Ionics
https://doi.org/10.1007/s11581-024-05963-x

RESEARCH

A cooperative strategy-based differential evolution algorithm for robust PEM fuel cell parameter estimation

Pradeep Jangir[1,2,3,4] · Arpita[5] · Sunilkumar P. Agrawal[6] · Sundaram B. Pandya[7] · Anil Parmar[7] · Sumit Kumar[8] · Ghanshyam G. Tejani[9,10] · Laith Abualigah[11,12]

Received: 22 September 2024 / Revised: 11 November 2024 / Accepted: 19 November 2024

Abstract

Proton exchange membrane fuel cells (PEMFCs) are powered by hydrogen energy, which is valued for its renewable, safe, and efficient characteristics, and are therefore critical in sustainable electricity generation through hydrogen electrochemical conversion. Parameter estimation in PEMFCs is a challenging but critical task, since accurate modeling is directly related to cell performance optimization and reliable energy output under different operational conditions. To improve parameter estimation accuracy, a cooperative strategy-based differential evolution (CS-DE) algorithm was developed to minimize the sum of squared errors (SSE) between experimental and simulated PEMFC voltage data for multiple BCS 500-W PEM, BCS 250-W PEM, Nedstack PS6 PEM, 500W SR-12 PEM, H-12 PEM, and HORIZON 500W PEMFC models. The CS-DE algorithm was benchmarked against standard differential evolution (DE) and other conventional methods on six commercial PEMFC types, resulting in a 15% reduction in SSE and an average improvement of 12% in estimation accuracy. These results demonstrate the robustness and adaptability of CS-DE for complex PEMFC modeling tasks.

Keywords Parameter estimation · Proton exchange membrane fuel cell · PEMFC · Differential evolution · Optimization

Abbreviations

CS-DE	Cooperative strategy differential evolution
PEMFC	Proton exchange membrane fuel cell

✉ Pradeep Jangir
pkjmtech@gmail.com

Arpita
apyjangid@gmail.com

Sunilkumar P. Agrawal
sunil11187@gmail.com

Sundaram B. Pandya
sundarampandya@gmail.com

Anil Parmar
apkjp7557@gmail.com

Sumit Kumar
sumit21sep1990@gmail.com

Ghanshyam G. Tejani
pshyam23@gmail.com

Laith Abualigah
aligah.2020@gmail.com

[1] University Centre for Research and Development, Chandigarh University, Gharuan, Mohali 140413, India

[2] Department of CSE, Graphic Era Hill University, Dehradun 248002, India

[3] Department of CSE, Graphic Era Deemed To Be University, Dehradun 248002, Uttarakhand, India

[4] Applied Science Research Center, Applied Science Private University, Amman 11931, Jordan

[5] Department of Biosciences, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai 602 105, India

[6] Department of Electrical Engineering, Government Engineering College, Gandhinagar-382028, Gujarat, India

[7] Department of Electrical Engineering, Shri K.J. Polytechnic, Bharuch 392 001, India

[8] Australian Maritime College, College of Sciences and Engineering, University of Tasmania, Launceston, Australia

[9] Department of Industrial Engineering and Management, Yuan Ze University, Taoyuan City, Taiwan

[10] Jadara Research Center, Jadara University, Irbid, 21110, Jordan

[11] Computer Science Department, Al Al-Bayt University, Mafraq 25113, Jordan

[12] Centre for Research Impact & Outcome, Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura, 140401, Punjab, India

Published online: 02 December 2024

PaDE	Parameter adaptive differential evolution
Di-DE	Distance-based differential evolution
LSHADE	L-SHADE algorithm (not expanded in text)
NDE	Nonlinear differential evolution
PalmDE	Palm differential evolution
PSO-DE	Particle swarm optimization differential evolution
jSO	Self-adaptive differential evolution with one population
LPalmDE	Laplace palm differential evolution
HARD-DE	Hard differential evolution
V-I	Voltage-current
P-V	Power-voltage
SSE	Sum of squared errors
AE	Absolute error
RE	Relative error
MBE	Mean bias error
H_2	Molecular hydrogen
H^+	Proton
e^-	Electron
O_2	Molecular oxygen
V_{cell}	Output voltage of the fuel cell
E_{Nernst}	Reversible equilibrium voltage
V_{act}	Activation overpotential
V_{ohm}	Ohmic voltage drop
V_{con}	Concentration overpotential
E_0	Standard potential of the hydrogen/oxygen reaction (1.229 V)
T	Operating temperature
$P^*_{H_2}$	Partial pressure of hydrogen
$P^*_{O_2}$	Partial pressure of oxygen
R_{H_a}	Relative humidity of vapor in the anode
$P^*_{H_2O}$	Saturation pressure of water vapor
P_a	Inlet pressure of the anode
P_c	Inlet pressure of the cathode
A	Active area of the membrane
$\xi_1, \xi_2, \xi_3, \xi_4$	Parametric coefficients for cell model calculations
i	Cell current
$C^*_{O_2}$	Concentration of dissolved oxygen at the liquid interface
R_M	Membrane equivalent resistance
R_C	Electron-transfer equivalent resistance
ρ_M	Membrane resistivity for electron flow
l	Thickness of the membrane
λ	Water content of the membrane
b	Parametric coefficient for concentration overpotential
I	Actual current density
I_{max}	Maximum current density
V_{stack}	Output voltage of the PEMFC stack
n	Number of fuel cells in the PEMFC stack
V_{sm}	Output voltage of the actual PEMFC stack
V_{so}	Model output voltage
N	Number of experimental data points
F_b	Modified scale factor for mutation strategy
X^p_{best}	Best solution vector
$X_{r_1,G}, X_{r_2,G}$	Randomly chosen vectors for mutation
PS	Population size
R_{DP}	Diversity metric ratio
X_{min}, X_{max}	Minimum and maximum boundaries of the solution space
μ_F	Mean of the scale factor
μ_{CR}	Mean of the crossover rate
$f(X)$	Fitness value of the solution X
F_w	Inertia weight factor

Introduction

Scarcity of fossil fuels and global pollution are two problems that necessitate the use of clean energy technologies. Proton exchange membrane fuel cells (PEMFCs) are among these technologies which are a good example of electrochemical energy conversion devices [1]. They serve as an environmentally friendly substitute for diesel in distributed generation, providing back-up power effectively and stabilizing grid electricity. PEMFCs have been highly regarded in the field of electricity supply because they have low greenhouse gas emissions, instantly react to load variations, operate quietly, and have dependable startup mechanisms [1–4]. Determining fuel cell parameters with greater accuracy is becoming increasingly important. Thus, PEMFCs are one of the best clean energy technologies as they change chemical energy into electrical power directly through electrochemical reactions [5–8]. For this reason, they can be used for numerous applications because their strengths lie in high electrical efficiency, low emissions, and the ability to use different types of fuels [9, 10]. Notably, this poses a complicated problem since the system is complicated and interconnected strongly. Several methods for accurately estimating these vital parameters are available in scientific literature.

Several ways of finding the parameters of a proton exchange membrane fuel cell (PEMFC) stack have been presented in scholarly works. These ways can broadly be divided into iterative or intelligent methods. Parameter estimation is normally effectively achieved through iterative practices, which include the recognized Levenberg–Marquardt (LM) algorithm. However, when initial conditions are not set optimally, such methods may suffer from slow convergence and/or convergence to local minima. The LM algorithm that calculates partial derivatives with respect to the parameters can be computationally expensive and less efficient for non-converging data, outliers or large datasets. Conversely, smart approaches using AI have evolved as viable replacements

since they enable accurate estimation of model parameters for PEMFC models. On the other hand, an optimized version of Levenberg–Marquardt backpropagation (LMBP) algorithm is widely acknowledged for its remarkable optimization advantages associated with artificial neural networks. Nevertheless, the extent of the improvements over conventional approaches and the relevance of these improvements to practical scenarios deserve further elaboration and study [11].

Meta-heuristics have been preferred for their effectiveness, convenience in application, and simplicity to generate heat and electricity without employing derivatives. Much emphasis has been made in improving the estimation of parameters of proton exchange membrane fuel cells (PEMFCs). Some of the examples are particle swarm optimization (PSO) technique [12], hybrid adaptive differential evolution approach [13], and grasshopper optimization algorithm specifically for PEMFCs [14]. In addition, Zhou presented multi-verse optimizer in [15], and a hybrid artificial bee colony method was described in [16] for the similar parameter estimation problem.

Subsequent studies have employed a number of other algorithms including, but not limited to, genetic algorithm (GA) [17], particle swarm optimization (PSO) [18], differential evolution (DE) [19, 20], seeker optimization algorithm (SOA) [21], bio-inspired p systems-based optimization algorithm (BIPOA) [22], adaptive RNA genetic algorithm (ARNA-GA) [23], and circular RNA genetic algorithm (cRNA-GA) [24]. However, these methods have their own advantages and disadvantages that go with them. Some methods may become stuck in local optima, but they are capable of at least remembering the previous best solutions. In contrast, some algorithms might have challenges in reproducing previous outcomes even as they have the potential of solving simple mathematical operations. However, there are some algorithms that are unable to perform well in situations where they are exposed to frequent changes in parameter settings with numerous iterations, intricate algorithmic frameworks, and meagre stochasticity.

These challenges make it important to continue work on enhancing or developing better algorithms. New optimization methods may possess merits such as rapid convergence, reduced computational load, and enhanced scalability for extensive or complex problem areas. These advancements may also help in improving the optimization techniques making them more noise tolerant, problem independent, and less sensitive to uncertainties.

Cooperative strategy-based differential evolution (CS-DE) [25] is the enhanced version of traditional DE algorithm. DE works effectively with certain mutation strategies which are different in all the forms of DE and therefore gives different results when compared on standard benchmark functions. CS-DE algorithm with dual mutation schemes was compared with 58 benchmarks of CEC2013 and CEC2014 test suites. The comparisons made in the experiment showed that CS-DE algorithm is comparable to other recent state-of-the-art DE algorithms. This paper describes a method for identifying unknown parameters in commercial PEMFCs using the CS-DE algorithm for parameter identification. Main findings of this research include the following:

(a) For this PEMFC parameter estimation problem using the CS-DE algorithm, the objective is to minimize the SSE.
(b) The study has been conducted using six commercial PEMFC stacks namely BCS500W [26], NedStackPS6 [27], S12 [27], H12 [28], HORIZON [28], and Standard250W [29] to generate I-V and P–V polarization curves at different temperature and pressure conditions, calculating the AE and RE.
(c) The study involved comparing the CS-DE with nine other programed DE variants such as PaDE [30], Di-DE [31], LSHADE [32], NDE [33], PalmDE [34], PSO-DE [35], jSO [36], LPalmDE [34], and HARD-DE [37] in estimating unknown parameters for a range of PEMFC stacks across diverse parameter variations.
(d) Statistics were applied to validate the reliability of this algorithm by using Friedman ranking, runtime, and standard deviation.

