.93E-03	2.65E-03	1.1E-03	9	**3.0E-11**
MPA	0.7604	5.6E-07	3.9E-09	1.2E-08	0.039	61.122	1.588	1.204	1.995	8.11E-04	4.35E-03	2.34E-03	8.2E-04	5	**3.0E-11**
SMA	0.7605	1.0E-09	6.9E-09	5.0E-06	0.040	100.236	1.464	1.200	2.000	1.09E-03	5.40E-03	2.48E-03	1.1E-03	7	**3.0E-11**
DO	0.7605	4.0E-07	8.1E-09	3.9E-07	0.038	66.403	1.609	1.245	1.682	8.23E-04	3.25E-03	2.04E-03	5.9E-04	4	**3.0E-11**
POA	0.7605	1.7E-06	2.1E-08	1.2E-07	0.038	60.495	2.000	2.000	1.400	7.53E-04	2.23E-03	1.48E-03	5.2E-04	3	**4.6E-10**

Bold values highlight the best findings.

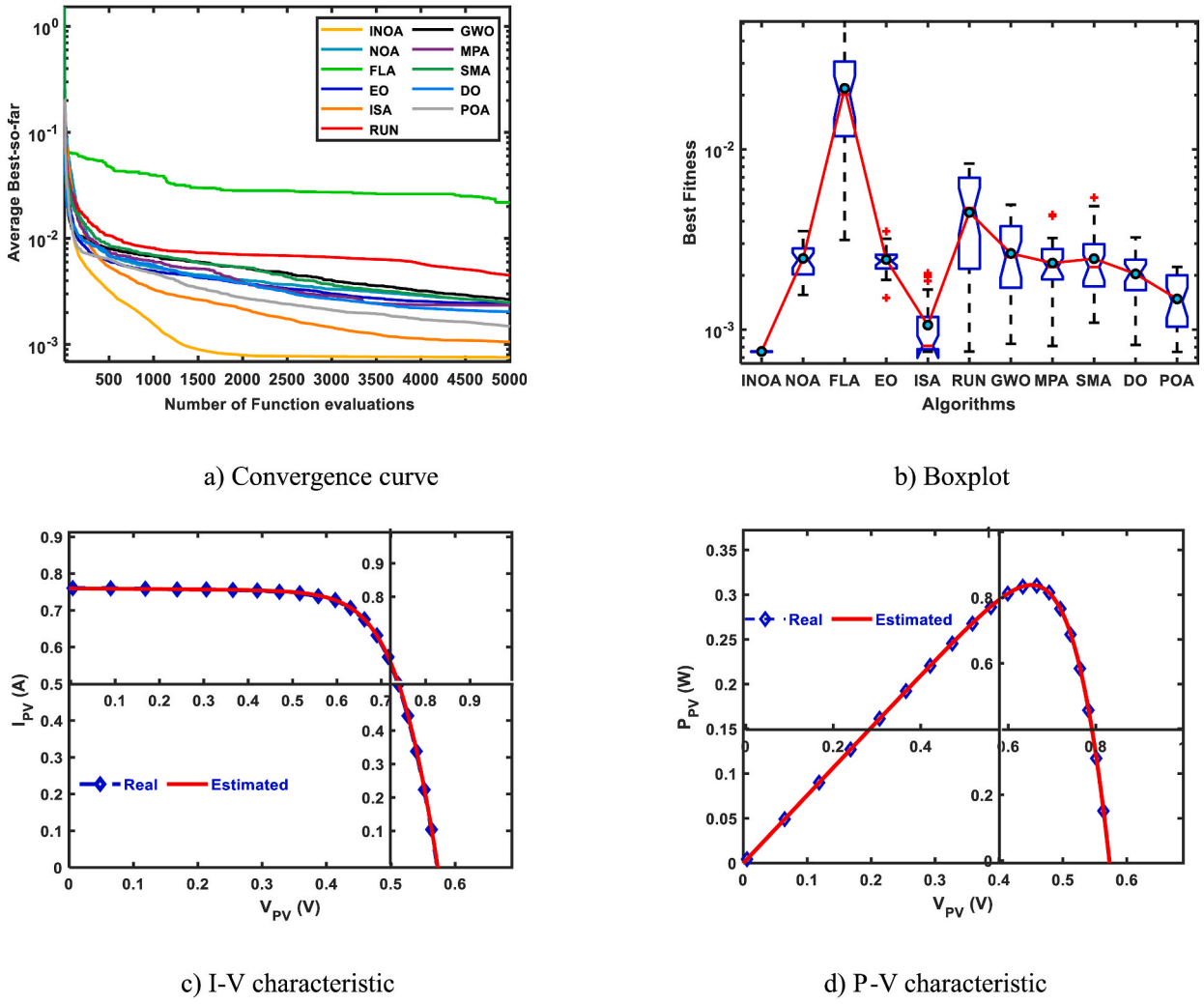


Fig. 11. Various comparisons among algorithms using TDM-based RTC France.

Table 6
Performance evaluation using SDM-based PWP module.

	Best-obtained parameters					RMSE					
	$I_{ph}(A)$	$I_{sd}(\mu A)$	$R_s(\Omega)$	$R_{sh}(\Omega)$	n	*Best*	*Worst*	*Avg*	*SD*	*Rank*	*p-value*
INOA	**1.0324**	**2.50E-06**	**3.45E-02**	**20.7867**	**1.3166**	**0.0020400**	**0.0024435**	**0.0020543**	**0.0000737**	**1**	
NOA	1.0322	2.58E-06	3.44E-02	21.4815	1.3199	0.0020423	0.0025238	0.0021666	0.0001228	2	9.73E-10
FLA	1.0259	6.62E-06	3.07E-02	100.0000	1.4229	0.0035667	0.1572649	0.0508148	0.0559063	10	3.01E-11
EO	1.0322	2.59E-06	3.43E-02	21.4115	1.3204	0.0020420	0.0951542	0.0060308	0.0168403	5	5.48E-11
ISA	1.0324	2.50E-06	3.45E-02	20.7876	1.3166	0.0020400	0.0951542	0.0486342	0.0473153	1	2.43E-10
RUN	1.0318	2.32E-05	2.54E-02	99.6524	1.5899	0.0062080	0.0951542	0.0280828	0.0376406	8	3.01E-11
GWO	1.0366	3.30E-06	3.31E-02	15.4049	1.3461	0.0028738	0.0951542	0.0608428	0.0416180	11	3.01E-11
MPA	1.0266	3.82E-06	3.36E-02	71.2188	1.3606	0.0029178	0.0054980	0.0041261	0.0007022	3	3.01E-11
SMA	1.0396	1.68E-06	3.51E-02	11.5031	1.2785	0.0032482	0.0951543	0.0103516	0.0161451	6	3.01E-11
DO	1.0272	4.03E-06	3.33E-02	61.1112	1.3665	0.0027780	0.0951542	0.0238049	0.0363285	7	3.01E-11
POA	1.0309	3.76E-06	3.32E-02	28.1371	1.3594	0.0022363	0.0081123	0.0047595	0.0015656	4	3.68E-11

Bold values indicate the best results.

Table 7
Performance evaluation using DDM-based PWP module.

	Best-obtained parameters							RMSE					
	$I_{ph}(A)$	$I_{sd1}(A)$	$I_{sd2}(A)$	$R_s(\Omega)$	$R_{sh}(\Omega)$	n_1	n_2	Best	Worst	Avg	SD	Rank	p-value
INOA	**1.0324**	**2.31E-09**	**2.49E-06**	**0.0345**	**20.7867**	**1.3166**	**1.3166**	**0.0020400**	**0.0023454**	**0.0020711**	**0.0000669**	**1**	
NOA	1.0325	1.49E-09	2.98E-06	0.0339	21.8105	1.1553	1.3354	0.0020856	0.0027508	0.0021795	0.0001296	2	**3.35E-08**
FLA	1.0242	1.99E-06	5.00E-06	0.0360	500.0000	1.2969	1.8684	0.0043064	0.1533691	0.0592287	0.0544239	9	**3.02E-11**
EO	1.0332	1.94E-08	2.13E-06	0.0350	18.3995	1.1801	1.3042	0.0020818	0.0939028	0.0175745	0.0347194	6	**1.41E-09**
ISA	1.0324	2.25E-06	2.25E-06	0.0346	20.9321	1.3072	2.0000	0.0020653	0.0940767	0.0724833	0.0395037	10	**6.12E-10**
RUN	1.0305	4.82E-07	2.97E-06	0.0349	26.6476	1.2233	1.4147	0.0022137	0.0939027	0.0061759	0.0165819	4	**4.50E-11**
GWO	1.0393	1.79E-08	4.80E-07	0.0403	10.2600	1.0184	1.2000	0.0043963	0.0939200	0.0582602	0.0436114	8	**3.02E-11**
MPA	1.0269	2.68E-09	2.54E-06	0.0356	56.3927	1.0562	1.3202	0.0033363	0.0115753	0.0078424	0.0028378	5	**3.02E-11**
SMA	1.0697	3.55E-06	3.74E-06	0.0236	496.1064	1.3353	1.9234	0.0675634	0.2703185	0.1801100	0.0447674	11	**3.02E-11**
DO	1.0256	3.93E-06	3.98E-06	0.0327	233.4422	1.3695	1.7740	0.0032496	0.0939030	0.0322190	0.0411050	7	**3.02E-11**
POA	1.0310	1.35E-06	1.83E-06	0.0345	24.6590	1.2834	1.4391	0.0021189	0.0078216	0.0037715	0.0012984	3	**4.98E-11**

Bold values indicate the best results.

Table 8
Performance evaluation using TDM-based PWP module.

	Best-obtained parameters									RMSE				
	$I_{ph}(A)$	$I_{sd1}(A)$	$I_{sd2}(A)$	$I_{sd3}(A)$	$R_s(\Omega)$	$R_{sh}(\Omega)$	n_1	n_2	n_3	*Best*	Worst	Avg	SD	Rank	p-value
INOA	**1.032**	**1.0E-09**	**2.6E-06**	**1.0E-09**	**0.034**	**22.16**	**1.32**	**1.32**	**1.40**	**2.0507E-03**	**2.20E-03**	**2.06E-03**	**2.7E-05**	**1**	
NOA	1.030	3.1E-06	1.3E-06	2.3E-06	0.033	37.25	1.35	1.61	1.86	2.4494E-03	3.29E-03	2.90E-03	2.5E-04	3	**3.02E-11**
FLA	1.025	2.7E-07	5.4E-08	5.0E-06	0.033	146.85	1.58	1.20	1.40	3.2097E-03	2.51E-02	6.61E-03	4.1E-03	11	**3.02E-11**
EO	1.031	1.1E-06	2.8E-06	2.6E-06	0.033	29.18	1.60	1.33	1.99	2.2483E-03	3.40E-03	2.92E-03	2.3E-04	4	**3.02E-11**
ISA	1.032	2.6E-06	1.2E-09	1.8E-08	0.034	22.20	1.32	1.79	2.00	2.0509E-03	3.39E-03	2.43E-03	4.6E-04	2	**3.02E-11**
RUN	1.032	3.3E-07	2.3E-07	3.0E-06	0.035	22.98	1.23	1.25	1.41	2.1025E-03	4.75E-03	2.97E-03	5.4E-04	6	**3.02E-11**
GWO	1.031	1.9E-07	3.6E-08	4.5E-06	0.033	28.98	1.25	1.44	1.40	2.4142E-03	4.09E-03	3.05E-03	3.9E-04	7	**3.02E-11**
MPA	1.030	1.0E-09	2.0E-06	8.0E-06	0.034	33.04	2.00	1.30	1.83	2.3276E-03	3.47E-03	2.95E-03	3.1E-04	5	**3.02E-11**
SMA	1.029	1.0E-09	1.6E-07	4.1E-06	0.034	40.92	1.00	1.20	1.40	2.5582E-03	5.05E-03	3.35E-03	7.1E-04	10	**3.02E-11**
DO	1.032	1.4E-06	6.6E-07	8.3E-07	0.036	20.59	1.46	1.23	1.40	2.2057E-03	4.36E-03	3.33E-03	4.6E-04	9	**3.02E-11**
POA	1.031	2.0E-06	8.9E-06	8.5E-08	0.034	28.17	1.30	2.00	1.40	2.2247E-03	4.14E-03	3.17E-03	3.7E-04	8	**3.02E-11**

Bold values highlight the best findings.

Meanwhile, its main advantage is that it has four controlling parameters that have to be accurately estimated before starting the optimization to improve its effectiveness.