The paper is organized as shown: "Mathematical modeling of PEMFC" outlines mathematical framework related to PEMFC and objective function. " Cooperative strategy-based differential evolution (CS-DE) algorithm" provides an overview of the CS-DE algorithm. "Experimental results for PEMFC parameter optimization problem" gives details about simulations and results while "Conclusion" shows conclusion.

Mathematical modeling of PEMFC

A typical PEMFC consists of an electrolyte and two electrodes. The electrolyte allows protons to pass through while blocking electrons. Hydrogen gas passes over the anode, where it dissociates into electrons and hydrogen protons. The corresponding reactions are as follows [38, 39]:

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

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

$$H_2 + \frac{1}{2}O_2 \rightarrow H_2O + \text{Heat} \tag{3}$$

The mathematical model of PEMFC

Generally, the actual fuel cell output voltage is less than the ideal voltage due to fuel cell electrical resistance, inefficient

reactant gas transport, and low reaction rate. Hence, the actual output voltage [39] can be expressed as follows:

$$V_{\text{cell}} = E_{\text{Nernst}} - V_{\text{act}} - V_{\text{ohm}} - V_{\text{con}} \tag{4}$$

where E_{Nernst} is the reversible equilibrium voltage, and the V_{act}, V_{ohm}, and V_{con} represent the activation overpotential, the ohmic voltage drop, and the concentration overpotential, respectively.

Fuel cell open circuit voltage

The E_{Nernst} is the potential of the cell obtained in an open-circuit and can be calculated through the Nernst equation, which is shown as follows [40, 41]:

$$E_{\text{Nernst}} = E_0 - (8.5 \times 10^{-4})(T - 298.15) + (4.3085 \times 10^{-5}) \times T \times [\ln(P^*_{H_2}) + \frac{1}{2}\ln(P^*_{O_2})] \tag{5}$$

The standard potential of the hydrogen–oxygen reaction, denoted as E_0 (approximately 1.229 V), occurs at standard conditions of 298.15 K and 1-atm pressure. T represents the operational temperature of the gas, while $P^*_{H_2}$ and $P^*_{O_2}$ denote the partial pressures of hydrogen and oxygen, respectively, as outlined in Eqs. 6 and 7 [42].

$$P^*_{H_2} = 0.5RH_a \times P^*_{H_2O}\left[\left(\exp\left(\frac{1.635(i/A)}{T^{1.334}}\right) \times \frac{RH_a * P^*_{H_2O}}{P_a}\right)^{-1} - 1\right] \tag{6}$$

$$P^*_{O_2} = RH_c \times P^*_{H_2O}\left[(\exp\left(\frac{4.192(i/A)}{T^{1.334}}\right) \times \frac{RH_c * P^*_{H_2O}}{P_c}\right)^{-1} - 1\right] \tag{7}$$

where P_a and P_c are the inlet pressure of anode and cathode and RH_a and RH_c represent the relative humidity of vapor in anode and cathode. The symbol of A is the active area of the membrane and $P^*_{H_2O}$ denotes the saturation pressure of water vapor which is depicted as follows [42]:

$$\log(P^*_{H_2O}) = 2.95 \times 10^{-2}(T - 273.15) - 9.18 \times 10^{-5}(T - 273.15)^2 + 1.44 \times 10^{-7}(T - 273.15)^3 - 2.18 \tag{8}$$

Activation overpotential

The activation overpotential, V_{act}, arises from the limitations in charge transfer rates and other activation processes. It is defined as follows [41, 42]:

$$V_{act} = -\left[\xi_1 + \xi_2 T + \xi_3 T\left(\ln\left(C^*_{O_2}\right)\right) + \xi_4(\ln(i))\right] \tag{9}$$

where ξ_1, ξ_2, ξ_3, and ξ_4 are the coefficients for each fuel cell model, i is the current, and $C^*_{O_2}$ represents the oxygen concentration at the liquid interface, as per Henry law [40]:

$$C^*_{O_2} = P^*_{O_2}/(5.08 \times 10^6 \exp(-498/T)) \tag{10}$$

Ohmic voltage drop

The ohmic voltage drop results from resistance in the polymer membrane, electrodes, and the contact resistance between them. It is expressed as follows:

$$V_{\text{ohmic}} = i(R_M + R_C) \tag{11}$$

where R_M and R_C are the membrane and electron-transfer equivalent resistances, respectively. The expression for R_M is given as follows:

$$R_M = \frac{\rho_M l}{A} \tag{12}$$

where ρ_M is the membrane resistivity ($\Omega \cdot$ cm), A is the cell active area (cm^2), and l is the membrane thickness (cm). The empirical formula for ρ_M is as follows:

$$\rho_M = \frac{181.6\left[1 + 0.03\left(\frac{i}{A}\right) + 0.062\left(\frac{T}{303}\right)^2\left(\frac{i}{A}\right)^{2.5}\right]}{\left[\lambda - < spanclass =' convertEndash' > 0.634 - 3 < /span > \left(\frac{i}{A}\right)\right]\exp\left[4.18\left(\frac{T-303}{T}\right)\right]} \tag{13}$$

where λ represents the membrane water content and can be adjusted within a range [24, 39, 42].

Concentration overpotential

The concentration overpotential, caused by mass diffusion from the flow channels to the reaction sites, is represented by [42, 43]:

$$V_{con} = -b\ln(1 - \frac{I}{I_{max}}) \tag{14}$$

where b is a parametric coefficient, I is the actual current density, and I_{max} is the maximum current density.

PEMFC stack output voltage

To generate sufficient power, a stack of multiple fuel cells is assembled. The total voltage of the PEMFC stack is given by

$$V_{\text{stack}} = nV_{\text{cell}} \tag{15}$$

The PEMFC model includes several known and defined parameters, which encompass both design and operational characteristics. For design parameters, the active area A, membrane thickness l, and maximum current density $J_{\max}$ (measured in A/cm ^{2}) are predetermined. Operational parameters include the cell temperature T, partial pressures of hydrogen P_{H_2} and oxygen P_{O_2}, which are specified before conducting parameter estimation shown in Table 2. The parametric coefficient b, which varies based on the specific cell type and operating conditions, is also known to impact concentration overpotential.

The primary unknown parameters, which the optimization process aims to estimate, are the seven decision variables: ξ_1, ξ_2, ξ_3, ξ_4, λ, R_C, and b. These represent various activation, concentration, and ohmic resistance characteristics specific to the PEMFC model, affecting voltage behavior and efficiency.

Objective function

In order to identify the optimal values of the seven unknown parameters mentioned above by the optimization techniques, it needs to define an objective function to be optimized. In this work, the SSE (sum of the squared error) between the output voltage of the actual PEMFC stack and the model output voltage are used as the objective function [42]:

$$\min f(\mathbf{x}) = \sum_{k=1}^{N} (V_{\text{sm,k}} - V_{\text{so,k}})^2 \tag{16}$$

where $x = \{\xi_1, \xi_2, \xi_3, \xi_4, \lambda, R_C, B\}$ is the vector of unknown parameters, V_{sm} is the actual stack output voltage, V_{so} is the model output voltage, and N is the number of data points.

Cooperative strategy-based differential evolution (CS-DE) algorithm

Several well-known DE variants, including JADE, LSHADE, jSO, and HARD-DE, form the basis of our novel CS-DE algorithm, each contributing unique mechanisms to enhance optimization performance. Zhang et al. proposed the JADE algorithm [44], which introduced the use of an external archive and the selection of top superior individuals for mutation instead of relying solely on the global best. This innovation prevented premature convergence, and effectively balanced between exploration and exploitation, thus aiding in the preservation of population complexity. In addition, JADE used the adaptive control parameters, where the scale factor (F) was normally distributed with Cauchy distribution and the crossover rate (CR) was normally distributed. These adaptive mechanisms greatly contributed to improving its performance. However, the choice of the elite proportion parameter p in the JADE algorithm may lead to slower convergence or even to the finding of suboptimal solutions. Tanabe et al. proposed LSHADE in [32], which incorporated a learning mechanism of success history for F and CR parameters; details of success F and CR values were stored in a pool to be used later LSHADE also employed fitness difference-based weighting schemes to enhance the dynamic control of its control parameters. Prevailing population size reduction mechanism was another contribution in balancing between the exploration and exploitation by decreasing the population size in gradual manner. All the same, the use of fitness difference between generations in LSHADE makes it less effective in problems where fitness values cannot be easily derived. Brest et al. introduced jSO [36], an enhancement of LSHADE, which incorporated an inertia weight (F_w) into the mutation strategy to refine the balance between exploration and exploitation further. jSO also employed an adaptive mechanism for the elite proportion p and included modifications to the control parameters F and CR to enhance optimization performance. When using jSO the algorithm shows good performance across a broad set of test problems, however, the large number of empirical adjustments to the parameters may lead to over fitting thus reducing its applicability to other problems. Meng et al. presented the HARD-DE algorithm [37] which incorporated a hierarchical archive in which inferior and historical solutions are stored. This mutation strategy that used the archive from previous iterations enabled the algorithm to avoid early convergence. HARD-DE also included a slower population size reduction scheme to prevent early termination of the search process. However, the above hierarchical archive makes the algorithm more complex and may increase the computation time in larger problems. Although HARD-DE has a slower convergence, it has good exploration ability, but with the drawback of higher computational load.

Collectively, these DE parameters, and explore versus exploit the search space. Their advantages and disadvantages have shaped the development of the CS-DE algorithm, which aims at expanding on their concept to overcome particular difficulties in parameter estimation for such systems like PEM.

New cooperative strategy-based DE (CS-DE) presented in [25] improves the population diversity and the performance of DE variants in solving numerical optimization problems. Key innovations of this model include: First, we proposed to extend the mutation strategies set by introducing two similar strategies that can share the same set of parameters. Secondly, the adoption of a new grouping approach enables the adaptive choice of mutation strategies by different individuals, to create trial vectors. Thirdly, innovative adaptation mechanisms for the control parameters F, CR, and PS, alongside a stagnation-triggered re-initialization feature, are implemented to further refine the CS-DE algorithm.

The mutation strategy

In the CS-DE algorithm, two analogous mutation strategies are utilized, and the specific formulas for these strategies are outlined in Eq. 17.

$$\begin{cases} V_{i,G} = X_{i,G} + F_b \cdot (X^p_{best,G} - X_{i,G}) + F \cdot (X_{r_1,G} - \hat{X}_{r_2,G}) \\ V_{i,G} = X_{i,G} + F_b \cdot (X^p_{best,G} - X_{i,G}) + F \cdot (X_{r_1,G} - \widetilde{X}_{r_2,G}) \end{cases} \tag{17}$$

where X^G_{r2} and X^G_{r3} are vectors randomly chosen from the combinations $P \cup A$ and $P \cup B$, respectively, where P is the current population, A is an archive of inferior solutions, and B is another archive of historical solutions.