Algorithm 1. Steps of the proposed INOA

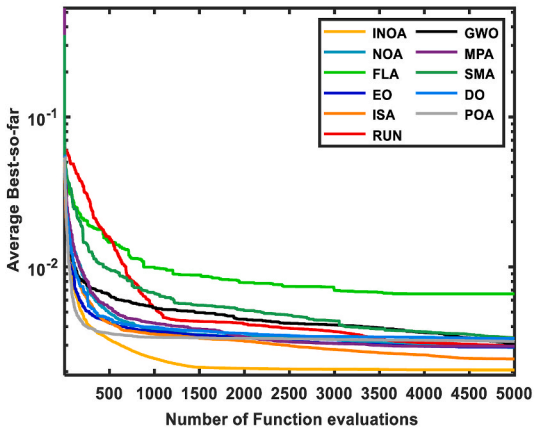
Input: population size N, $\overrightarrow{L}$, $\overrightarrow{U}$, and T_{max};
Output: $\overrightarrow{X}_{best}^{t}$
1. Initialize N solutions by Eq. (49);
2. Evaluation and extracting the best solution
3. $t = 1$;//the current evaluation//**4. while** $(t < T_{max})$
5. r: A number generated randomly between 0 and 1.
6. if $r < P_{se}$
7. σ and σ_1: Numbers generated randomly between 0 and 1.
8. **If** $\sigma < \sigma_1$//* **Foraging and storage strategy***//.
9. φ is a random number between 0 and 1.
10. **for** i=1:N
11. **if** $\varphi < P_{a_1}$ /* **Exploration phase1***/
12. Create $\overrightarrow{X}_i^{t+1}$ by Eq. (32) and Eq. (48)
13. **else**/*Exploitation phase1*/
14. Create $\overrightarrow{X}_i^{t+1}$ by Eq. (34) and Eq. (48)
15. **end if**
16. $t = t + 1$
17. **end for**
18. **else**//* **Cache-search and recovery strategy** *//.
19. Compute RP matrix by Eqs. (36)–(38).
20. : A number generated randomly between 0 and 1.
21. **for** i=1:N
22. **if** $\phi < P_{a_2}$/***Exploitation phase2***/**
23. Create $\overrightarrow{X}_i^{t+1}$ by Eq. (46) and Eq. (48)
24. **else**/*Exploration phase2*/
25. Create $\overrightarrow{X}_i^{t+1}$ by Eq. (47) and Eq. (48)
26. **end if**
27. $t = t + 1$
28. **end for**
29. **end if**
30. **end if**
31.//* **Applying CIS Strategy** *//
32. **for** $i = 1{:}N$
33. Updating $\overrightarrow{X}_i^{t+1}$ according to Eq. (58) and Eq. (48)
34. $t = t + 1$
35. **end for**
36. **end while**

6. Results and discussion

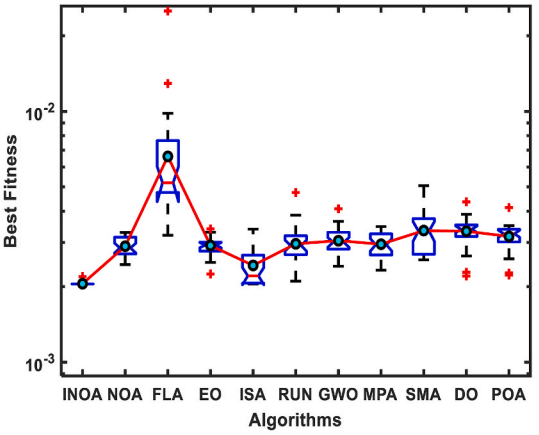
In this paper, two well-known optimization problems, namely parameter estimation of both PEMFC and PV models, are used to check the efficacy of the proposed algorithm for problems with continuous decision variables. In addition, a well-known discrete problem known as the multi-thresholding image segmentation problem is used to disclose the efficiency of INOA for problems with discrete decision variables. INOA is extensively compared to several well-known state-of-the-art in terms of several pieces of statistical information extracted after 30 independent runs for each algorithm, including the best, average, worst, standard deviation (SD), rank, and p-value returned by the Wilcoxon rank-sum test. Experiments with MATLAB R2019a are performed on a machine with the following specifications: an Intel(R) Core(TM) i7-4700MQ CPU running at 2.40 GHz, 32 GB of random access memory (RAM), and a 64-bit version of Windows 10 Pro.

6.1. Application I: Parameter estimation of PV models

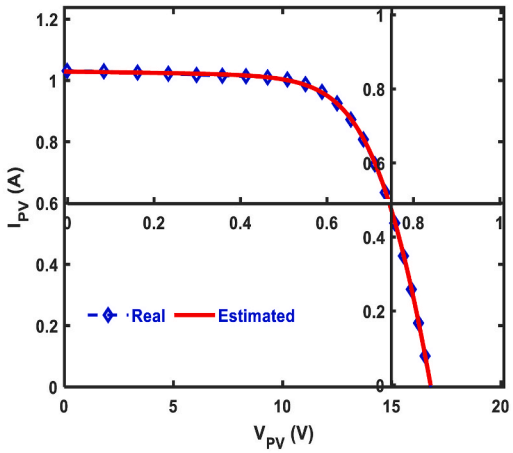
This section examines the effectiveness of INOA on the RTC France solar cell and a well-known PV module: Photowatt-PWP201 (PWP) [97]. Table 1 displays the open-circuit current-voltage (V_{oc}), the short-circuit current-temperature factor (k_i), the maximum output current (I_m), maximum output power (P_m), short-circuit current point (I_{SC}), the number of cells connected in series in PV modules (N_s), the maximum output voltage (V_m), and the temperature coefficient of open-circuit voltage (k_v) for these PV models at standard conditions (STC) [98]. Table 2 displays the maximum and minimal values for each of the unknown parameters. Here we compare the proposed INOA to ten popular and recently proposed metaheuristic-based algorithms: NOA [96], MPA [98], Fick's law algorithm (FLA) [99], interior search algorithm (ISA) [100], grey wolf optimizer (GWO) [101], slime mould algorithm (SMA) [102], pelican optimization algorithm (POA) [103], dandelion optimizer (DO), and EO [104]. The controlling parameters of these algorithms



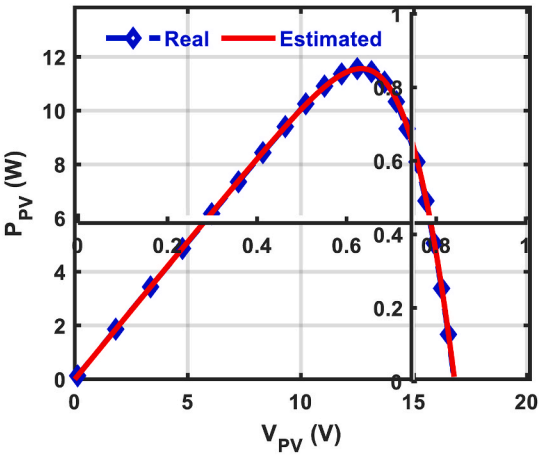
a) Convergence curve



b) Boxplot



c) I-V characteristic



d) P-V characteristic

Fig. 12. Various comparisons among algorithms using TDM-based PWP module.

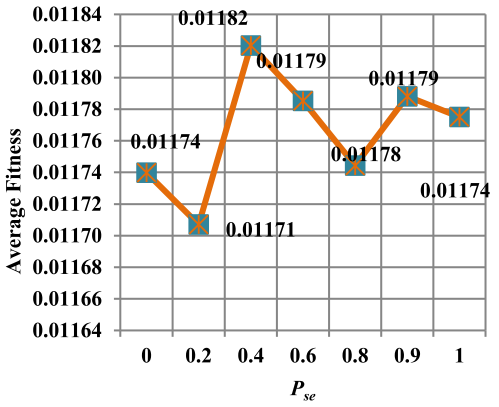
are set as recommended in the cited paper except for T_{max} and N, which are both set to 50000 and 35 in order to provide a fair comparison.

6.1.1. *INOA's sensitivity analysis*

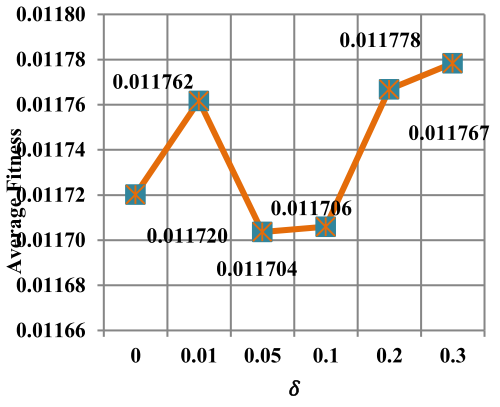
The optimal performance of INOA in dealing with parameter estimation of the PV models requires accurate identification of four newly proposed regulating parameters. In this case, we speculate based on data from extensive experiments performed on the PWP module, as shown in Fig. 8(a–d). Fig. 8(b) shows that the best value for δ is 0.01, Fig. 8(c) reports that the performance of the proposed is maximized when $\delta_1 = 0.3$, and Fig. 8(d) illustrates the best performance of the proposed when setting σ to 9. The likelihood that the standard NOA will be implemented is determined by another controlling parameter, P_{se}. Fig. 8(a) displays the outcomes of multiple experiments in which different values were chosen for this parameter. This figure demonstrates that a value of 0.6 is optimal for this parameter. All other parameters of classical NOA are set in accordance with the suggestions made in Ref. [96].

6.1.2. *RTC France solar cell*

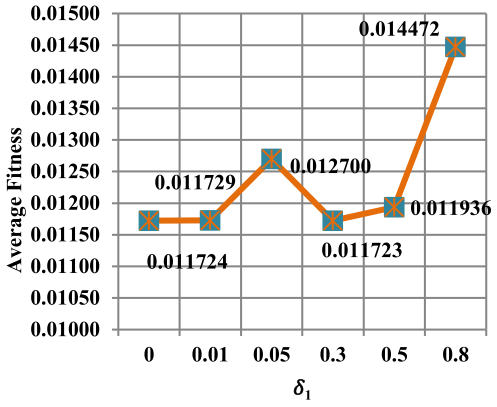
INOA is employed in this section to estimate the unknown parameters of the R.T.C France solar cell based on SDM, DDM, and TDM. The current-voltage data is measured on a commercial silicon RTC France solar cell with a 57-mm diameter and a temperature of 33 °C [97].



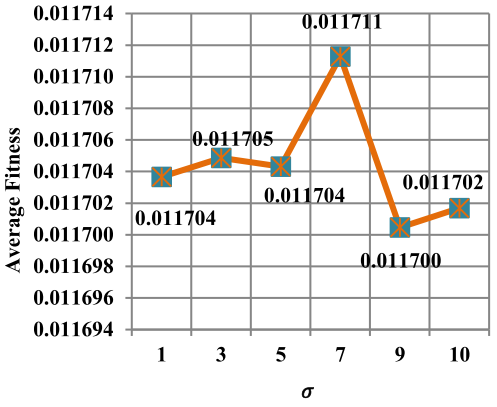
a) Adjusting the parameter P_{se}



b) Adjusting the parameter δ



c) Adjusting the parameter δ_1



d) Adjusting the parameter σ

Fig. 13. Adjusting the parameters of the proposed INOA for the parameter estimation of PEMFC.

A. Single-Diode model

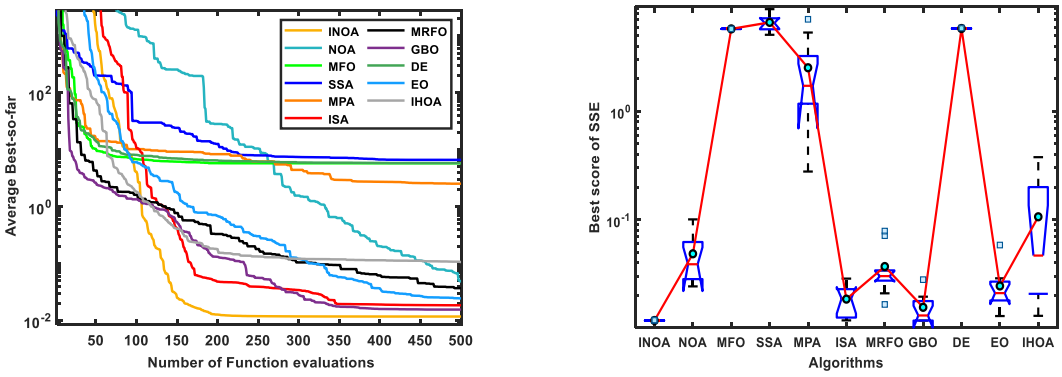
After running each algorithm 30 independent times over the SDM-based RTC France, various statistical information is calculated and described in Table 3. Based on the results presented in Table 3, INOA is the best for all the performance metrics, followed by the classical NOA as the second-best one, while ISA is the worst. In addition, the Wilcoxon rank-sum test was applied to observe the difference between the outcomes of INOA and those of the other algorithms. This test returns a value, referred to as p-value, between each pair of algorithms; this value determines that the Null hypothesis, determining that there is no difference, is accepted if this value is greater than the significance level of 0.05; otherwise, the alternative one is accepted. To affirm the superiority of INOA, Fig. 9 (a) and (b) are presented to show the convergence curve and the five-number summary for each algorithm. This figure affirms that INOA is the best, followed by NOA as the second-best one, while ISA is the worst one. Fig. 9 (c) and (d) show the measured and calculated I-V, and the measured and estimated P-V, which appear that the estimated values significantly agree with the measured ones.

B. Double-Diode model

Various aforementioned performance metrics are computed and reported in Table 4 after each method has been run 30 times on the DDM-based RTC France. This table shows that INOA ranks first, followed by NOA as the second-best one, while ISA performs the poorest. The p-value presented in this table shows that the outcomes of INOA are significantly different from those of the others. Fig. 10 (a) and (b) display the convergence curve and the five-number summary for each algorithm, respectively, to demonstrate INOA's superiority. Based on these figures, INOA is the best, followed by NOA in second place and SMA in last. The estimated values appear to significantly match with the measured ones, as shown in Fig. 10(c) and (d), which display the measured and calculated I-V and the measured and estimated P-V, respectively.

Table 9
Performance evaluation on 500W stack.

Parameter	$\xi_1(V)$	$\xi_2(V/k)$	$\xi_3(V/k)$	$\xi_4(V/k)$	λ	$R_c(\Omega)$	β	*SSE (best)*	*SSE (Avg)*	*SSE (Worst)*	*SSE (SD)*	*p-value*
INOA	**−0.8913**	**3.36E-03**	**8.78E-05**	**−1.93E-04**	**20.8772**	**1.00E-04**	**0.01613**	**0.01170**	**0.01177**	**0.01353**	**0.00034**	
NOA	−0.8544	2.46E-03	4.52E-05	−1.94E-04	21.0093	2.17E-04	0.01505	0.02351	0.07006	0.23119	0.04758	**3.02E-11**
MFO	−1.1854	4.43E-03	9.80E-05	−1.89E-04	18.7558	1.38E-04	0.01474	0.01700	5.89177	8.57576	1.80059	**2.98E-11**
SSA	−0.9214	3.04E-03	6.62E-05	−1.57E-04	14.3364	1.34E-04	0.02493	2.16320	6.20032	9.48733	1.24187	**3.02E-11**
MPA	−0.9262	2.51E-03	3.60E-05	−2.02E-04	22.9994	1.35E-04	0.01399	0.10044	2.94899	7.25196	2.18282	**3.02E-11**
ISA	−0.9450	3.57E-03	9.05E-05	−1.93E-04	22.0370	2.07E-04	0.01619	0.01178	0.02007	0.06372	0.01093	**2.61E-10**
MRFO	−1.1835	4.43E-03	9.80E-05	−1.91E-04	20.4146	1.95E-04	0.01530	0.01325	0.03764	0.13636	0.02553	**3.34E-11**
GBO	−1.1997	4.17E-03	8.16E-05	−1.93E-04	20.8610	1.00E-04	0.01612	0.01170	0.02071	0.03248	0.00807	**8.89E-10**
DE	−0.8532	2.16E-03	3.60E-05	−9.54E-05	13.0000	8.00E-04	0.02214	5.77578	5.80561	5.94820	0.03746	**3.02E-11**
EO	−1.1619	4.30E-03	9.48E-05	−1.91E-04	20.2425	1.99E-04	0.01540	0.01400	0.03138	0.11614	0.02188	**3.02E-11**
IHOA	−1.0823	4.01E-03	9.19E-05	−1.94E-04	22.9988	1.92E-04	0.01681	0.01261	0.10058	0.52536	0.13554	**4.50E-11**



a) Convergence curve b) Boxplot

Fig. 14. Various comparisons among algorithms using the BCS 500W stack module.

C. Triple-Diode model

After 30 independent runs for each algorithm on the TDM-based RTC France, a variety of performance indicators are calculated and provided in Table 5. This table reveals that INOA is the best, followed by ISA as the second best one. This table also includes a p-value, which demonstrates that INOA results are statistically different compared to all the other algorithms. For further illustration of INOA's superiority, Fig. 11(a) and (b) show the convergence curve and the five-number summary for each algorithm, respectively. From this data, INOA is the best, followed by ISA in second place and POA in third. As can be seen in Fig. 11(c) and (d), which show the measured and calculated I-V and the measured and estimated P-V, respectively, the estimated values appear to considerably correspond with the measured ones.

6.1.3. Photowatt-PWP201 module

Here, we use the photowatt-PWP201, another PV module, to compare INOA's performance to that of competing optimizers in determining the PV module's as-yet-unknown parameters under a variety of PV models.

A. Single-Diode model

We calculate and provide in Table 6 a wide range of performance metrics after 30 replicated runs of each method on the SDM-based PWP module. Based on this data, INOA emerges as the top one, followed by NOA as the second one. A p-value is also provided to reveal that INOA yields significantly different outcomes than the rest of the algorithms.

B. Double-Diode model

After running each method on the DDM-based PWP module 30 times, we calculate and give a wide variety of performance data in Table 7. From these results, INOA is the winner, with NOA coming in second. Furthermore, a p-value is presented to demonstrate that INOA yields significantly different results than the rest of the algorithms.

C. Triple-Diode model

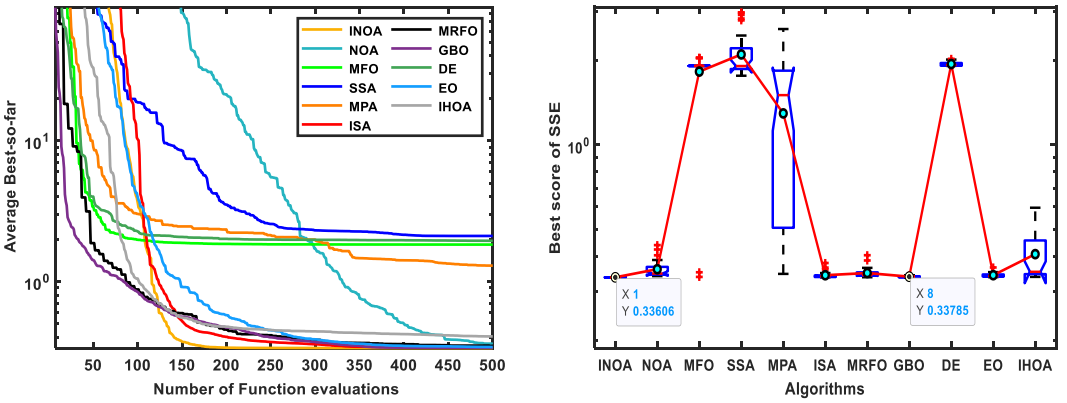
After 30 independent runs for each algorithm on the TDM-based PWP module, a variety of performance indicators are calculated and provided in Table 8. This table reveals that INOA is the best, followed by ISA as the second best one. This table also includes a p-value, which demonstrates that INOA results are statistically different compared to all the other algorithms. For further illustration of INOA's superiority, Fig. 12(a) and (b) show the convergence curve and the five-number summary for each algorithm, respectively. From this data, INOA is the best, followed by ISA in second place and NOA in third. As can be seen in Fig. 12(c) and (d), which show the measured and calculated I-V and the measured and estimated P-V, respectively, the estimated values appear to considerably correspond with the measured ones.

6.2. Application II: Parameters selection of PEMFCs

As an additional engineering optimization problem, parameter estimation for PEMFCs is discussed here in order to evaluate the performance of the proposed INOA. In order to demonstrate the efficacy of the proposed INOA, it is compared to nine well-known metaheuristic algorithms and the classical NOA. These algorithms are ISA [37], GBO [38], DE [22], MPA [23], MRFO [39], equilibrium optimizer (EO), MFO [39], SSA [40], and IHOA [38]. Six popular PEMFC stacks (500W Stack [88], 250W stack [105], Ballard

Table 10
Performance evaluation on 250W stack.

Parameter	$\xi_1(\boldsymbol{V})$	$\xi_2(\boldsymbol{V}/\boldsymbol{k})$	$\xi_3(\boldsymbol{V}/\boldsymbol{k})$	$\xi_4(\boldsymbol{V}/\boldsymbol{k})$	λ	$\boldsymbol{R}_c(\Omega)$	β	*SSE (best)*	*SSE (Avg)*	*SSE (Worst)*	*SSE (SD)*	*p-value*
INOA	**−1.0301**	**3.53E-03**	**7.32E-05**	**−1.75E-04**	**19.9358**	**1.00E-04**	**0.01453**	**0.33598**	**0.33606**	**0.33733**	**0.00026**	
NOA	−1.1091	3.23E-03	4.17E-05	−1.77E-04	21.9517	3.98E-04	0.01549	0.33952	0.35935	0.43802	0.02519	**3.02E-11**
MFO	−0.8532	2.91E-03	6.74E-05	−1.74E-04	19.3380	1.00E-04	0.01360	0.33718	1.82701	2.04445	0.40567	**3.33E-11**
SSA	−1.0932	3.35E-03	5.51E-05	−1.03E-04	13.6648	3.47E-04	0.01389	1.76612	2.10430	2.97399	0.39110	**3.02E-11**
MPA	−0.8535	2.38E-03	3.60E-05	−1.73E-04	22.1000	8.00E-04	0.01486	0.34576	1.29498	2.59359	0.65279	**3.02E-11**
ISA	**−0.9950**	**3.78E-03**	**9.45E-05**	**−1.75E-04**	**19.9358**	**1.00E-04**	**0.01453**	**0.33598**	0.34190	0.37809	0.00935	**1.07E-09**
MRFO	−1.1997	4.45E-03	9.78E-05	−1.75E-04	20.1825	1.00E-04	0.01474	0.33632	0.34820	0.40312	0.01460	**4.50E-11**
ChOA	**−0.9820**	**3.66E-03**	**8.95E-05**	**−1.75E-04**	**19.9383**	**1.00E-04**	**0.01453**	**0.33598**	0.33785	0.34230	0.00166	**5.46E-09**
DE	−0.9401	3.00E-03	6.11E-05	−9.54E-05	14.0492	8.00E-04	0.01360	1.91526	1.94427	2.02862	0.03166	**3.02E-11**
EO	−1.1808	3.97E-03	7.31E-05	−1.74E-04	19.9934	1.59E-04	0.01462	0.33665	0.34202	0.36445	0.00551	**3.69E-11**
IHOA	−0.8899	3.52E-03	9.72E-05	−1.73E-04	19.8430	2.18E-04	0.01401	0.33731	0.40675	0.59601	0.08003	**3.34E-11**



a) Convergence curve b) Boxplot

Fig. 15. Various comparisons among algorithms using 250W stack module.