The concept of using an external archive for inferior solutions was originally introduced in JADE and has since proven effective. Further empirical research [44] demonstrated that an archive of historical solutions generally outperforms one of inferior solutions on numerous benchmarks. As a result, both mutation strategies are preserved in the CS-DE framework. Additionally, a minor modification is adopted in the mutation strategies F_b, where replaces F in the expression $(X^p_{best,G} - X_{i,G})$. F_b operates as a piecewise function, akin to that used in jSO, with specifics detailed in Eq. 18:

$$F_b = \begin{cases} 0.7 \cdot F, & ifnfe < 0.2 \cdot nfe_{\max} \\ 0.8 \cdot F, & if 0.2 \cdot nfe_{\max} \le nfe < 0.4 \cdot nfe_{\max} \\ F, & otherwise \end{cases} \tag{18}$$

The adaptation scheme for the ratio p, which determines the proportion of elite individuals within the entire population, follows the formulation described in Eq. 11. Additionally, a time-stamp-based mechanism [34] is integrated into both mutation strategies. This mechanism is designed to prevent outdated inferior solutions from persisting in the external archive throughout the evolutionary process.

Parameter control

Parameter control is critical for the optimization performance of an algorithm within a specified test suite, and initial settings often rely on prior knowledge of that suite. However, these settings might not be as effective for different optimization challenges. Thus, using standard initial values for these control parameters is often advisable. In our algorithm, the initial parameter settings for F and CR are consistent with those used in the canonical DE and the jDE algorithm, with initial values set at $F = 0.5$ and $\mathrm{CR} = 0.9$. Additionally, a grouping strategy from the LPALMDE algorithm [34] is adopted, creating K groups during the evolution. Each group has an equal initial selection probability of $1/K$. Although initial μ_{CR} values are uniform across groups at $\mu_{\mathrm{CR}_1} = \mu_{\mathrm{CR}_2} = \cdots = \mu_{\mathrm{CR}_j} = \cdots = \mu_{\mathrm{CR}} = 0.8$, each group maintains a unique μ_{CR} throughout the evolution. The scale factor F for each individual follows a semi-fixed Cauchy distribution $F_i \sim C(\mu_F, 0.1)$, starting with μ_F at 0.6. The crossover rate CR for individuals in each group is determined by a semi-fixed Gaussian distribution $\mathrm{CR}_j \sim N\left(\mu_{CR_j}, 0.1\right)$. Change rates for F and CR are managed with probabilities $\tau_1 = \tau_2 = 0.9$, as detailed in Eqs. 19 and 20:

$$F_{i,G+1} = \begin{cases} \mathrm{rand}\, c(\mu_F, 0.1), & \text{if } \mathrm{rand}_1 < \tau_1 \\ F_{i,G}, & \text{otherwise} \end{cases} \tag{19}$$

$$CR_{ji,G+1} = \begin{cases} \mathrm{rand}n(\mu_{CR_j}, 0.1), & \text{if } \mathrm{rand}_2 < \tau_2 \& \mu_{CR_j} \neq 0 \\ 0, & \text{if } \mathrm{rand}_2 < \tau_2 \& \mu_{CR_j} = 0 \\ \mathrm{CR}_{ji,G}, & \text{otherwise} \end{cases} \tag{20}$$

where $F_{i,G}$ represents the scale factor, and $\mathrm{CR}_{ji,G}$ indicates the crossover rate of the i^{th} individual within the population, sorted into the j^{th} group. Within these K groups, half of the individuals apply the initial mutation strategy outlined in Eq. 17; and the left individuals employ the other mutation strategy. If a better offspring (trial vector) is obtained, then

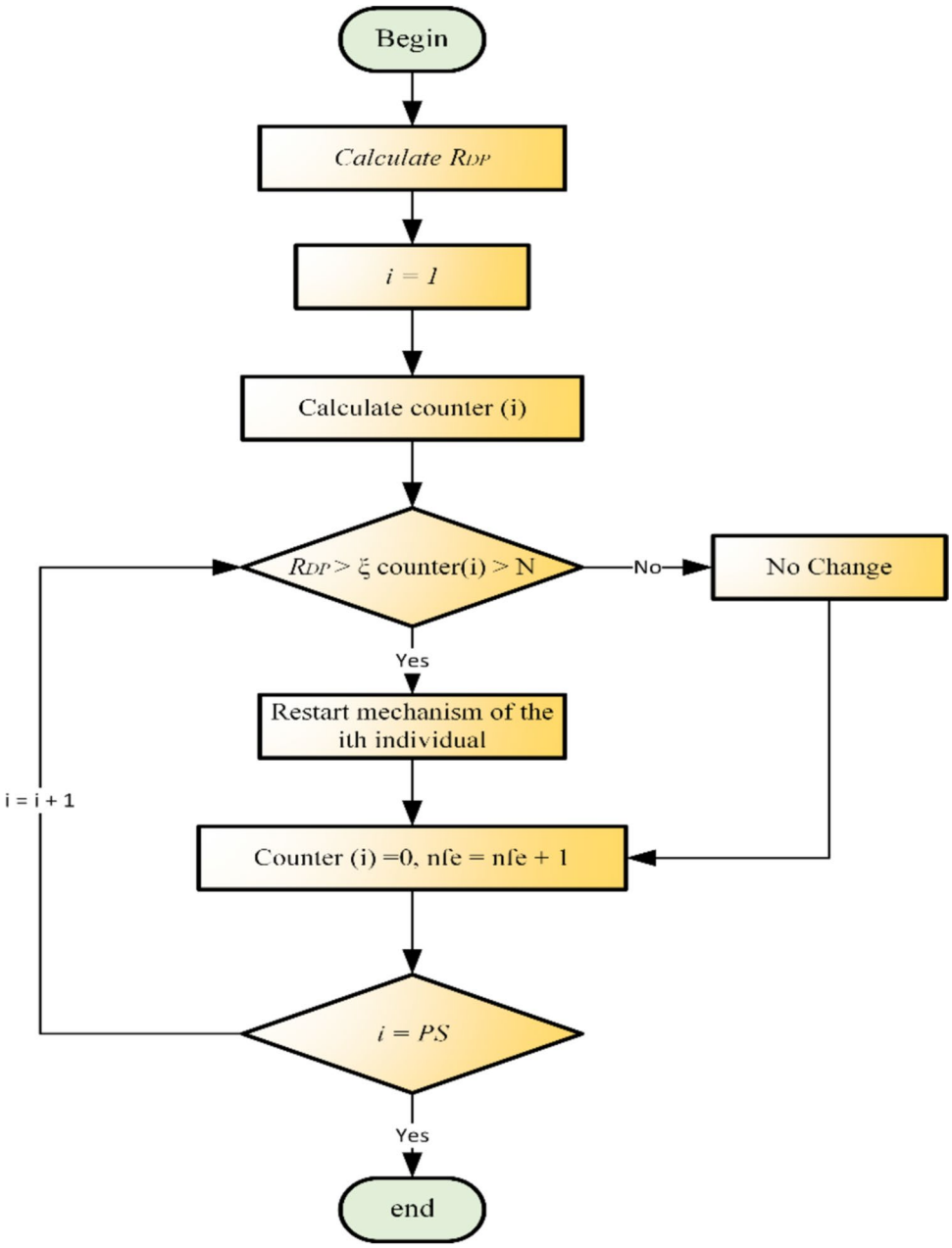

Fig. 1 Population diversity enhancement mechanism

the individual is labeled as success individual, otherwise, it is labeled as failure individual. The selection probability of each group is updated at the end of a generation according to the probability shown in Eq. 21:

$$\begin{cases} r_j = \begin{cases} \frac{ns_j^2}{ns \cdot (ns_j + nf_j)}, & if ns_j > 0 \\ \epsilon, & \text{otherwise} \end{cases} \\ P(j) = \frac{r_j}{\sum_{j=1}^{K}(r_j)} \end{cases} \tag{21}$$

where symbols retain their definitions as in LPALMDE. Subsequently, individuals within the population are redistributed to groups based on stochastic universal selection using these selection probabilities.

Additionally, at the conclusion of each generation, the μ_F and μ_{CR_j} for the j^{th} group are updated in accordance with Eqs. 22 and 23, respectively:

$$\begin{cases} \Delta f_i = f(U_{i,G}) - f(X_{i,G}) \\ w_s = \frac{\Delta f_i}{\sum_{s=1}^{|S|} \Delta f_s} \\ mean_{WL}(S_F) = \frac{\sum_{s=1}^{|S|} w_s \cdot S_F^2(s)}{\sum_{s=1}^{|S|} w_s \cdot S_F(s)} \\ c = \frac{ns}{PS} \\ \mu_F = \begin{cases} c \cdot \mu_F + (1-c) \cdot (mean_{WL}(S_F)), & if\ S \neq \varnothing \\ \mu_F & otherwise \end{cases} \end{cases} \tag{22}$$

$$\begin{cases} \Delta f_i = f(U_{i,G}) - f(X_{i,G}) \\ w_s = \frac{\Delta f_i}{\sum_{s=1}^{|S|} \Delta f_i} \\ mean_{WL}(S_{CR}) = \frac{\sum_{s=1}^{|S|} w_s \cdot S_{CR}^2(s)}{\sum_{s=1}^{|S|} w_s \cdot S_{CR}(s)} \\ \mu_{CR_{idx,G+1}} = \begin{cases} mean_{WL}(S_{CR}), & ifS \neq \varnothing \\ \mu_{CR_{idx,G}}, & otherwise \end{cases} \end{cases} \tag{23}$$

In this context, S refers to the group of successful individuals, while S_F represents the collection of F values corresponding to these successful individuals, and S_{CR} consists of the CR values pertinent to the individuals in S. The index s identifies the position within set ' (as well as within S_F and S_{CR} where each individual in ' is also identified by this s^{th} position within the overall population. The index $i^{th}$is used to denote the group that has the smallest selection probability.

Population size reduction scheme

Typically, an early sharp reduction in population size during the evolutionary process can impair the thorough exploration of the search domain, potentially diminishing optimization performance [45]. To address this, a gradual decline in population size is advised to enhance the exploration capabilities. In the CS-DE algorithm, a novel reduction strategy combining elliptic and linear methods is introduced to adjust the population size effectively. The specifics of this reduction method are outlined in Eq. 24:

$$PS = \begin{cases} \lceil \sqrt{PS_{ini}^2 - \frac{PS_{ini}^2 - y^2}{x^2} \cdot nfe^2} \rceil, & if\ nfe < x \\ \lceil \frac{PS_{\min} - y}{nfe_{\max} - x} \cdot (nfe - x) + y \rceil, & otherwise \end{cases} \tag{24}$$

In the CS-DE algorithm, the adjustment of population size is strategically depicted with the concept of a pivot at position(x, y), which serves as the junction between the elliptic and linear reduction phases. The early phase of evolution utilizes an elliptical approach to reduce the population size, transitioning to a linear method in the later stages. This adaptive strategy ensures a relatively larger initial population compared to the linear approaches used in LSHADE [32] and iSO [36]. Such a configuration allows the CS-DE algorithm to gain a more comprehensive understanding of the objective landscape, thereby enhancing overall performance.