Mark V5 [106], and Avista SR-12 [107]) have been utilized to evaluate the effectiveness of each of the aforementioned techniques. Except for the number of function evaluations and the population size, which are both set to 5000 and 25, respectively, we have followed the recommendations of the cited studies when configuring the parameters of the methods being compared. The suggested HGTO has its parameters determined in a manner to be further upon in a subsequent section. The search boundary for these unknown factors according to several studies is as follows [108]:

$$-1.1997 < \boldsymbol{\xi_1} < -0.8532$$

$$0.001 < \boldsymbol{\xi_2} < 0.005$$

$$3.6e-5 < \boldsymbol{\xi_3} < 9.8e-5$$

$$-2.6e-4 < \boldsymbol{\xi_4} < -9.54e-5$$

$$13.000 < \boldsymbol{\lambda} < 23.000$$

$$1e-4 < \boldsymbol{R_c} < 8e-4$$

$$1.36e-2 < \boldsymbol{\beta} < 50e-2$$

6.2.1. *INOA's parameter tuning for PEMFC*

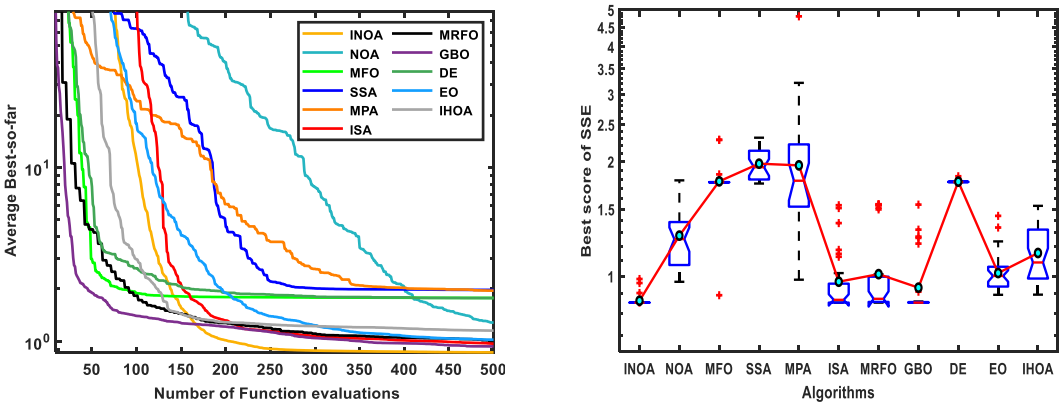
Accurately identifying four newly proposed regulating parameters is a prerequisite for optimizing INOA's performance when dealing with the parameter estimation of the PEMFC. Here, we make an educated guess by testing a broad variety of settings on a BCS 500W stack module, as illustrated in Fig. 13. Three of these parameters: δ, δ_1 and σ are for the newly proposed CIS. According to Fig. 13 (b–d), the optimal values for these parameters are determined to be 0.05, 0.3, and 9, respectively. During optimization, another parameter, P_{se}, decides how likely the conventional NOA will be carried out. Several trials with varying values for this parameter have been performed, and the results are shown in Fig. 13(a). As can be seen in this figure, a value of 0.2 is the best for this parameter. The other classical NOA parameters are established according to the recommendations provided in Ref. [96].

6.2.2. *BCS 500W stack module*

This commercial PEMFC module is commonly constructed with the following structural parameters and operating circumstances for evaluating optimization methods in the literature: $N_{cells} = 32$, $P_{O_2}(atm) = 0.2095$, $A\ (cm^2) = 64$, $l\ (\mu m) = 178$, $T_{fc}(K) = 333$, $J_{max}\left(\frac{mA}{cm^2}\right) = 469$, and $P_{H_2}(atm) = 1.0$ [109,110]. Table 9 displays the best, average (Avg), and worst SSE values, together with their SD collected across 30 independent times to demonstrate stability for each technique. This table shows that INOA is far superior to its competitors in terms of all the performance metrics. Results from INOA and competing optimizers are compared using the Wilcoxon rank-sum test to establish whether or not there is a statistically significant difference between them. Table 9 displays the p-values that determine whether or not there is a statistically significant difference between INOA and each of the other algorithms when comparing their results using a significance threshold of 0.05. INOA's potential for a significantly different outcome compared to competing optimizers is laid out in this table. The accuracy requirements of some algorithms may necessitate several function evaluations. However, this may greatly increase the temporal complexity of the algorithms, making them less acceptable for solving optimization issues than alternatives that reach the same degree of accuracy with fewer function evaluations. Thus, the convergence curves of

Table 11
Performance evaluation on Ballard Mark V stack.

Parameter	$\xi_1(\boldsymbol{V})$	$\xi_2(\boldsymbol{V}/\boldsymbol{k})$	$\xi_3(\boldsymbol{V}/\boldsymbol{k})$	$\xi_4(\boldsymbol{V}/\boldsymbol{k})$	λ	$\boldsymbol{R}_c(\Omega)$	β	*SSE (best)*	*SSE (Avg)*	*SSE (Worst)*	*SSE (SD)*	*p-value*
INOA	**−0.9116**	**2.88E-03**	**3.87E-05**	**−1.63E-04**	**23.0000**	**1.00E-04**	**0.01360**	**0.85361**	**0.86342**	**0.98448**	0.03128	
NOA	−0.9709	3.10E-03	4.36E-05	−1.48E-04	22.9419	2.58E-04	0.01433	0.96809	1.27955	1.78641	0.22339	**3.69E-11**
MFO	−0.8532	2.76E-03	4.30E-05	−1.55E-04	23.0000	1.64E-04	0.01360	0.89261	1.77566	2.28188	0.21182	**4.07E-11**
SSA	−0.8903	3.20E-03	7.02E-05	−9.64E-05	22.7479	7.93E-04	0.01360	1.75249	1.97583	2.31132	0.19027	**3.02E-11**
MPA	−1.1995	3.73E-03	4.21E-05	−1.47E-04	23.0000	1.00E-04	0.01804	0.98041	1.95581	4.81013	0.76040	**3.34E-11**
ISA	**−0.8542**	**2.67E-03**	**3.62E-05**	**−1.63E-04**	**23.0000**	**1.00E-04**	**0.01360**	**0.85361**	0.96864	1.53719	0.19570	**4.98E-04**
MRFO	−1.1997	4.72E-03	9.78E-05	−1.63E-04	22.9998	1.00E-04	0.01360	0.85364	1.01387	1.54656	0.26593	**2.57E-07**
GBO	**−1.1997**	**4.62E-03**	**9.23E-05**	**−1.63E-04**	**23.0000**	**1.00E-04**	**0.01360**	**0.85361**	0.93443	1.54282	0.18395	**2.71E-02**
DE	−0.8532	3.21E-03	7.73E-05	−9.54E-05	22.8884	8.00E-04	0.01360	1.76560	1.77265	1.83107	**0.01271**	**3.02E-11**
EO	−0.9250	3.29E-03	6.19E-05	−1.54E-04	22.8786	1.14E-04	0.01474	0.89492	1.02066	1.44160	0.12491	**3.47E-10**
IHOA	−1.1977	3.74E-03	4.10E-05	−1.56E-04	23.0000	1.87E-04	0.01360	0.89576	1.15194	1.53166	0.21424	**1.33E-10**



a) Convergence curve b) Boxplot

Fig. 16. Various comparisons among algorithms using Ballard Mark V stack module.

various algorithms on this test case are displayed in Fig. 14 (a), revealing that INOA converged faster than all the rival optimizers. Fig. 14 (b) shows a boxplot of the five-number summary for thirty fitness values obtained by each algorithm. The five-number summary appears to support INOA according to this figure.

6.2.3. 250W stack module

Here, we evaluate the performance of our proposed INOA against that of several competing optimizers by utilizing a different PEMFC stack, namely 250W stack, which is commonly utilized to probe the efficacy of the parameter estimation algorithms with the following parameters: $N_{cells} = 24, l\,(\mu m) = 127, T_{fc}(K) = 343.15, A\,(cm^2) = 27, A\,(cm^2) = 27, J_{max}\left(\frac{mA}{cm^2}\right) = 860, P_{O_2}(atm) = 1.0$, and $P_{H_2}(atm) = 1.0$ [109]. Results for all aforementioned performance metrics, optimal parameter values, and Wilcoxon rank-sum test results for each method after 30 separate runs are shown in Table 10. This table clearly shows that INOA outperforms all other algorithms in the vast majority of the evaluated criteria. The p-value from the Wilcoxon rank-sum test shown in the same table demonstrates how significantly different INOA is from the other optimizers. The rates of convergence reached by each approach in this test case are shown in Fig. 15(a), which serves as an additional performance metric. This figure shows that INOA is faster than any of the other rival algorithms. A boxplot presenting a five-number summary of the results achieved by each algorithm is shown in Fig. 15(b), which affirms the superiority of INOA.

6.2.4. Ballard Mark V stack

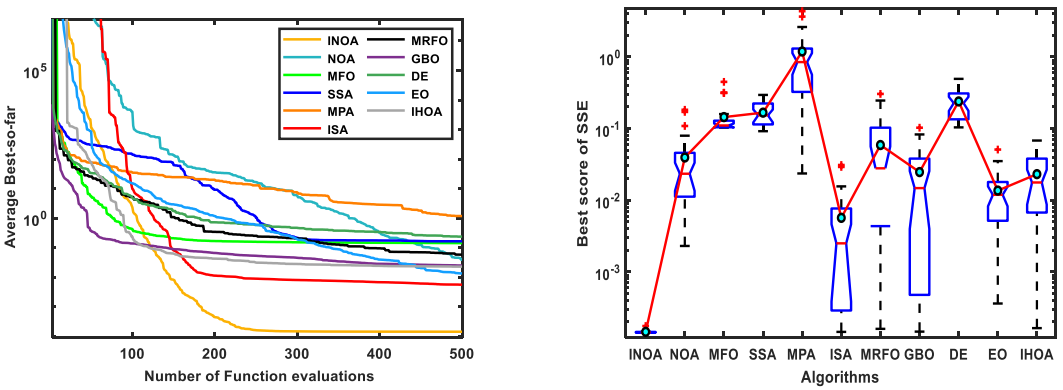
In this section, we will compare the performance of the proposed optimizer to that of its competitors using another PEMFC stack, namely Ballard Mark V, which is frequently used in the literature [28,87,94,111–115]. The characteristics of this stack are as follows [116]: $N_{cells} = 35, J_{max}\left(\frac{mA}{cm^2}\right) = 1500, A\,(cm^2) = 50.6, P_{O_2}(atm) = 1.0, l\,(\mu m) = 178, T_{fc}(K) = 343$, and $P_{H_2}(atm) = 1.0$. In Table 11, the results of 30 independent runs are presented, including the performance indicators mentioned earlier, optimal parameter values, and the Wilcoxon rank-sum test. The table demonstrates that INOA outperforms other competing algorithms across a wide range of metrics. This distinction is supported by the p-value obtained from the Wilcoxon rank-sum test, which is also reported in the table. The convergence rates achieved by each method on this specific test case are visually depicted in Fig. 16(a), providing an additional measure of performance. The figure clearly illustrates that INOA significantly outperforms its competitors in terms of speed. Furthermore, Fig. 16(a), displays a boxplot with a five-number summary of the outcomes obtained by each method, further confirming the superiority of INOA over other approaches.

6.2.5. SR-12 PEMFC stack

In this part, the algorithm's efficacy is further evaluated with an additional widely-used PEMFC stack, SR-12, with the following characteristics: $N_{cells} = 48, P_{O_2}(atm) = 0.2095, l\,(\mu m) = 25, J_{max}\left(\frac{mA}{cm^2}\right) = 672, T_{fc}(K) = 343, A\,(cm^2) = 62.5$, and $P_{H_2}(atm) = 1.47628$ [28,116–119]. Table 12 displays the outcomes of 30 independent runs for each method, including results for the aforementioned performance metrics, optimal parameter values, and Wilcoxon rank-sum tests. As can be seen from the comparison table below, INOA performs better than its competitors for all performance metrics. In the same table, the p-value from the Wilcoxon rank-sum test reveals that INOA is significantly different from the other optimizers. A further measure of performance is provided by displaying the rates of convergence achieved by each method in this test case, as shown in Fig. 17(a). This figure demonstrates that INOA is faster than all the rival optimizers. The superiority of INOA is confirmed by a boxplot in Fig. 17(b), which displays a five-number summary of the results achieved by each algorithm.

Table 12
Performance evaluation on SR-12 stack.

	$\xi_1(V)$	$\xi_2(V/k)$	$\xi_3(V/k)$	$\xi_4(V/k)$	λ	$R_c(\Omega)$	β	*SSE (best)*	*SSE (Avg)*	*SSE (Worst)*	*SSE (SD)*	*p-value*
INOA	**−1.0641**	**3.07E-03**	**3.93E-05**	**−1.06E-04**	**21.5177**	**2.73E-04**	**0.15001**	**1.4210E-04**	**1.4482E-04**	**1.7467E-04**	**6.145E-06**	
NOA	−1.0018	3.63E-03	7.99E-05	−1.06E-04	14.2199	1.82E-04	0.14832	2.2926E-03	3.9442E-02	1.8278E-01	4.482E-02	**3.02E-11**
MFO	−1.0365	2.92E-03	3.60E-05	−9.54E-05	15.7556	8.00E-04	0.14314	1.0304E-01	1.4450E-01	4.4878E-01	8.515E-02	**3.02E-11**
SSA	−1.0172	2.92E-03	3.91E-05	−9.62E-05	13.7118	7.96E-04	0.14164	9.1869E-02	1.6677E-01	2.9486E-01	6.196E-02	**3.02E-11**
MPA	−0.8534	2.38E-03	3.76E-05	−1.03E-04	22.9992	4.31E-04	0.14996	2.3594E-02	1.1917E+00	5.3421E+00	1.275E+00	**3.02E-11**
ISA	−1.0473	3.64E-03	7.31E-05	−1.06E-04	18.7773	2.28E-04	0.14985	1.4463E-04	5.6636E-03	3.0966E-02	8.050E-03	**2.61E-10**
MRFO	−0.8532	2.35E-03	3.60E-05	−1.06E-04	20.4930	2.49E-04	0.15010	1.5950E-04	5.8870E-02	3.0440E-01	7.573E-02	**3.34E-11**
GBO	−1.1997	4.44E-03	9.07E-05	−1.06E-04	22.9427	2.95E-04	0.15002	1.4589E-04	2.4672E-02	1.0271E-01	2.825E-02	**5.49E-11**
DE	−1.0784	3.05E-03	3.60E-05	−9.54E-05	13.2577	8.00E-04	0.14120	1.0384E-01	2.3918E-01	4.9404E-01	1.122E-01	**3.02E-11**
EO	−0.9243	3.15E-03	6.73E-05	−1.06E-04	18.7067	2.67E-04	0.14932	3.6024E-04	1.3605E-02	5.0885E-02	1.177E-02	**3.02E-11**
IHOA	−0.9868	3.40E-03	6.99E-05	−1.07E-04	20.8479	2.55E-04	0.15004	1.6325E-04	2.3076E-02	6.7990E-02	2.015E-02	**3.34E-11**



a) Convergence curve b) Boxplot

Fig. 17. Various comparisons among algorithms using SR-12 stack module.

6.3. *Application III: Multilevel thresholding image segmentation problem*

In this section, an additional optimization problem, namely a multithresholding image segmentation problem, with discrete variables is employed to investigate the performance of the proposed algorithm. Twenty test images taken from the Berkeley Segmentation Dataset (BSDS500) are used to validate the performance of INOA and the rival optimizers [120]. The original images and the histogram of some of these images are shown in Fig. 18(a–p). Several robust image segmentation algorithms, such as WOA [121], SMA [122], DE [123], TLBO, GWO [124], NOA [96], EO [2], MPA [125], and IMPA [125], are used to reveal the effectiveness of the proposed INOA. The controlling parameters of those rival algorithms are set to the recommended values in the cited papers, except t_{max} and N, which are set to 50 and 25 to ensure a fair comparison. Table 13 shows the parameters of the rival optimizers as recommended in the cited papers.

6.3.1. *Performance metrics*

Several performance metrics are employed in this study to observe the quality of the segmented images compared to the original images. The first metric is the peak signal-to-noise ratio (PSNR), which is used to compute the mean square error (MSE) between the segmented and original images [127]. PSNR does not consider the image's structure, so the structural similarity index (SSIM) was designed to remedy this shortcoming by comparing segmented and original images based on similarity, contrast distortion, and brightness [127]. The last metric is the feature similarity index (FSIM) proposed to compute the feature similarity between the original image and the segmented one [128].

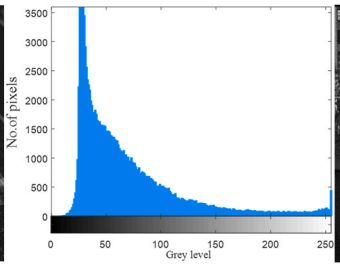
6.3.2. *INOA's parameter tuning for image segmentation*

To optimize INOA's performance when dealing with the multithresholding picture segmentation problem, it is necessary to first precisely identify four newly proposed regulating parameters. These are estimated here by conducting extensive experiments with a wide range of values on the test image: image_1 with a threshold level of 10, as shown in Fig. 19(a–d). Three of these parameters: δ, δ_1 and σ are for the newly proposed CIS. According to Fig. 19(b–d), we can deduce that the best values for these parameters are 0.05, 0.5, and 10, respectively. There is another parameter, namely P_{se} responsible for determining the probability of executing the classical NOA during the optimization parameters. To estimate the optimal value for this parameter, several experiments under various values have been conducted and their outcomes are reported in Fig. 19(a). This figure shows that the optimal value for this parameter is 10. Regarding the other parameters of the classical NOA, they are set as recommended in Ref. [96].

6.3.3. *The performance evaluation of INOA and other optimizers*

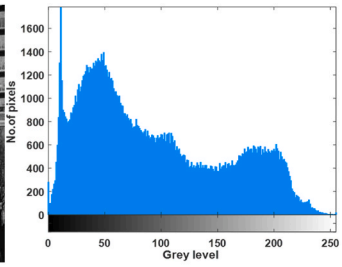
All algorithms are executed 30 independent times and the average fitness values under different Threshold levels (k), including 5-k, 6-k, 10-n, 20-n, and 30-n, for each algorithm on all test images are presented in Fig. 20. This figure shows that INOA is competitive with the majority of the others for the small threshold levels up to 10, and superior for the threshold levels greater than that. Also, Fig. 21 shows the summation of the total average fitness values for all threshold levels. The figure demonstrates the superiority of the proposed algorithm compared with the other algorithms in fitness results. In addition, Fig. 22 displays the average PSNR values for each algorithm applied to all test images, based on the various threshold levels (k). According to this figure, INOA is comparable with the vast majority of the other algorithms for the minor threshold levels up to 10, and it is superior for the threshold levels that are higher than that. Additionally, the sum of the total average PSNR values is displayed in Fig. 23 for all threshold levels on all test images. The comparison shown in this figure between the proposed method and the other algorithm reveals that the proposed algorithm is superior. Regarding the other metrics, Figs. 24–27 are presented to show the average value for these metrics obtained by each algorithm under various investigated threshold levels. Inspecting this figure reveals that INOA is on par with some of the others for the threshold smaller than or equal to 10, and superior for the threshold levels greater than that. A further comparison of performance is provided by

(a) Image_1

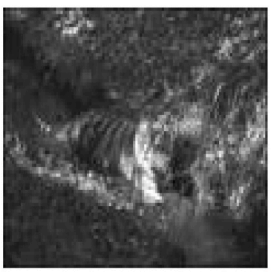


(b) Histogram of image_1

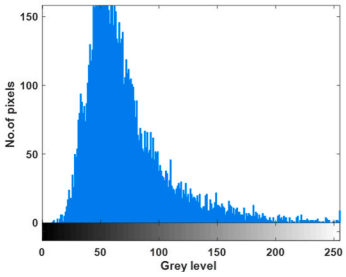
(c) Image_2



(d) Histogram of image_2

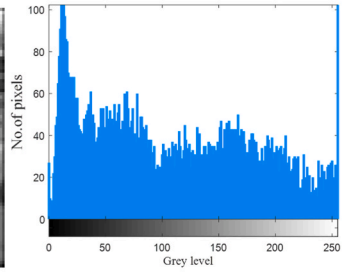


(e) Image_3



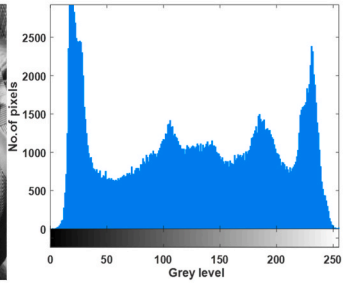
(f) Histogram of image_3

(g) Image_4



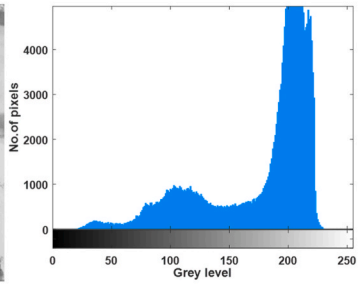
(h) Histogram of Image_4

(i) Image_5

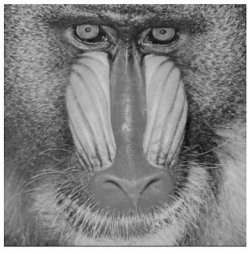


(j) Histogram of image_5

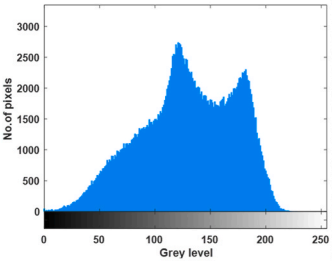
(k) Image_6



(l) Histogram of Image_6

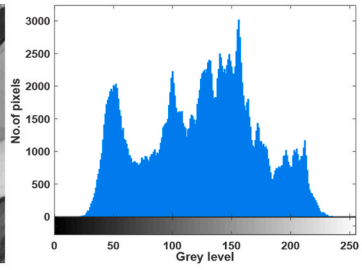


(m) Image_7



(n) Histogram of image_7

(o) Image_8

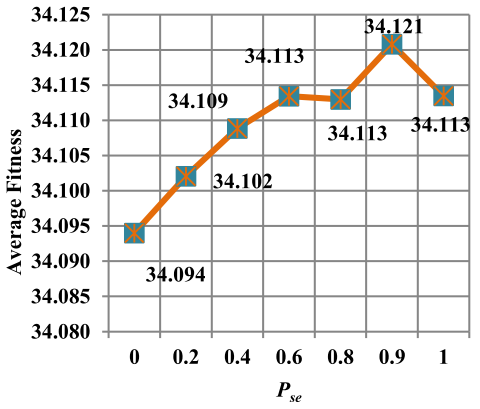


(p) Histogram of Image_8

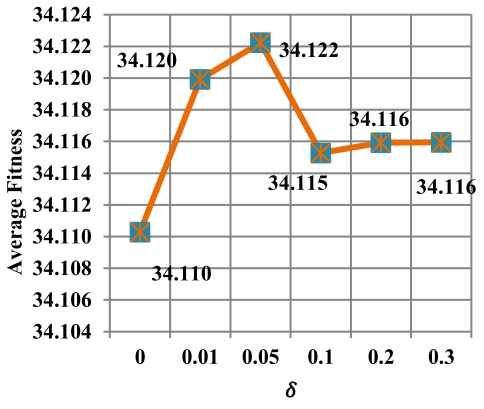
Fig. 18. Original image and its histogram.

Table 13
Controlling parameter values of rival optimizers.

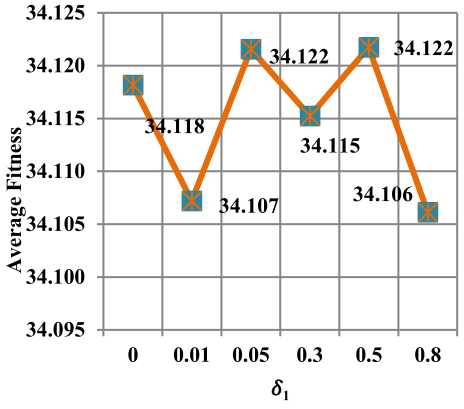
Algorithm	Parameter	Value
DE [123]	Cr	0.5
	F	0.5
WOA [121]	Constant (*a*)	Is linearly decreased from 2 to 0
GWO [124]	Self-adaptive	
SMA [122]	Constant (*z*)	0.03
EO [2]	a_1	1
	a_2	2
IMPA [125]	*P*	0.5
	FADs	0.2
	perIter	3
TLBO [126]	*TF*	[1,2]
NOA [96]	P_{rp}	0.2
	P_{a_2}	0.4
	δ	0.05
MPA [125]	*P*	0.5
	FADs	0.2



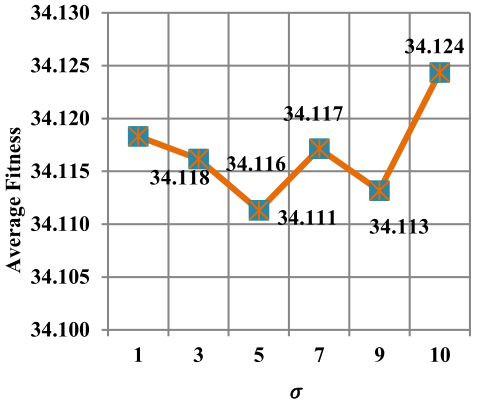
a) Adjusting the parameter P_{se}



b) Adjusting the parameter δ



c) Adjusting the parameter δ_1



d) Adjusting the parameter σ

Fig. 19. Adjusting the parameters of the proposed INOA for multilevel threshold image segmentation.

displaying the convergence rates achieved by each method on some test images under the threshold level of 30, as shown in Fig. 28 (a–c). This figure demonstrates that INOA is faster than all the rival optimizers. The superiority of INOA is confirmed by a boxplot in Fig. 29(a–c), which displays a five-number summary of the results achieved by each algorithm on some test images on the threshold level of 30.