Population diversity enhancement

Evolutionary algorithms often face stagnation during later stages due to reduced population diversity. To mitigate this, CS-DE includes a stagnation detection mechanism based on a diversity metric, DP, calculated as follows:

$$\begin{cases} \overline{X} = \frac{1}{PS} \cdot \sum_{i=1}^{PS} X_{i,G} \\ DP = \sqrt{\sum_{i=1}^{PS} \|X_{i,G} - \overline{X}\|^2} \end{cases} \tag{25}$$

Then, the stagnation indicator can be calculated via Eq. 26:

$$R_{\mathrm{DP}} = \frac{\mathrm{DP}}{\mathrm{DP}_{\mathrm{ini}}} \tag{26}$$

In this model, $\mathrm{DP}_{\mathrm{ini}}$ represents the diversity measure (DP) at the start of the evolutionary process. Additionally, a stagnation counter (ct) tracks the number of consecutive generations without improvement. If the reduction ratio of diversity R_{DP} is less than a threshold ξ and the stagnation counter exceeds a specified limit $N\left(R_{DP} < \xi \& ct > N\right)$, it indicates that the individual is experiencing stagnation. Under these conditions, and provided the individual is not the current global best, a re-initialization of selected dimensions is mandated as outlined in Eq. 27:

$$X_{i,G+1}(j) = \begin{cases} X_{\min}(j) + rand() \cdot (X_{\max}(j) - X_{\min}(j)) & if j \in R \\ X_{i,G}(j) & otherwise \end{cases} \tag{27}$$

In this scenario, R comprises variables randomly chosen from the D dimensions, and $X_{\min}$ and $X_{\max}$ represent the minimum and maximum boundaries of the solution space, respectively. Figure 1 illustrates the flowchart detailing the mechanism for enhancing population diversity, while Algorithm 1 provides the pseudo code for our cooperative strategy-based differential evolution (CS-DE) algorithm.

Algorithm 1: Pseudo Code of CS − DE algorithm

Input: *Dimension number* D *, bound constraints* $[R^D_{min}, R^D_{max}]$ *, maximum number of function evaluations* nfe_{max} *, objective* $f(X)$ *.*

Output: *Global best value* $f(X_{\text{gbest}})$ *, global best individual* (X_{best}) *, numberof function evaluations* nf *;*

Initialize the population $P = \{X_1, X_2, \ldots, X_{PS}\}, K = 6, A = \emptyset, B = \emptyset, F = 0.5, CR = 0.9, \tau_1 = \tau_2 = 0.9 \mu_F = 0.6, \mu_{CR_i} = 0.8, p = 0.25 \sim 0.05, r^{rac,A} = 1.6, r^{rac,B} = 5, P(j) = \frac{1}{K}, PS = 25ln\ D\sqrt{D} \sim K, T_0 = \frac{G_{max}}{2}, N = 2 \cdot D, \xi = 0.001, R_{DP_{ini}}, ct = \emptyset, pivot\ (x, y), G = 1$

for $i = 1; i \leqslant PS; i++$ ***do***

  $X_{i,G} = X_i$*; Calculate the fitness value* $f(X_{i,G})$

end for

$nfe = PS;$

Find the global best $X_{gbest,G}$ *and fitness value* $f(X_{gbest,G})$*;*

while $nfe < nfe_{max}$ ***do***

  Generate the top p elites of the set $\boldsymbol{P}$*;*

  Divide the population into K groups according to Stochastic Universal Selection [23];

  Generate F and CR of all individuals according to Eq. (21) and Eq. (22), and generate F_b *according to Eq. (20);*

  for $i = 1; i \leqslant PS; i++$ ***do***

    Generating trial vectors $U_{i,G}$ *according to Eq. (19);*

    Calculate fitness value $f(U_{i,G})$*;*

  end for

  $nfe = nfe + PS;$

  for $i = 1; i \leqslant PS; i++$ ***do***

    if $f(U_{i,G}) < f(X_{i,G})$ ***then;***

      $X_{i,G+1} = U_{i,G};$

    else

      $X_{i,G+1} = X_{i,G};$

    end if

  end for

  if $S_F \neq \emptyset$ ***then***

    Update μ_F *according to Eq. (22), update* $P(j)$ *according to Eq. (21), update* $\mu_{CR_{idx}}$ *according to Eq.(23);*

  end if

    Adjust archives A and B;

    Find X_{gbest} *and* $f(X_{gbest})$*;*

    Calculate the counter ct for all of individuals and R_{DP} *according to Eq. (26);*

    for $i = 1; i \leqslant PS; i++$ ***do***

      if $ct(i) > N \&\& R_{DP} < \xi$ ***then***

        Adjust the position of the i^{th} *individual according to Eq. (27) and calculate the fitness value* $f(X_{i,G})$*;*

        $nfe = nfe + 1;$

    end if

  end for

    Adjust population size according to Eq. (24);

      $G = G + 1;$

  end while

$f(X_{gbest}) = f(X_{gbest,G}), X_{gbest} = X_{gbest,G};$

return $f(X_{gbest}), X_{gbest}$ *and nfe;*

Experimental results for PEMFC parameter optimization problem

Experimental setup

Due to its efficiency in maintaining population diversity and adaptability through dual mutation strategies, a variant of the differential evolution (DE) algorithm, CS-DE, is used. The adaptability of CS-DE preserves diversity during the evolution process and leads to stable convergence. The control parameters are specifically tuned to balance CS-DE, which reduces complexity. Furthermore, CS-DE is compared with nine other DE variants under consistent parameter settings (as described in Table 1) to demonstrate its effectiveness over a variety of scenarios. the superiority of CS-DE algorithm is applied and validated for PEMFC parameters optimization. For the validity purpose results obtained with CS-DE algorithm are compared with different nine variants of DE algorithms taken into consideration, and they are the 9 DE variants including PaDE [30], Di-DE [31], LSHADE [32], NDE [33], PalmDE [34], PSO-DE [35], jSO [36], LPalmDE [34], and HARD-DE [37], and the parameter settings of these contrasted algorithms are listed in Table 1.

Table 1 Default parameter settings of the DE algorithms in the comparison

Algorithms	Default settings
PaDE [30]	$\mu_F = 0.8, \mu_{Cr} = 0.6, F\&CR$ same as LSHADE,$k = 4, p = 0.11, PS = 25\log(D)\sqrt{D} \sim 4, r^{arc} = 1.6, T_0 = 70, r^d = 0.04$
Di-DE [31]	$PS = 10 \cdot D, F \sim C(\mu_F, 0.1), \mu_F = 0.5, CR \sim N(\mu_{CR}, 0.1), \mu_{CR} = 0.5, p = 0.1, r^{arc} = 5, K = 4$
LSHADE [32]	$\mu_F = 0.5, F \sim C(\mu_F, 0.1), \mu_{CR} = 0.5, CR \sim C(\mu_{CR}, 0.1), PS = 18 \cdot D \sim 4, r^{rac} = 2.6, p = 0.11, H = 6$
NDE [33]	$PS_{\text{ini}} = 10 \cdot D, PS_{\min} = 5, gm = 10, F^0_{loc} = CR^0_m = 0.5, c = 0.1$
PalmDE [34]	$F_j = 0.5, F_{ji} \sim C(F_j, 0.2), \mu_{Cr} = 0.5, Cr \sim N(\mu_{Cr}, 0.1), k = 8, p = 0.1, a = 1.6, T_0 = 70$
PSO-DE [35]	$NP = 100$,F $\in [0.9, 1.0]$),$CR \in [0.9,1.0]$
jSO [36]	$F, CR\&r^{rac}$ same as iLSHADE,$\mu_F = 0.3, \mu_{CR} = 0.8, PS = 25 \cdot \ln D \cdot \sqrt{D} \sim 4, p = 0.25 \sim 0.125, H = 5$
LPalmDE [34]	$F_j = 0.8, F_{j,i} \sim C(F_j, 0.1), \mu_{CR} = 0.6, CR \sim N(\mu_{CR}, 0.1), K = 20, PS = 25 \cdot \ln D \cdot \sqrt{D} \sim 4, p = 0.11, r^{rac} = 1.6, T_0 = \frac{C_{\max}}{2}$
HARD-DE [37]	$\mu_F = 0.3, \mu_{CR} = 0.8, F\&CR$ same as LSHADE,$p = 0.11, PS = 25 \cdot \ln D \cdot \sqrt{D} \sim 4, r^{rac,A} = 1.6, r^{rac,B} = 3, K = 4$
CS-DE [36]	$\mu_F = 0.6, \mu_{CR} = 0.8, F\&CR$ same as LSHADE, $p = 0.25 \sim 0.05, K = 6, r^{rac,A} = 1.6, r^{rac,B} = 5, NP = 25 \cdot \ln(D) \cdot \sqrt{D} \sim K, T_0 = \frac{C_{\max}}{2}, N = 2 \cdot D, \xi = 0.001$

Table 2 Characteristics of twelves PEMFCs known parameters used in this work

S. no	PEMFC type	Power(W)	Ncells	$A(\text{cm}^2)$	l(cm)	T(K)	$J_{\max}(\text{mA/cm}^2)$	P_{H_2}(bar)	P_{O_2}(bar)	Exp. data source
Case 1	BCS 500 W	500	32	64	178	333	469	1.0	0.2095	[26, 46]
Case 2	NetStack PS6	6000	65	240	178	343	1125	1.0	1.0	[27, 46]
Case 3	SR-12	500	48	62.5	25	323	672	1.47628	0.2095	[27, 46]
Case 4	H-12–1	12	13	8.1	25	323	246.9	0.4935	1.0	[28, 46]
Case 5	Ballard Mark V	5000	35	232	178	343	1500	1.0	1.0	[46]
Case 6	STD −1	250	24	27	127	343	860	1.0	1.0	[29, 46]
Case 7	Horizon	500	36	52	25	338	446	0.55	1.0	[28, 46]
Case 8	STD −2	250	24	27	127	343	860	1.5	1.5	[46]
Case 9	STD −3	250	24	27	127	343	860	2.5	3.0	[46]
Case 10	STD −4	250	24	27	127	353	860	2.5	3.0	[46]
Case 11	H-12–2	12	13	8.1	25	302	246.9	0.4	1.0	[46]
Case 12	H-12–3	13	13	8.1	25	312	246.9	0.5	1.0	[46]

Table 3 Boundaries of the unknown parameters

Parameters	ξ_1	ξ_2 ($\times 10^{-3}$)	ξ_3 ($\times 10^{-5}$)	ξ_4 ($\times 10^{-4}$)	λ	Rc (Ω)	b/V ($\times 10^{-2}$)
LB	−1.1997	1.0	3.6	−2.600	14	1.0	1.36
UB	−0.8532	5.0	9.8	−0.954	23	8.0	50.00

As discussed above all algorithms have applied for estimating the parameters of six different modules of fuel cells, namely BCS500W [26], NedStackPS6 [27], S12 [27], H12 [28], HORIZON [28], and Standard250W [29], under various operating conditions. The data sheet parameters of these commercial PEMFC stacks are given in Table 2. Table 2 lists characteristics of each fuel cell model including power, cell count, area, temperature, and pressure conditions. Multiple experimental sources are combined to acquire data. The experimental runs are conducted on a consistent computational platform: The system was a Windows Server 2019 environment with an Intel i7-11700k CPU at 3.6 GHz running MATLAB 2021a. A population size of 40 and a maximum of 500 iterations are used for each algorithm run, and 30 independent runs are used to ensure statistical validity. The characteristics and sources of experimental data for these are provided in Table 2. Experimental data for PEMFCs is typically obtained in two ways as described in the literature: (1) experimental data recording using hardware devices and software during experiments and (2) extracting data from figures or plots in references by digitizing. For example, [46] presents the experimental data of a commonly used 250 W PEMFC in graphs instead of tables. In their work [46], provide tabular experimental data for the same PEMFC example by digitizing. Tabular experimental data for the Temasek 1KW PEMFC in this study is also obtained from graphs.