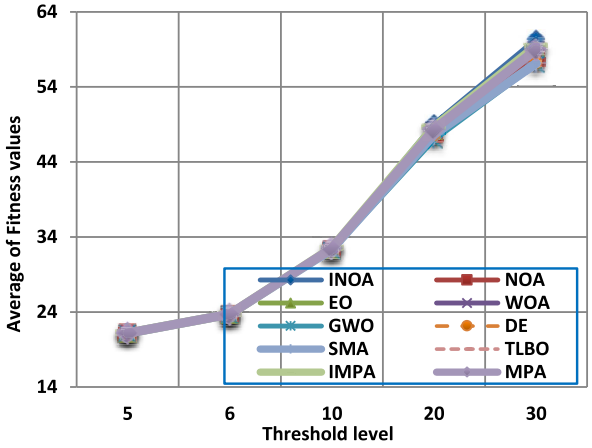


Fig. 20. Average Fitness values on each threshold level.

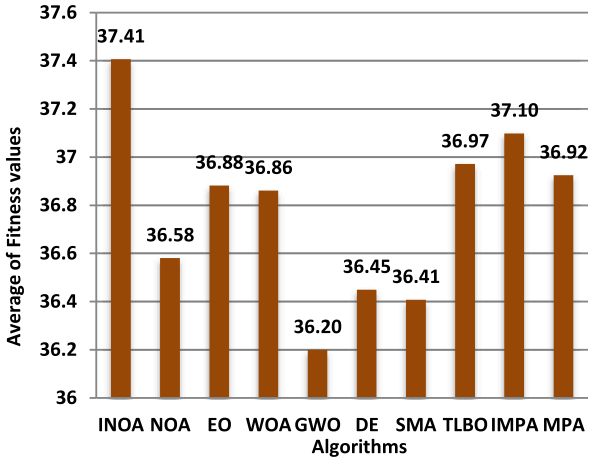


Fig. 21. Average Fitness values for all the test images.

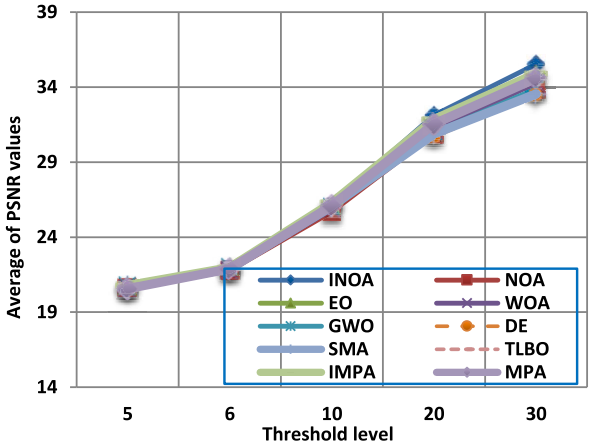


Fig. 22. Average PSNR values on each threshold level.

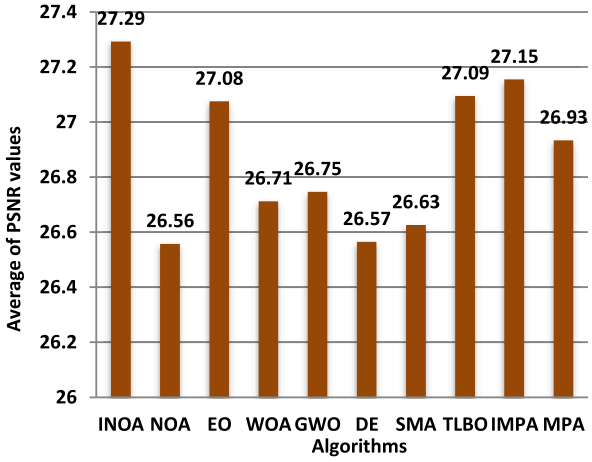


Fig. 23. Average PSNR values for all the test images.

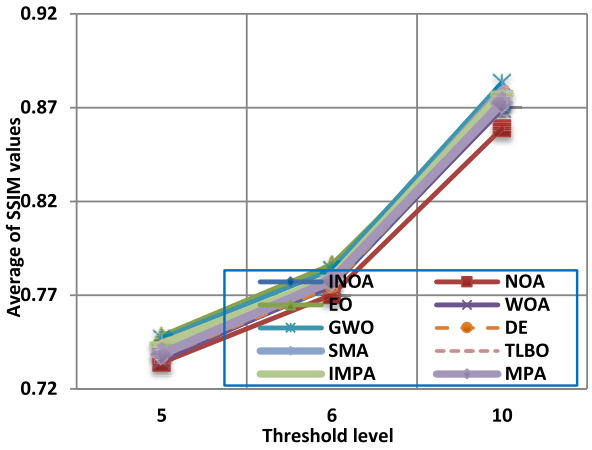


Fig. 24. Average SSIM values on the threshold levels: 5, 6, and 10.

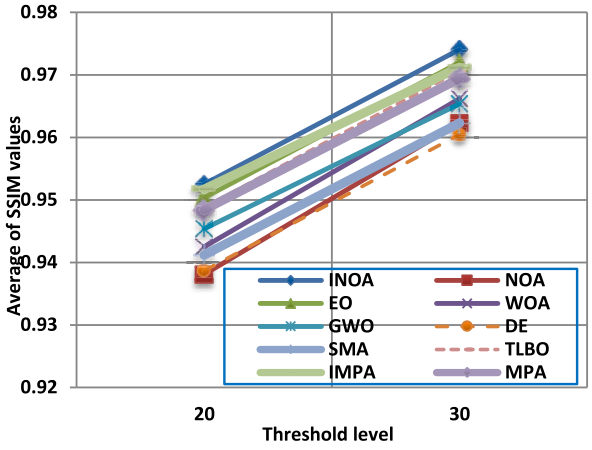


Fig. 25. Average SSIM values on the threshold levels: 20, and 30.

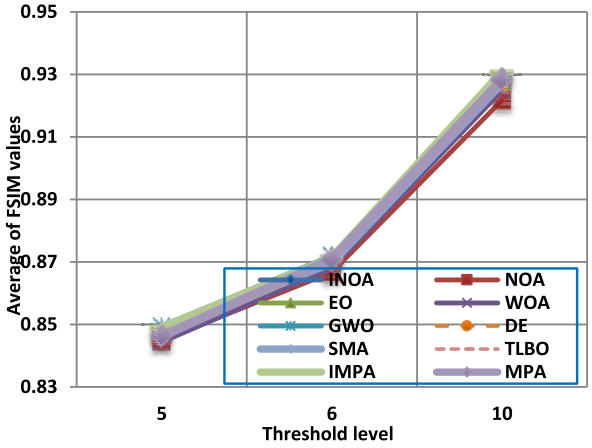


Fig. 26. Average SSIM values on the threshold levels: 5, 6, and 10.

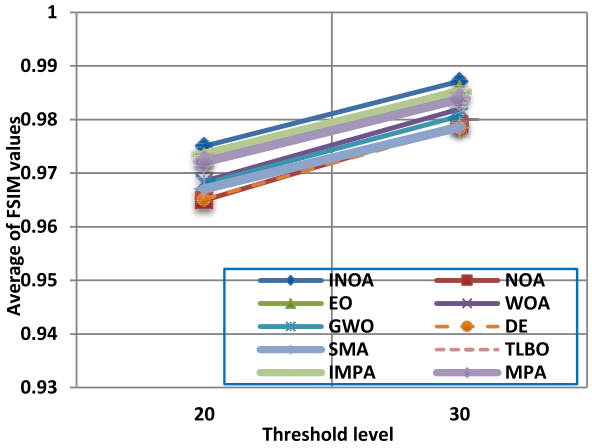
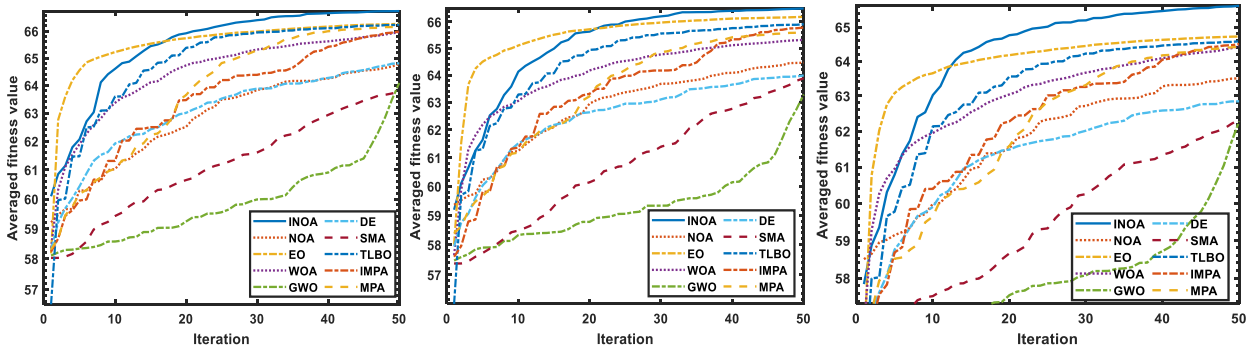
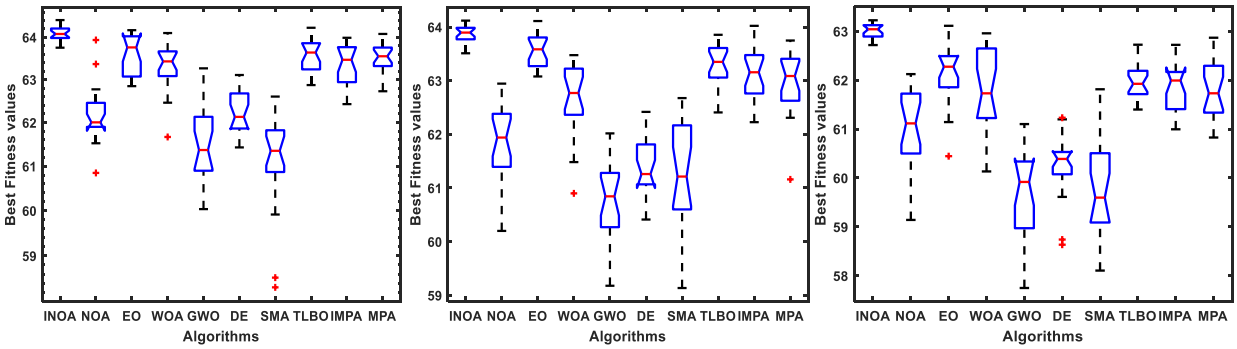


Fig. 27. Average SSIM values on the threshold levels: 20, and 30.



a) On image_1 under threshold level of 30 b) On image_2 under threshold level of 30 c) On image_3 under threshold level of 30

Fig. 28. Comparison in terms of convergence curve on some test images.



a) On image_1 under threshold level of 30 b) On image_2 under threshold level of 30 c) On image_3 under threshold level of 30

Fig. 29. Comparison in terms of Five-number summary on some test images.

7. Conclusion and future work

Solar photovoltaics (PV) and proton exchange membrane fuel cells (PEMFCs) are two well-known renewable energy sources with several benefits over other kinds of renewable energy, including no safety issues in addition to being easy to build and transport. The mathematical models of those sources contain some unknown parameters that must be accurately estimated to simulate them more accurately for generating a better amount of electricity. This problem is a complex nonlinear optimization problem with multiple local optima, thereby making it hard to solve using traditional techniques. Therefore, metaheuristic algorithms have been recently used to solve it due to their ability to solve several optimization problems. However, the existing metaheuristic algorithms for this problem still suffer from stagnation in local optima and slow convergence speed. In addition, there is another essential problem, known as the multithresholding image segmentation problem that needs to be addressed for accurately segmenting various objects within the images, thereby improving final precision in image processing and computer vision. The existing metaheuristic algorithms could not achieve accurate results for this problem, especially with increasing threshold levels. So, this paper presents a strong alternative metaheuristic algorithm based on the improved nutcracker optimization algorithm (INOA) to accurately solve the parameter estimation problem of PEMFC and PV models as well as the multilevel image segmentation problem. INOA uses an effective method that hastens the convergence rate and prevents stagnation in local minima. Four commercial PEMFC stacks, the multi-thresholding image segmentation problem, and several PV modules with single, double, and third diode models are used to test and verify the performance of INOA when applied to the discrete and continuous problems. In addition, this algorithm is extensively compared to several metaheuristic algorithms to expose its effectiveness and efficiency for each solved problem. The experimental results show that INOA is a potent substitute for solving discrete and continuous optimization issues. Future work involves applying this technique to additional optimization issues, such as the 0–1 knapsack problem and feature selection.