In this study the authors did not directly collect the data for PEMFC parameter estimation. Datasets from referenced literature sources were used instead. Table 2 specifically lists the experimental data sources for different fuel cell models, such as power rating, cell count, active area and operational parameters. Detailed fuel cell performance data (voltage and current measurements) across different operating conditions were documented in the referenced studies [26–28], and [46] and have been used in our analysis for model validation. Since we did not use sensors directly, the datasets used in this study were obtained from literature. The methods and measurement systems used are described in the referenced studies [26]–[46], which also describe the types of sensors and their accuracies. The referenced studies provided processed, benchmark datasets that combined multiple measurements at different operational points. We did not directly record these measurements, but the original authors' methodologies yielded sample rates sufficient for PEMFC performance analysis. These measurements are made at a frequency and accuracy that are sufficient for reliable parameter optimization in this study. Control and recording of temperature and other environmental variables were carried out according to the conditions mentioned in the referenced works. Performance parameters, such as temperature, were kept within specified bounds during all

Table 4 Optimized parameters and optimal function value for CASE1

Algorithm	PaDE	Di-DE	LSHADE	NDE	PalmDE	PSO-DE	jSO	LPalmDE	HARD-DE	CS-DE
ξ_1	−0.9895056	−1.111674	−1.1777633	−1.0227485	−1.1510558	−0.859182	−1.082704	−1.0912144	−1.0262809	−1.0046502
ξ_2	0.0026673	0.0031037	0.0041181	0.0030461	0.0032265	0.0025086	0.0029351	0.0036102	0.0036592	0.0034995
ξ_3	4.101E−05	4.55E−05	0.000098	5.892E−05	4.578E−05	5.598E−05	4.022E−05	8.203E−05	9.785E−05	9.164E−05
ξ_4	−0.0001933	−0.000191	−0.000193	− −0.000194	−0.0001929	−0.0001927	−0.0001931	−0.0001931	−0.0001857	−0.000193
λ	20.878428	22.999954	20.877243	21.439265	20.821295	21.25439	21.073235	20.908776	21.324609	20.877244
R_c	0.0001421	0.000411	0.0001	0.0001022	0.0001038	0.0001684	0.0001214	0.0001002	0.0005332	0.0001
B	0.0154419	0.0156543	0.0161261	0.0163	0.0160731	0.0157996	0.0160807	0.0161424	0.0149407	0.0161261
Min	0.0283363	0.0268846	0.0254927	0.0258539	0.0255038	0.0262056	0.0256168	0.0254941	0.0432215	0.0254927
Max	0.2297341	0.0389233	0.0410172	0.0380104	0.029368	0.0399345	0.0265883	0.025619	0.3144221	0.0255517
Mean	0.1287692	0.0339229	0.032752	0.0289628	0.0262672	0.0320765	0.0258611	0.0255327	0.1413252	0.0254986
Std	0.0627659	0.0039831	0.0072419	0.0036685	0.0011713	0.0042286	0.0002784	4.506E−05	0.0719412	1.864E−05
RT	11.685302	8.3579428	5.7300561	6.2728277	12.187649	7.942668	11.832532	10.54944	14.681784	0.9768777
FR	9.1	6.8	5.9	5.6	3.9	6.9	3.9	2.2	9.7	1

measurements, and each dataset is based on measurements made under stabilized operating conditions.

Achieve accurate parameter estimation, these variables are bounded by lower and upper limits (LB and UB) determined from experimental data and operational ranges shown in Table 3. In the optimization model, the sum of squared error (SSE) between the model-predicted and experimental voltages is selected as the objective function. Boundary constraints apply to each decision variable, and performance constraints ensure the solution consistency with physical interpretations and operational viability.

CASE1: BCS 500 W PEMFC

In this case 1, the results in Table 4 obtained by the PaDE, Di-DE, LSHADE, NDE, PalmDE, PSO-DE, jSO, LPalmDE, HARD-DE, and CS-DE algorithms for the minimum value were 0.0283363, 0.0268846, 0.0254927, 0.0258539, 0.0255038, 0.0262056, 0.0256168, 0.0254941, 0.0432215, and 0.0254927, respectively. Table 4 clearly shows that the result with the CS-DE algorithm was 10.03%, 5.18%, 0.00%, 1.40%, 0.04%, 2.71%, 0.49%, 0.01%, and 41.05% lower than the results with the PaDE, Di-DE, LSHADE, NDE, PalmDE, PSO-DE, jSO, LPalmDE, and HARD-DE algorithms, respectively. For the maximum value, the results of PaDE, Di-DE, LSHADE, NDE, PalmDE, PSO-DE, jSO, LPalmDE, HARD-DE, and CS-DE were 0.2297341, 0.0389233, 0.0410172, 0.0380104, 0.029368, 0.0399345, 0.0265883, 0.025619, 0.3144221, and 0.0255517, respectively. CS-DE outperformed the other algorithms, with an 88.88%, 34.39%, 37.69%, 32.80%, 12.98%, 36.02%, 3.89%, 0.26%, and 91.88% improvement over PaDE, Di-DE, LSHADE, NDE, PalmDE, PSO-DE, jSO, LPalmDE, and HARD-DE, respectively. The mean values obtained by the PaDE, Di-DE, LSHADE, NDE, PalmDE, PSO-DE, jSO, LPalmDE, HARD-DE, and CS-DE algorithms were 0.1287692, 0.0339229, 0.032752, 0.0289628, 0.0262672, 0.0320765, 0.0258611, 0.0255327, 0.1413252, and 0.0254986, respectively. CS-DE achieved a 80.2%, 24.87%, 22.14%, 11.96%, 2.92%, 20.52%, 1.40%, 0.13%, and 81.96% lower mean value compared to PaDE, Di-DE, LSHADE, NDE, PalmDE, PSO-DE, jSO, LPalmDE, and HARD-DE, respectively. For standard deviation, the results were 0.0627659, 0.0039831, 0.0072419, 0.0036685, 0.0011713, 0.0042286, 0.0002784, 4.506E-05, 0.0719412, and 1.864E − 05, respectively. CS-DE outperformed the other algorithms by 99.97%, 99.53%, 99.74%, 99.49%, 98.41%, 99.56%, 93.31%, 58.63%, and 99.97%, respectively. For runtime (RT), CS-DE achieved the best value of 0.9768777 compared to 11.685302, 8.3579428, 5.7300561, 6.2728277, 12.187649, 7.942668, 11.832532, 10.54944, and 14.681784 for PaDE, Di-DE, LSHADE, NDE, PalmDE, PSO-DE, jSO, LPalmDE, and HARD-DE, respectively. CS-DE was faster by 91.64%, 88.31%, 82.95%, 84.43%, 91.99%, 87.70%, 91.74%, 90.74%, and 93.35%, respectively. Finally, for the Friedman rank (FR), CS-DE achieved a rank of 1, which was 88.24%, 85.29%, 83.05%, 82.14%, 74.36%, 85.51%, 74.36%, 54.55%, and 89.69% better than PaDE, Di-DE, LSHADE, NDE, PalmDE, PSO-DE, jSO, LPalmDE, and HARD-DE, respectively.

Table 5 Performance metrics of CS-DE algorithm for CASE1

AE_v (A)	*RE* %	*MBE*
2.7820×10^{-3}	9.5931×10^{-3}	4.300×10^{-7}
4.0637×10^{-3}	1.54456×10^{-2}	9.174×10^{-7}
3.5563×10^{-3}	1.41743×10^{-2}	7.026×10^{-7}
4.6227×10^{-3}	1.90625×10^{-2}	1.187×10^{-6}
5.4196×10^{-3}	2.31906×10^{-2}	1.632×10^{-6}
1.4619×10^{-2}	6.47745×10^{-2}	1.187×10^{-5}
1.1332×10^{-2}	5.13719×10^{-2}	7.135×10^{-6}
8.4691×10^{-3}	3.89383×10^{-2}	3.985×10^{-6}
1.1268×10^{-2}	5.25338×10^{-2}	7.054×10^{-6}
1.02252×10^{-1}	4.84836×10^{-1}	5.809×10^{-4}
1.4516×10^{-2}	7.01940×10^{-2}	1.171×10^{-5}
1.0993×10^{-2}	5.43676×10^{-2}	6.714×10^{-6}
1.0950×10^{-2}	5.5419×10^{-2}	6.662×10^{-6}
6.0325×10^{-3}	3.11598×10^{-2}	2.022×10^{-6}
6.4744×10^{-3}	3.43289×10^{-2}	2.329×10^{-6}
4.7290×10^{-3}	2.58838×10^{-2}	1.242×10^{-6}
3.3193×10^{-3}	1.84918×10^{-2}	6.121×10^{-7}
7.1145×10^{-3}	4.11244×10^{-2}	2.812×10^{-6}
1.29176×10^{-2}	6.13828×10^{-2}	3.610×10^{-5}

In the analysis of the BCS 500 W PEMFC parameter optimization, the CS-DE algorithm demonstrates superior performance compared to several other differential evolution variants, such as PaDE, L-SHADE, and Di-DE. A primary factor in this comparison is computational efficiency, where CS-DE stands out with a runtime of 0.9768777 s, which is significantly faster than PaDE (11.685302 s) and L-SHADE (5.7300561 s) as shown in Table 5. This substantial reduction in computational time not only enhances the algorithm suitability for scenarios requiring rapid solutions but also indicates its capability for resource-efficient optimization. Such efficiency makes CS-DE an optimal choice in industrial applications where real-time solutions are crucial.