Ethics approval and consent to participate

The results/data/figures in this manuscript have not been published elsewhere, nor are they under consideration by another publisher. All the material is owned by the authors, and/or no permissions are required.

Consent for publication

This article does not contain any studies with human participants or animals performed by any of the authors.

Availability of data and materials

The data that support the findings of this study are available from the corresponding author upon reasonable request.

Funding

This research is supported by the Researchers Supporting Project number (RSP2024R389), King Saud University, Riyadh, Saudi Arabia.

CRediT authorship contribution statement

Mohamed Abdel-Basset: Writing – review & editing, Writing – original draft, Validation, Software, Methodology, Investigation, Conceptualization, Funding acquisition, Resources, Supervision, Visualization. **Reda Mohamed:** Writing – review & editing, Writing –

original draft, Software, Methodology, Investigation, Conceptualization, Funding acquisition, Resources, Supervision, Validation, Visualization. **Ibrahim M. Hezam:** Writing – review & editing, Writing – original draft, Funding acquisition, Conceptualization, Investigation, Methodology, Resources, Software, Supervision, Validation, Visualization. **Karam M. Sallam:** Writing – review & editing, Writing – original draft, Methodology, Investigation, Funding acquisition, Validation, Visualization. **Ibrahim A. Hameed:** Writing – review & editing, Writing – original draft, Funding acquisition, Investigation, Methodology, Validation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This research is supported by the Researchers Supporting Project number (RSP2024R389), King Saud University, Riyadh, Saudi Arabia.

References

[1] S. Korkmaz, et al., An artificial algae algorithm for solving binary optimization problems 9 (7) (2018) 1233–1247.
[2] M. Abdel-Basset, V. Chang, R. Mohamed, A novel equilibrium optimization algorithm for multi-thresholding image segmentation problems, Neural Comput. Appl. (2020) 1–34.
[3] M. Abdel-Basset, R. Mohamed, M. Abouhawwash, Balanced multi-objective optimization algorithm using improvement based reference points approach, Swarm Evol. Comput. (2020) 100791.
[4] F.A. Hashim, et al., Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems 51 (3) (2021) 1531–1551.
[5] S. Mirjalili, et al., Multi-verse optimizer: a nature-inspired algorithm for global optimization 27 (2) (2016) 495–513.
[6] J.H.J.S.a. Holland, Genetic algorithms 267 (1) (1992) 66–73.
[7] X. Yao, Y. Liu, G. Lin, Evolutionary programming made faster 3 (2) (1999) 82–102.
[8] W. Zhao, et al., Artificial Hummingbird Algorithm: A New Bio-Inspired Optimizer with its Engineering Applications, vol. 388, 2022 114194.
[9] B. Abdollahzadeh, F. Soleimanian Gharehchopogh, S. Mirjalili, Artificial gorilla troops optimizer: a new nature-inspired metaheuristic algorithm for global optimization problems 36 (10) (2021) 5887–5958.
[10] Y. Shi, Brain storm optimization algorithm, in: International Conference in Swarm Intelligence, Springer, 2011.
[11] A. Naik, S.C.J.S.C. Satapathy, Past present future: a new human-based algorithm for stochastic optimization 25 (20) (2021) 12915–12976.
[12] W. Zhang, N. Wang, S.J.I.j.o.h.e. Yang, Hybrid artificial bee colony algorithm for parameter estimation of proton exchange membrane fuel cell 38 (14) (2013) 5796–5806.
[13] A.J.I.j.o.h.e. Askarzadeh, Parameter estimation of fuel cell polarization curve using BMO algorithm 38 (35) (2013) 15405–15413.
[14] J.C. Amphlett, et al., Performance modeling of the Ballard Mark IV solid polymer electrolyte fuel cell: I. Mechanistic model development 142 (1) (1995) 1.
[15] K. Yao, et al., A review of mathematical models for hydrogen and direct methanol polymer electrolyte membrane fuel cells 4 (1-2) (2004) 3–29.
[16] D. Cheddie, N. Munroe, Review and comparison of approaches to proton exchange membrane fuel cell modeling 147 (1–2) (2005) 72–84.
[17] N. Rajasekar, et al., Comparative study of PEM fuel cell parameter extraction using Genetic Algorithm, Ain Shams Eng. J. 6 (4) (2015) 1187–1194.
[18] R.F. Mann, et al., Development and application of a generalised steady-state electrochemical model for a PEM fuel cell 86 (1–2) (2000) 173–180.
[19] S. Xu, Y. Wang, 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.
[20] G.M. Sarhan, A.M. Agwa, A.A. El-Fergany, Steady-state modeling of fuel cells based on atom search optimizer, Energies 12 (10) (2019).
[21] Y. Chen, N. Wang, Cuckoo search algorithm with explosion operator for modeling proton exchange membrane fuel cells, International Journal of Hydrogen Energy 44 (5) (2019) 3075–3087.
[22] D. Miao, et al., Parameter estimation of PEM fuel cells employing the hybrid grey wolf optimization method, Energy 193 (2020) 116616.
[23] A.A.Z. Diab, et al., Fuel cell parameters estimation via marine predators and political optimizers, IEEE Access 8 (2020) 166998–167018.
[24] R. Abbassi, et al., Accurate key parameters estimation of PEMFCs' models based on dandelion optimization algorithm, Mathematics 11 (6) (2023) 1298.
[25] B. Zhang, et al., Parameter identification of proton exchange membrane fuel cell based on swarm intelligence algorithm, Energy (2023) 128935.
[26] H.M. Sultan, et al., Standard and Quasi Oppositional bonobo optimizers for parameter extraction of PEM fuel cell stacks, Fuel 340 (2023) 127586.
[27] Z. Yuan, W. Wang, H. Wang, Optimal parameter estimation for PEMFC using modified monarch butterfly optimization 44 (11) (2020) 8427–8441.
[28] F. Qin, et al., Parameter estimation of PEMFC based on improved fluid search optimization algorithm, Energy Rep. 6 (2020) 1224–1232.
[29] Z. Yang, et al., Model Parameter Estimation of the PEMFCs Using Improved Barnacles Mating Optimization Algorithm, vol. 212, 2020 118738.
[30] M. Abdel-Basset, et al., Developments on metaheuristic-based optimization for numerical and engineering optimization problems: analysis, design, validation, and applications, Alex. Eng. J. 78 (2023) 175–212.
[31] S.M. Ebrahimi, S. Hasanzadeh, S. Khatibi, Parameter identification of fuel cell using Repairable Grey Wolf Optimization algorithm, Appl. Soft Comput. 147 (2023) 110791.
[32] A. Shaheen, et al., Fuel-cell parameter estimation based on improved gorilla troops technique, Sci. Rep. 13 (1) (2023) 8685.
[33] A. Fathy, A. Alanazi, An efficient white shark optimizer for enhancing the performance of proton exchange membrane fuel cells, Sustainability 15 (15) (2023) 11741.
[34] M. Hou, et al., A new optimum technique for parameter identification of the proton exchange membrane fuel cells based on improved remora optimizer, Energy Sources, Part A Recovery, Util. Environ. Eff. 45 (1) (2023) 3019–3040.
[35] S. Ferahtia, et al., Red-tailed hawk algorithm for numerical optimization and real-world problems, Sci. Rep. 13 (1) (2023) 12950.
[36] H. Rezk, et al., Optimal Parameter Estimation Strategy of PEM Fuel Cell Using Gradient-Based Optimizer, vol. 239, 2022 122096.
[37] E. El-Hay, M. El-Hameed, A. El-Fergany, Optimized parameters of SOFC for steady state and transient simulations using interior search algorithm, Energy 166 (2019) 451–461.
[38] M. Abdel-Basset, et al., Improved Meta-Metaheuristic Algorithms for Optimal Parameters Selection of Proton Exchange Membrane Fuel Cells: A Comparative Study, 2023.
[39] A. Fathy, H. Rezk, H.S.M. Ramadan, Recent moth-flame optimizer for enhanced solid oxide fuel cell output power via optimal parameters extraction process, Energy 207 (2020) 118326.
[40] A.A. El-Fergany, Extracting optimal parameters of PEM fuel cells using salp swarm optimizer, Renew. Energy 119 (2018) 641–648.
[41] Y. Wang, A.O. Rousis, G. Strbac, On microgrids and resilience: a comprehensive review on modeling and operational strategies, Renew. Sustain. Energy Rev. 134 (2020) 110313.