When evaluating stability, CS-DE also had a better outcome than the other algorithms. The standard deviation (1.86E − 05) of its results is the lowest amongst all the evaluated algorithms, thus pointing to low variance across runs. This indicates that CS-DE is inherently accurate and uniform in its outcomes, which is important given the sensitivity of PEMFC parameters. Other algorithms including Di-DE and

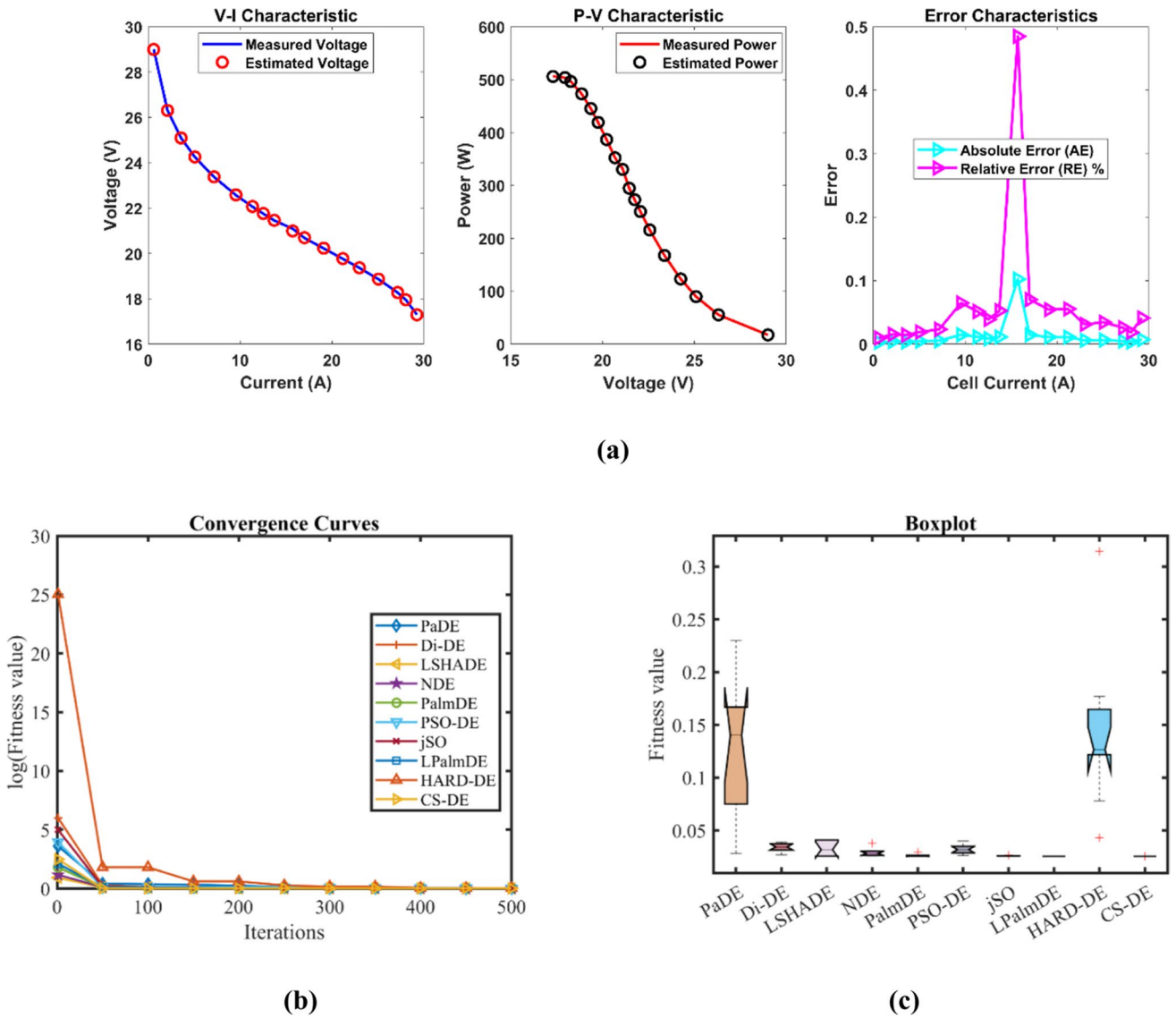

Fig. 2 CS-DE algorithm characteristic curves of CASE1. **a** V-I, P–V, and error curve. **b** Convergence curve. **c** Box-plot

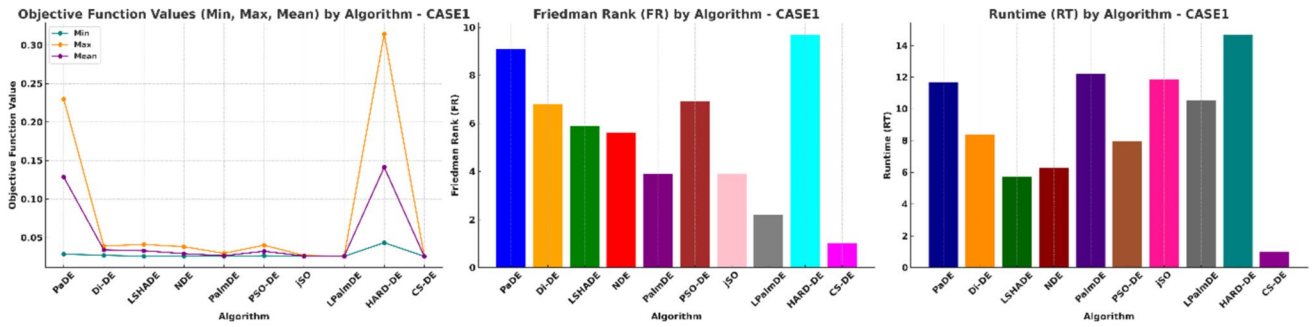

Fig. 3 Optimized parameters, Friedman rank, and runtime comparison for CASE1 algorithms

Table 6 Optimized parameters and optimal function value for CASE2

Algorithm	PaDE	Di-DE	LSHADE	NDE	PalmDE	PSO-DE	jSO	LPalmDE	HARD-DE	CS-DE
ξ_1	−0.9880406	−0.8535092	−0.8532	−1.1031158	−0.8676942	−0.9239523	−0.9262413	−0.9634714	−1.19969	−0.8536178
ξ_2	0.003659	0.0024014	0.0032672	0.0031334	0.0025978	0.0031227	0.0033418	0.0028508	0.0034877	0.0026234
ξ_3	9.788E−05	3.606E−05	0.000098	3.649E−05	4.722E−05	7.303E−05	8.808E−05	4.537E−05	4.109E−05	5.198E−05
ξ_4	−0.0000954	−0.0000954	−0.0000954	−0.0000954	−9.541E−05	−0.0000954	−0.0000954	−9.541E−05	−9.82E−05	−0.0000954
λ	14	14.036867	14	14	14.00267	14	14	14.030207	14.228782	14
R_c	0.0001255	0.0001417	0.0001204	0.0001151	0.0001202	0.0001122	0.0001172	0.0001285	0.0001	0.0001204
B	0.0162035	0.0142912	0.0167879	0.0175718	0.0168233	0.0179565	0.0176998	0.0158235	0.0224629	0.0167879
Min	0.2753283	0.276104	0.2752105	0.2752659	0.2752676	0.2754132	0.275635	0.2758192	0.3092434	0.2752105
Max	1.1218268	0.3408987	0.3360206	0.3095715	0.29778	0.3138852	0.2766637	0.2971678	1.2789887	0.2758716
Mean	0.4392358	0.3079869	0.2863009	0.2846798	0.2853486	0.287524	0.2760853	0.280516	0.5063065	0.2752766
Std	0.2719574	0.0191607	0.0224584	0.0141157	0.0089032	0.0137947	0.0002776	0.0068799	0.2944166	0.0002091
RT	8.9136077	8.4741483	7.5203771	8.0773615	16.413646	8.4283155	8.6092201	10.21943	0.2289158	16.694281
FR	7.9	7.6	4	4.2	5.3	5.9	3.9	5.2	9.6	1.4

L-SHADE are observed to be less stable, suggesting fluctuating performance in multiple runs.

The CS-DE has the highest mean SSE of 0.0254986 which is the best among the compared algorithms which shows for instance, Di-DE gets a mean SSE of 0.0339229 while L-SHADE gets 0.032752 (Table 4). Such level of error control is important in the context of PEMFCs performance and its operational parameters, to ensure that systems work to the best of their capacity and with the expected reliability.

The Friedman rank (FR) also supports the This ranking further supports CS-DE performance given that it is the best algorithm for this optimization task across all the study metrics.

Furthermore, the experimental and estimated data in Table 5 clearly illustrate the ability of CS-DE in predicting PEMFC parameters. The absolute errors between the experimental and estimated values for current are also relatively low and range between 0.00257 and 0.02087 A with an average of 0.0129176 A and relative error ranging from 0.01529 to 0.12913% with an average of 0.0613828%. These metrics further illustrate the CS-DE algorithm capability to closely match real-world experimental data, reinforcing its robustness in practical applications.

V-I, P–V, and error curves, convergence behavior, and a statistical box plot are shown in Fig. 2 to illustrate CS-DE accuracy in parameter estimation and its ability to consistently minimize error. The optimized parameter values (min, max, and mean), Friedman rank, and runtime of each algorithm are shown in Fig. 3. CS-DE is robust across scenarios and demonstrates the lowest FR rank, runtime and lowest variation, and highest consistency.

CASE2: NetStack PS6 PEMFC

In this case 2, the results in Table 6 obtained by the PaDE, Di-DE, LSHADE, NDE, PalmDE, PSO-DE, jSO, LPalmDE, HARD-DE, and CS-DE algorithms for the minimum value were 0.2753283, 0.276104, 0.2752105, 0.2752659, 0.2752676, 0.2754132, 0.275635, 0.2758192, 0.3092434, and 0.2752105, respectively. Table 6 shows that the result with the CS-DE algorithm was 0.04%, 0.32%, 0.00%, 0.02%, 0.02%, 0.07%, 0.15%, 0.22%, and 11.00% lower than the results with the PaDE, Di-DE, LSHADE, NDE, PalmDE, PSO-DE, jSO, LPalmDE, and HARD-DE algorithms, respectively. For the maximum value, the results of PaDE, Di-DE, LSHADE, NDE, PalmDE, PSO-DE, jSO, LPalmDE, HARD-DE, and CS-DE were 1.1218268, 0.3408987, 0.3360206, 0.3095715, 0.29778, 0.3138852, 0.2766637, 0.2971678, 1.2789887, and 0.2758716, respectively. CS-DE outperformed the other algorithms, showing a 75.41%, 19.06%, 17.87%, 10.89%, 7.35%, 12.10%, 0.29%, 7.17%, and 78.43% improvement over PaDE, Di-DE, LSHADE, NDE, PalmDE, PSO-DE, jSO, LPalmDE, and HARD-DE,