[42] I.A. Ibrahim, et al., An improved wind driven optimization algorithm for parameters identification of a triple-diode photovoltaic cell model, Energy Convers. Manag. 213 (2020) 112872.
[43] P. Bharadwaj, B. Lehman, Modelling flexible a-si pv for increased energy capture and improved reliability, in: 2020 IEEE Applied Power Electronics Conference and Exposition (APEC), IEEE, 2020.
[44] M. Qaraad, et al., Quadratic interpolation and a new local search approach to improve particle swarm optimization: solar photovoltaic parameter estimation, Expert Syst. Appl. 236 (2024) 121417.
[45] F. Belabbes, et al., Using the snake optimization metaheuristic algorithms to extract the photovoltaic cells parameters, Energy Convers. Manag. 292 (2023) 117373.
[46] M. Lakshmanan, C. Kumar, J.S. Jasper, Optimal parameter characterization of an enhanced mathematical model of solar photovoltaic cell/module using an improved white shark optimization algorithm, Optim. Control Appl. Methods 44 (2023) 2374–2425.
[47] M. Madhiarasan, D.T. Cotfas, P.A. Cotfas, Black widow optimization algorithm used to extract the parameters of photovoltaic cells and panels, Mathematics 11 (4) (2023) 967.
[48] M. Abdel-Basset, et al., Kepler optimization algorithm: a new metaheuristic algorithm inspired by Kepler's laws of planetary motion, Knowl. Base Syst. 268 (2023) 110454.
[49] A. Beşkirli, İ.J.E.R. Dağ, Parameter Extraction for Photovoltaic Models with Tree Seed Algorithm, vol. 9, 2023, pp. 174–185.
[50] A.R.J.S.E. Jordehi, Enhanced Leader Particle Swarm Optimisation (ELPSO): an Efficient Algorithm for Parameter Estimation of Photovoltaic (PV) Cells and Modules, vol. 159, 2018, pp. 78–87.
[51] J. Gupta, P. Nijhawan, S. Ganguli, Parameter extraction of solar PV cell models using novel metaheuristic chaotic tunicate swarm algorithm 31 (12) (2021) e13244.
[52] T.S. Ayyarao, P.P. Kumar, Parameter estimation of solar PV models with a new proposed war strategy optimization algorithm 46 (6) (2022) 7215–7238.
[53] W. Long, et al., Parameters Estimation of Photovoltaic Models Using a Novel Hybrid Seagull Optimization Algorithm, vol. 249, 2022 123760.
[54] T.S.L.V. Ayyarao, G.I. Kishore, Parameter estimation of solar PV models with artificial humming bird optimization algorithm using various objective functions, Soft Comput. (2023) 1–22.
[55] N. Rawat, et al., A new grey wolf optimization-based parameter estimation technique of solar photovoltaic. Sustainable Energy Technologies and Assessments 57 (2023) 103240.
[56] S. Saber, S. Salem, High-performance technique for estimating the unknown parameters of photovoltaic cells and modules based on improved spider wasp optimizer, SMIJ 5 (5) (Oct. 2023), https://doi.org/10.61185/SMIJ.2023.55102.
[57] P. Sharma, et al., Parameter estimation of solar PV using Ali baba and forty thieves optimization technique, Math. Probl Eng. (2022) 2022.
[58] M. Qaraad, et al., Photovoltaic parameter estimation using improved moth flame algorithms with local escape operators, Comput. Electr. Eng. 106 (2023) 108603.
[59] M. Abdel-Basset, et al., Parameter Estimation of Photovoltaic Models Using an Improved Marine Predators Algorithm, vol. 227, 2021, p. 113491.
[60] S. Saber, S. Salem, An improved light Spectrum optimizer for parameter identification of triple-diode PV model, SMIJ 4 (4) (Sep. 2023), https://doi.org/10.61185/SMIJ.2023.44105.
[61] M.A. Soliman, H.M. Hasanien, A. Alkuhayli, Marine predators algorithm for parameters identification of triple-diode photovoltaic models, IEEE Access 8 (2020) 155832–155842.
[62] C. Touabi, H. Bentarzi, Photovoltaic panel parameters estimation using grey wolf optimization technique, Engineering Proceedings 14 (1) (2022) 3.
[63] M. Mostafa, et al., A new strategy based on slime mould algorithm to extract the optimal model parameters of solar PV panel, Sustain. Energy Technol. Assessments 42 (2020) 100849.
[64] A. Elhammoudy, et al., Dandelion Optimizer algorithm-based method for accurate photovoltaic model parameter identification, Energy Convers. Manag. X (2023) 100405.
[65] J. Kuruvilla, et al., A review on image processing and image segmentation, in: 2016 International Conference on Data Mining and Advanced Computing (SAPIENCE), IEEE, 2016.
[66] R. Hu, et al., Utilizing Large Scale Vision and Text Datasets for Image Segmentation from Referring Expressions, 2016 arXiv preprint arXiv:1608.08305.
[67] A. Dirami, et al., Fast multilevel thresholding for image segmentation through a multiphase level set method, Signal Process. 93 (1) (2013) 139–153.
[68] D. Oliva, M.A. Elaziz, S. Hinojosa, Fuzzy entropy approaches for image segmentation, in: Metaheuristic Algorithms for Image Segmentation: Theory and Applications, Springer, 2019, pp. 141–147.
[69] J.N. Kapur, P.K. Sahoo, A.K. Wong, A new method for gray-level picture thresholding using the entropy of the histogram, Comput. Vis. Graph Image Process 29 (3) (1985) 273–285.
[70] N. Otsu, A threshold selection method from gray-level histograms, IEEE transactions on systems, man, and cybernetics 9 (1) (1979) 62–66.
[71] A.K. Bhandari, I.V. Kumar, A context sensitive energy thresholding based 3D Otsu function for image segmentation using human learning optimization, Appl. Soft Comput. 82 (2019) 105570.
[72] P. Upadhyay, J.K. Chhabra, Kapur's Entropy Based Optimal Multilevel Image Segmentation Using Crow Search Algorithm, vol. 97, 2020 105522.
[73] A. Sharma, et al., A novel opposition based improved firefly algorithm for multilevel image segmentation 81 (11) (2022) 15521–15544.
[74] Y. Zhou, et al., Meta-heuristic moth swarm algorithm for multilevel thresholding image segmentation 77 (18) (2018) 23699–23727.
[75] R. Ben Messaoud, 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. 45 (3) (2021) 4694–4708.
[76] S.K. Dinkar, et al., Opposition-based Laplacian Equilibrium Optimizer with Application in Image Segmentation Using Multilevel Thresholding, vol. 174, 2021 114766.
[77] H. Liang, et al., Modified Grasshopper Algorithm-Based Multilevel Thresholding for Color Image Segmentation, vol. 7, 2019, pp. 11258–11295.
[78] K.B. Resma, M.S.J.J.o.k.s.u.-c. Nair, i. sciences, Multilevel thresholding for image segmentation using Krill Herd Optimization algorithm 33 (5) (2021) 528–541.
[79] B.J. Ma, et al., Manta Ray Foraging Optimizer-Based Image Segmentation with a Two-Strategy Enhancement, 2023 110247.
[80] Q. Zhang, et al., Growth Optimizer: A Powerful Metaheuristic Algorithm for Solving Continuous and Discrete Global Optimization Problems, 2022 110206.
[81] E.H. Houssein, et al., An efficient multilevel image thresholding method based on improved heap-based optimizer, Sci. Rep. 13 (1) (2023) 9094.
[82] M.M. Emam, E.H. Houssein, R.M. Ghoniem, A modified reptile search algorithm for global optimization and image segmentation: case study brain MRI images, Comput. Biol. Med. 152 (2023) 106404.
[83] X. Yang, et al., Multi-level threshold segmentation framework for breast cancer images using enhanced differential evolution, Biomed. Signal Process Control 80 (2023) 104373.
[84] Y.T. Tan, D.S. Kirschen, N. Jenkins, A model of PV generation suitable for stability analysis, IEEE Trans. Energy Convers. 19 (4) (2004) 748–755.
[85] A. Askarzadeh, A.J.S.E. Rezazadeh, Parameter identification for solar cell models using harmony search-based algorithms 86 (11) (2012) 3241–3249.
[86] H. Nunes, et al., A new high performance method for determining the parameters of PV cells and modules based on guaranteed convergence particle swarm optimization, Appl. Energy 211 (2018) 774–791.
[87] A. Askarzadeh, A. Rezazadeh, A new heuristic optimization algorithm for modeling of proton exchange membrane fuel cell: bird mating optimizer, Int. J. Energy Res. 37 (10) (2013) 1196–1204.
[88] J.M. Corrêa, et al., An electrochemical-based fuel-cell model suitable for electrical engineering automation approach, IEEE Trans. Ind. Electron. 51 (5) (2004) 1103–1112.
[89] A. Mohammadi, et al., A novel approach for modeling the internal behavior of a PEMFC by using electrical circuits, International Journal of Hydrogen Energy 43 (25) (2018) 11539–11549.

[90] A.A. El-Fergany, Electrical characterisation of proton exchange membrane fuel cells stack using grasshopper optimiser, IET Renew. Power Gener. 12 (1) (2017) 9–17.
[91] R.F. Mann, et al., Development and application of a generalised steady-state electrochemical model for a PEM fuel cell, J. Power Sources 86 (1–2) (2000) 173–180.
[92] I.M. Saleh, R. Ali, H. Zhang, Simplified mathematical model of proton exchange membrane fuel cell based on horizon fuel cell stack, Journal of Modern Power Systems and Clean Energy 4 (4) (2016) 668–679.
[93] G. Zhang, K. Jiao, Three-dimensional multi-phase simulation of PEMFC at high current density utilizing Eulerian-Eulerian model and two-fluid model, Energy Convers. Manag. 176 (2018) 409–421.
[94] M.K. Singla, P. Nijhawan, A.S. Oberoi, Parameter estimation of proton exchange membrane fuel cell using a novel meta-heuristic algorithm, Environ. Sci. Pollut. Control Ser. (2021) 1–16.
[95] M. Outeiro, et al., A new parameter extraction method for accurate modeling of PEM fuel cells, Int. J. Energy Res. 33 (11) (2009) 978–988.
[96] M. Abdel-Basset, et al., Nutcracker Optimizer: A Novel Nature-Inspired Metaheuristic Algorithm for Global Optimization and Engineering Design Problems, 2023 110248.
[97] T. Easwarakhanthan, et al., Nonlinear minimization algorithm for determining the solar cell parameters with microcomputers, Int. J. Sol. Energy 4 (1) (1986) 1–12.
[98] M.A. Soliman, H.M. Hasanien, A.J.I.A. Alkuhayli, Marine predators algorithm for parameters identification of triple-diode photovoltaic models 8 (2020) 155832–155842.
[99] F.A. Hashim, et al., Fick's Law Algorithm: A Physical Law-Based Algorithm for Numerical Optimization, 2022 110146.
[100] A.H.J.I.t. Gandomi, Interior search algorithm (ISA): a novel approach for global optimization 53 (4) (2014) 1168–1183.
[101] S. Mirjalili, S.M. Mirjalili, A.J.A.i.e.s. Lewis, Grey Wolf Optimizer, vol. 69, 2014, pp. 46–61.
[102] C. Kumar, et al., A new stochastic slime mould optimization algorithm for the estimation of solar photovoltaic cell parameters, Optik 223 (2020) 165277.
[103] P. Trojovský, M.J.S. Dehghani, Pelican optimization algorithm: a novel nature-inspired algorithm for engineering applications 22 (3) (2022) 855.
[104] A. Faramarzi, et al., Equilibrium Optimizer: A Novel Optimization Algorithm, vol. 191, 2020 105190.
[105] O.E. Turgut, M.T. Coban, Optimal proton exchange membrane fuel cell modelling based on hybrid Teaching Learning Based Optimization–Differential Evolution algorithm, Ain Shams Eng. J. 7 (1) (2016) 347–360.
[106] A. Askarzadeh, A. Rezazadeh, A grouping-based global harmony search algorithm for modeling of proton exchange membrane fuel cell, International Journal of Hydrogen Energy 36 (8) (2011) 5047–5053.
[107] A.M. Agwa, A.A. El-Fergany, G.M. Sarhan, Steady-state modeling of fuel cells based on atom search optimizer, Energies 12 (10) (2019) 1884.
[108] A.A.Z. Diab, et al., Coyote optimization algorithm for parameters estimation of various models of solar cells and PV modules, IEEE Access 8 (2020) 111102–111140.
[109] H.M. Sultan, et al., JAYATree growth algorithm for parameter identification of proton exchange membrane fuel cell models, International Journal of Interactive Multimedia & Artificial Intelligence 6 (2) (2020).
[110] H.M. Sultan, et al., Parameter identification of proton exchange membrane fuel cells using an improved salp swarm algorithm, Energy Convers. Manag. 224 (2020) 113341.
[111] Y. Zhu, N. Yousefi, Optimal parameter identification of PEMFC stacks using adaptive sparrow search algorithm, Int. J. Hydrogen Energy 46 (2021) 9541–9552.
[112] M. Alizadeh, F. Torabi, Precise PEM fuel cell parameter extraction based on a self-consistent model and SCCSA optimization algorithm, Energy Convers. Manag. 229 (2021) 113777.
[113] M.T. Özdemir, Optimal parameter estimation of polymer electrolyte membrane fuel cells model with chaos embedded particle swarm optimization, Int. J. Hydrogen Energy 46 (2021) 16465–16480.
[114] Y. Song, X. Tan, S. Mizzi, Optimal parameter extraction of the proton exchange membrane fuel cells based on a new Harris Hawks Optimization algorithm, Energy Sources, Part A Recovery, Util. Environ. Eff. (2020) 1–18.
[115] S.I. Selem, H.M. Hasanien, A.A. El-Fergany, Parameters extraction of PEMFC's model using manta rays foraging optimizer, Int. J. Energy Res. 44 (6) (2020) 4629–4640.
[116] B. Duan, Q. Cao, N. Afshar, Optimal parameter identification for the proton exchange membrane fuel cell using Satin Bowerbird optimizer, Int. J. Energy Res. 43 (14) (2019) 8623–8632.
[117] A. Askarzadeh, A. Rezazadeh, An innovative global harmony search algorithm for parameter identification of a PEM fuel cell model, IEEE Trans. Ind. Electron. 59 (9) (2011) 3473–3480.
[118] A. Askarzadeh, Parameter estimation of fuel cell polarization curve using BMO algorithm, Int. J. Hydrogen Energy 38 (35) (2013) 15405–15413.
[119] A.S. Menesy, et al., Effective parameter extraction of different polymer electrolyte membrane fuel cell stack models using a modified artificial ecosystem optimization algorithm, IEEE Access 8 (2020) 31892–31909.
[120] University of California, B. https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html.
[121] S. Mirjalili, A.J.A.i.e.s. Lewis, The Whale Optimization Algorithm, vol. 95, 2016, pp. 51–67.
[122] S. Li, et al., Slime Mould Algorithm: A New Method for Stochastic Optimization, vol. 111, 2020, pp. 300–323.
[123] R. Storn, K.J.J.o.g.o. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces 11 (4) (1997) 341–359.
[124] B. Wang, et al., Robust grey wolf optimizer for multimodal optimizations: a cross-dimensional coordination approach 92 (3) (2022) 1–30.
[125] M. Abdel-Basset, et al., A hybrid COVID-19 detection model using an improved marine predators algorithm and a ranking-based diversity reduction strategy, IEEE Access 8 (2020) 79521–79540.
[126] H.S. Gill, et al., Teaching-learning-based optimization algorithm to minimize cross entropy for Selecting multilevel threshold values 20 (1) (2019) 11–25.
[127] A. Hore, D. Ziou, Image quality metrics: PSNR vs. SSIM, in: 2010 20th International Conference on Pattern Recognition, IEEE, 2010.
[128] L. Zhang, et al., FSIM: a feature similarity index for image quality assessment, IEEE Trans. Image Process. 20 (8) (2011) 2378–2386.