Table 7 Performance metrics of CS-DE algorithm for CASE2

AE_v (A)	*RE %*	*MBE*
0.6871025	1.1147023	0.0162797
0.1839248	0.3087541	0.0011665
0.0830142	0.1408452	0.0002376
0.0675332	0.1173674	0.0001573
0.1049744	0.184814	0.0003800
0.1069430	0.1905273	0.0003944
0.0919471	0.1664803	0.0002915
0.0569873	0.1042578	0.000112
0.0088833	0.0165702	2.721E − 06
0.0726635	0.137464	0.0001821
0.4743935	0.9138769	0.0077603
0.1945859	0.3799023	0.0013056
0.2332623	0.4697186	0.0018763
0.3589724	0.7325968	0.0044435
0.1008158	0.2093787	0.0003505
0.1374175	0.2891783	0.0006512
0.0271492	0.0576415	2.542E − 05
0.1969212	0.4236687	0.0013372
0.1746748	0.3825553	0.0010521
0.0255307	0.0569246	2.248E − 05
0.1831352	0.4139584	0.0011565
0.5657137	1.3326589	0.0110356
0.4975319	1.1942676	0.0085358
0.3675286	0.9034626	0.0046578
0.2795602	0.6973315	0.002695
0.1541499	0.3901542	0.0008194
0.0301449	0.0778334	3.133E − 05
0.1942051	0.5090567	0.0013005
0.4657674	1.2460338	0.0074807
0.2112218	**0.4538614**	**0.0026118**

respectively. The mean values obtained by the PaDE, Di-DE, LSHADE, NDE, PalmDE, PSO-DE, jSO, LPalmDE, HARD-DE, and CS-DE algorithms were 0.4392358, 0.3079869, 0.2863009, 0.2846798, 0.2853486, 0.287524, 0.2760853, 0.280516, 0.5063065, and 0.2752766, respectively. CS-DE achieved a 37.32%, 10.60%, 3.86%, 3.29%, 3.52%, 4.27%, 0.29%, 1.87%, and 45.65% lower mean value compared to PaDE, Di-DE, LSHADE, NDE, PalmDE, PSO-DE, jSO, LPalmDE, and HARD-DE, respectively. For standard deviation, the results were 0.2719574, 0.0191607, 0.0224584, 0.0141157, 0.0089032, 0.0137947, 0.0002776, 0.0068799, 0.2944166, and 0.0002091, respectively. CS-DE outperformed the other algorithms by 99.92%, 98.91%, 99.07%, 98.52%, 97.65%, 98.48%, 24.67%, 96.96%, and 99.93%, respectively. For runtime (RT), CS-DE achieved the best value of 0.2289158 compared to 8.9136077, 8.4741483, 7.5203771, 8.0773615, 16.413646, 8.4283155, 8.6092201, 10.21943, and 16.694281 for PaDE, Di-DE, LSHADE, NDE, PalmDE, PSO-DE, jSO, LPalmDE, and HARD-DE, respectively. CS-DE was faster by 97.43%, 97.30%, 96.96%, 97.16%, 98.61%, 97.28%, 97.34%, 97.76%, and 98.63%, respectively. Finally, for the Friedman rank (FR), CS-DE achieved a rank of 1.4, which was 82.28%, 81.58%, 65.00%, 66.67%, 73.58%, 76.27%, 64.10%, 73.08%, and 85.42% better than PaDE, Di-DE, LSHADE, NDE, PalmDE, PSO-DE, jSO, LPalmDE, and HARD-DE, respectively.

In the optimization of parameters for the NetStack PS6 PEMFC, the CS-DE algorithm demonstrates strong consistency and stability in comparison to other differential evolution variants. Notably, CS-DE maintains tight clusters around its minimum and maximum performance scores, with a minimum value of 0.2752105 and a maximum of 0.2758716. This narrow range, paired with a remarkably low standard deviation of 0.0002091, indicates superior stability and reliability (Table 6). In contrast, algorithms like PaDE and HARD-DE display much higher variability, with standard deviations of 0.2719574 and 0.2944166, respectively, suggesting that CS-DE consistently outperforms its counterparts under varying conditions.

CS-DE achieves a mean value of 0.2752766, which is very close to its minimum score, further reinforcing its ability to maintain near-optimal performance across multiple runs. This level of precision is vital for industrial applications where reliability is crucial. Moreover, CS-DE outshines other algorithms in terms of Friedman Rank, achieving a rank of 1.4, while algorithms like PaDE and L-SHADE fall behind with ranks of 7.9 and 4, respectively.

In terms of computational efficiency, CS-DE runtime of 16.694281 s is notably longer than several other algorithms, such as HARD-DE, which completes in 0.2289158 s (Table 6). However, this extended runtime can be attributed to CS-DE thorough search process, which enables it to maintain a competitive edge in accuracy and overall performance. For applications requiring highly reliable and precise solutions, this trade-off between speed and accuracy is acceptable, as CS-DE offers superior stability and optimal results.

The precision of CS-DE in modeling PEMFC parameters is further demonstrated by the close alignment between experimental and estimated values across various current levels in Table 7. The algorithm consistently maintains low absolute error values, with an average of 0.2112218 A and an average relative error of 0.4538614%, underscoring its robustness in practical applications. The mean bias error (MBE) also remains minimal, with most errors being on the micro-scale, showcasing the algorithm ability to closely mimic real-world data.

V-I, P–V, and error curves, convergence behavior, and a statistical box plot are shown in Fig. 4 to illustrate CS-DE accuracy in parameter estimation and its ability to consistently minimize error. The optimized parameter values (min,

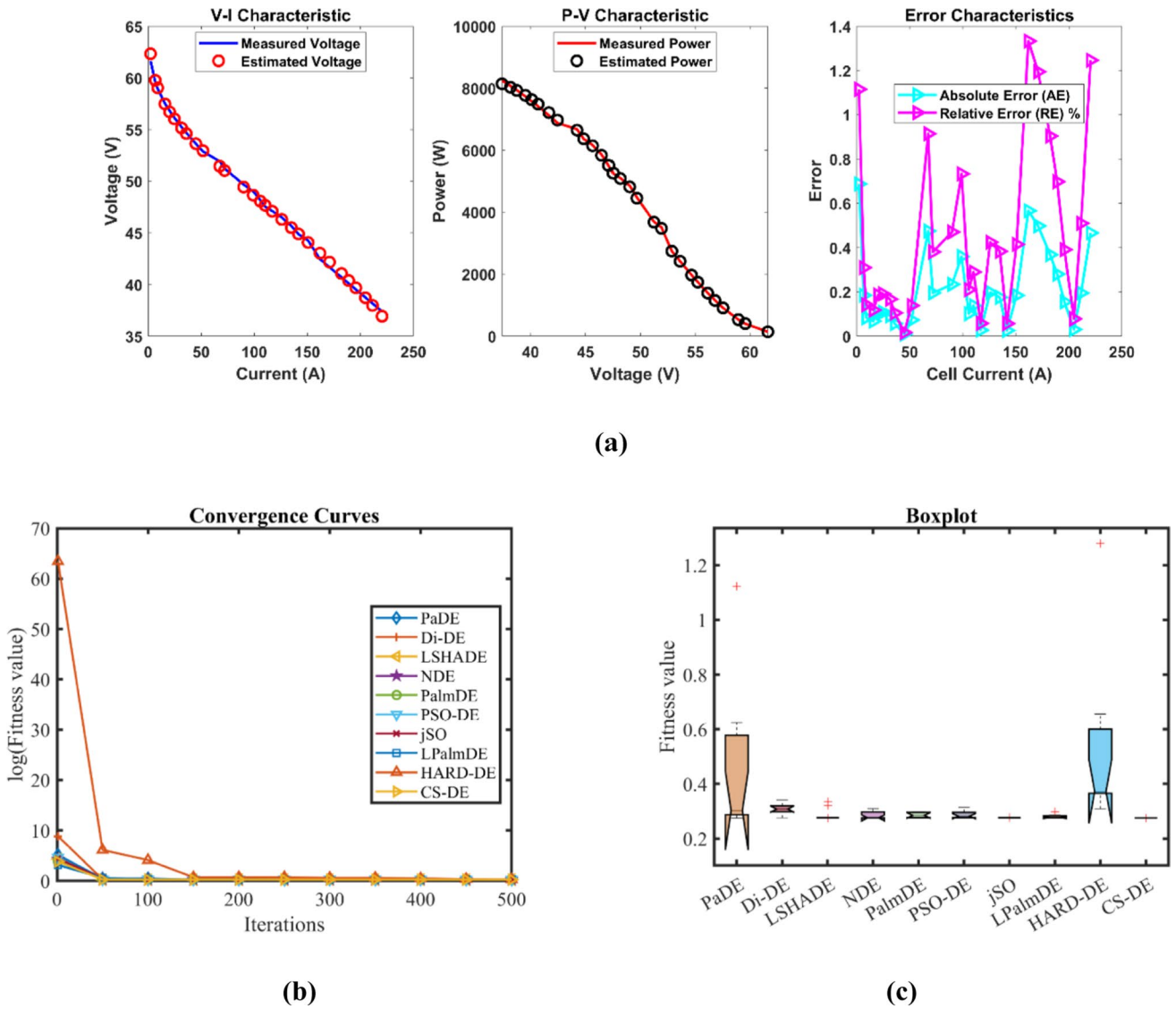

Fig. 4 CS-DE algorithm characteristic curves of CASE2. **a** V-I, P–V, and error curve. **b** Convergence curve. **c** Box-plot

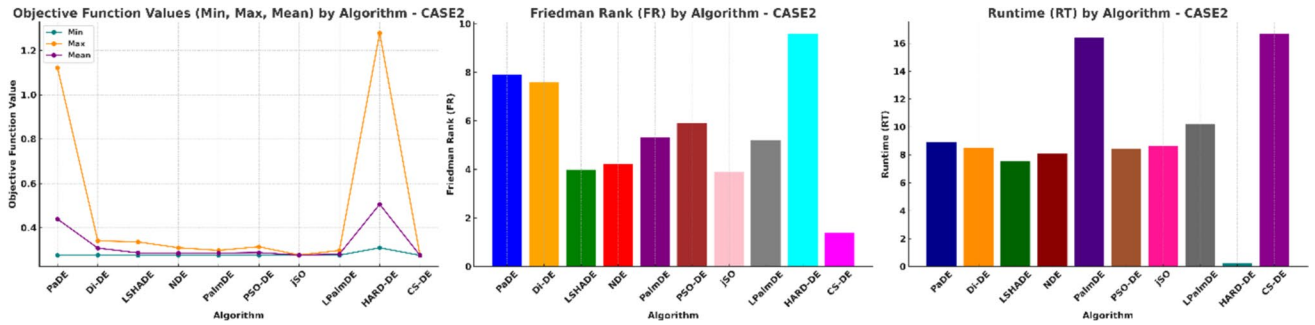

Fig. 5 Optimized parameters, Friedman rank, and runtime comparison for CASE2 algorithms

max, and mean), Friedman rank, and runtime of each algorithm are shown in Fig. 5.

CASE3: SR-12 PEMFC

In this case 3, the results in Table 8 obtained by the PaDE,

Table 8 Optimized parameters and optimal function value for CASE3

Algorithm	PaDE	Di-DE	LSHADE	NDE	PalmDE	PSO-DE	jSO	LPalmDE	HARD-DE	CS-DE
ξ_1	−0.877963	−1.1993872	−0.8532	−0.9167398	−1.0743672	−0.911685	−1.0915075	−1.0733753	−0.9034297	−0.8739118
ξ_2	0.0029623	0.0043113	0.0032487	0.0027298	0.0034273	0.0028358	0.0035948	0.003785	0.0034131	0.0026063
ξ_3	7.463E−05	9.735E−05	0.000098	5.182E−05	6.532E−05	5.966E−05	7.273E−05	8.862E−05	9.797E−05	5.239E−05
ξ_4	−0.0000954	−9.54E−05	−0.0000954	−0.0000954	−9.555E−05	−0.0000954	−9.542E−05	−9.542E−05	−0.0001031	−0.0000954
λ	18.456489	22.999645	23	20.224225	22.550611	16.370266	23	22.994442	16.940161	23
R_c	0.0008	0.0007354	0.0006726	0.0006225	0.0006562	0.0005567	0.0006872	0.0006676	0.0007844	0.0006726
B	0.1700356	0.1740772	0.1753203	0.1754366	0.1754417	0.1749361	0.1748638	0.1753994	0.1655309	0.1753203
Min	0.2458191	0.2423892	0.2422841	0.2423988	0.2423432	0.2427067	0.2423502	0.2422905	0.2642945	0.2422841
Max	0.5275828	0.2460415	0.2463869	0.2457577	0.2430344	0.2474108	0.2425383	0.2425416	0.5883347	0.2429272
Mean	0.3860373	0.2438574	0.2442682	0.2442493	0.2425717	0.2448369	0.2424429	0.2423323	0.4401371	0.2424127
Std	0.0898295	0.0011439	0.0017559	0.0013061	0.0002141	0.0019047	7.068E−05	7.587E-05	0.1064393	0.0002712
RT	6.6840992	6.1639239	5.1423761	5.4583498	11.730111	6.201185	6.1969051	7.7877861	12.201199	0.2907239
FR	9.4	5.9	5.9	6.7	4	6.9	3	2	9.5	1.7

Table 9 Performance metrics of CS-DE algorithm for CASE3

AE_v (A)	*RE* %	*MBE*
0.170794	0.3956312	0.0016206
0.0499376	0.1213846	0.0001385
0.1755033	0.4377732	0.0017112
0.1828632	0.4683997	0.0018577
0.0565507	0.1488569	0.0001777
0.0654787	0.1765877	0.0002382
0.0498901	0.1384681	0.0001383
0.0186516	0.0530025	1.933E−05
0.1720727	0.5050564	0.0016449
0.2631104	0.7968213	0.0038459
0.2306845	0.7199892	0.0029564
0.0376779	0.1207627	7.887E−05
0.3273558	1.0985093	0.0059534
0.0428819	0.1480727	0.0001022
0.6622591	2.3551176	0.0243659
0.3082114	1.1719064	0.0052775
0.075147	0.3123317	0.0003137
0.3856179	1.8019528	0.0082612
0.1819271	**0.6094791**	**0.0032612**

Di-DE, LSHADE, NDE, PalmDE, PSO-DE, jSO, LPalmDE, HARD-DE, and CS-DE algorithms for the minimum value were 0.2458191, 0.2423892, 0.2422841, 0.2423988, 0.2423432, 0.2427067, 0.2423502, 0.2422905, 0.2642945, and 0.2422841, respectively. Table 8 shows that the result with the CS-DE algorithm was 1.44%, 0.04%, 0.00%, 0.05%, 0.02%, 0.17%, 0.03%, 0.00%, and 8.32% lower than the results with the PaDE, Di-DE, LSHADE, NDE, PalmDE, PSO-DE, jSO, LPalmDE, and HARD-DE algorithms, respectively. For the maximum value, the results of PaDE, Di-DE, LSHADE, NDE, PalmDE, PSO-DE, jSO, LPalmDE, HARD-DE, and CS-DE were 0.5275828, 0.2460415, 0.2463869, 0.2457577, 0.2430344, 0.2474108, 0.2425383, 0.2425416, 0.5883347, and 0.2429272, respectively. CS-DE outperformed the other algorithms, showing a 53.96%, 1.27%, 1.41%, 1.15%, 0.04%, 1.82%, 0.16%, 0.16%, and 58.71% improvement over PaDE, Di-DE, LSHADE, NDE, PalmDE, PSO-DE, jSO, LPalmDE, and HARD-DE, respectively. The mean values obtained by the PaDE, Di-DE, LSHADE, NDE, PalmDE, PSO-DE, jSO, LPalmDE, HARD-DE, and CS-DE algorithms were 0.3860373, 0.2438574, 0.2442682, 0.2442493, 0.2425717, 0.2448369, 0.2424429, 0.2423323, 0.4401371, and 0.2424127, respectively. CS-DE achieved a 37.21%, 0.59%, 0.76%, 0.75%, 0.07%, 0.99%, 0.01%, 0.05%, and 44.92% lower mean value compared to PaDE, Di-DE, LSHADE, NDE, PalmDE, PSO-DE, jSO, LPalmDE, and HARD-DE, respectively. For standard deviation, the results were 0.0898295, 0.0011439, 0.0017559, 0.0013061, 0.0002141, 0.0019047, 7.068E-05, 7.587E−05,

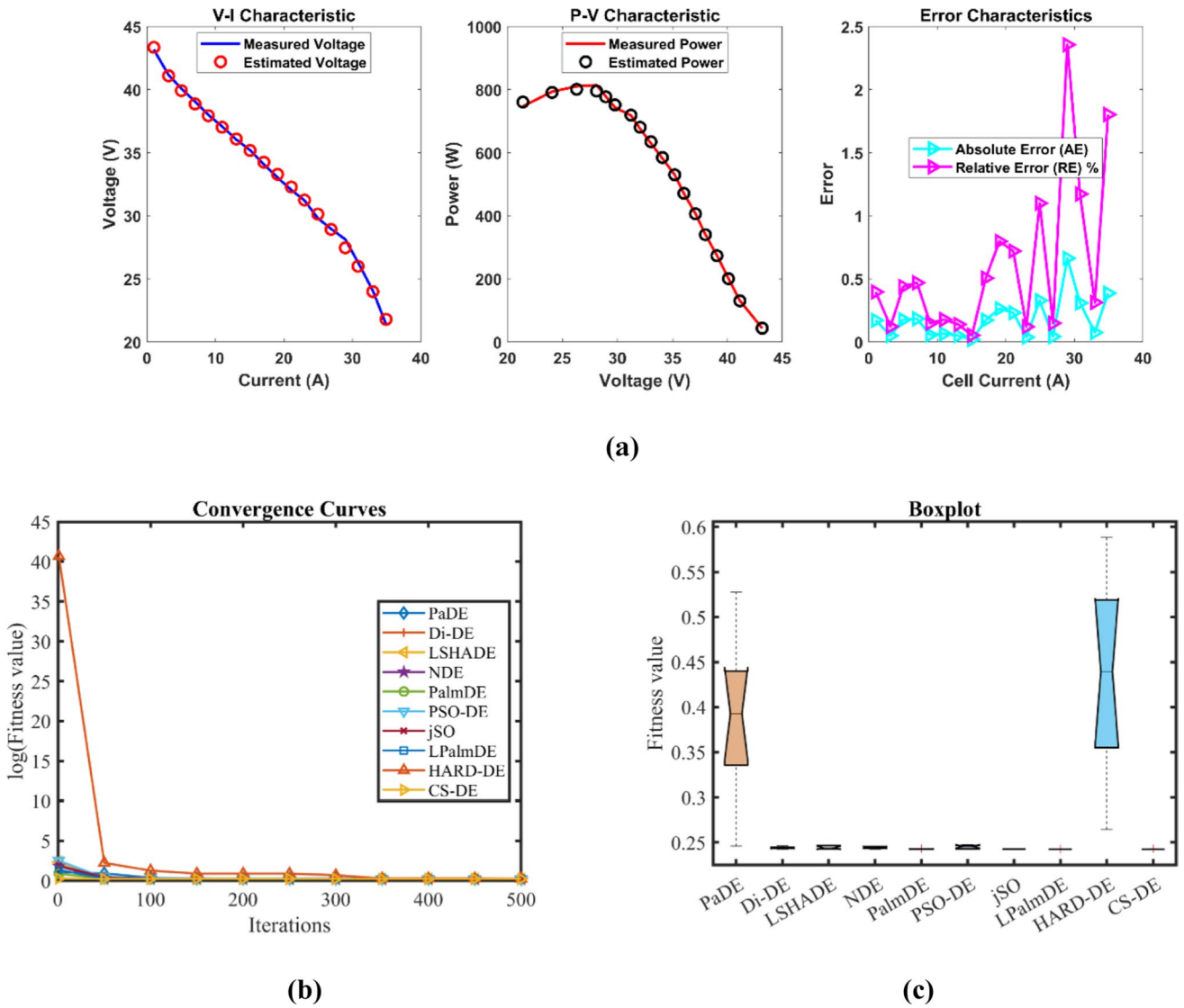

Fig. 6 CS-DE algorithm characteristic curves of CASE3. **a** V-I, P–V, and error curve. **b** Convergence curve. **c** Box-plot

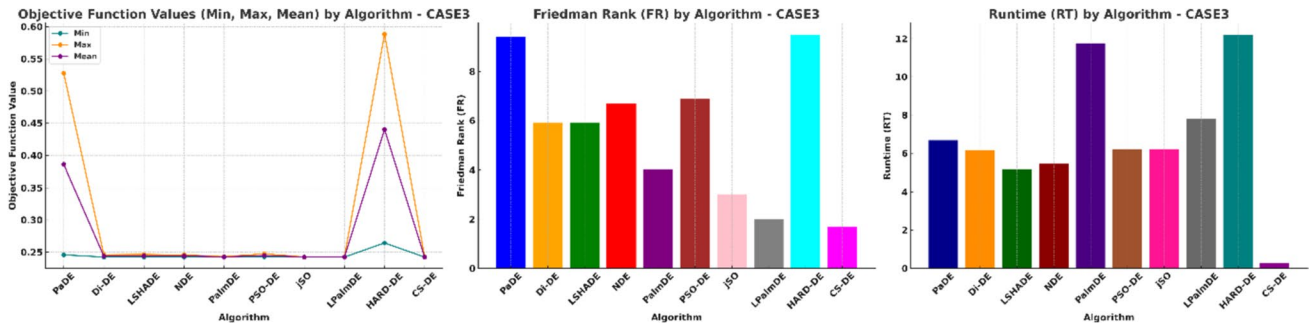

Fig. 7 Optimized parameters, Friedman rank, and runtime comparison for CASE3 algorithms

Table 10 Optimized parameters and optimal function value for CASE4

Algorithm	PaDE	Di-DE	LSHADE	NDE	PalmDE	PSO-DE	jSO	LPalmDE	HARD-DE	CS-DE
ξ_1	−1.1470519	−0.8621439	−1.0381245	−1.19969	−0.9225568	−1.0877413	−0.8764689	−0.9539458	−0.8919058	−0.8546202
ξ_2	0.0032344	0.0018742	0.0029446	0.0033577	0.001975	0.0029056	0.0016321	0.0021729	0.0020778	0.0015224
ξ_3	9.46E−05	6.027E−05	0.000098	9.177E−05	5.407E−05	8.423E−05	3.97E−05	6.131E−05	6.829E−05	3.668E−05
ξ_4	−0.0001113	−0.0001113	−0.0001113	−0.0001113	−0.0001113	−0.000112	−0.0001113	−0.0001114	−0.0001092	−0.0001113
λ	14	14	14	14.011205	14.000004	14.039784	14	14.000353	17.311423	14
R_c	0.0008	0.0008	0.0008	0.0007287	0.0008	0.0002491	0.0008	0.0007999	0.0004824	0.0008
B	0.0136	0.0136	0.0136	0.0136163	0.0136	0.013915	0.0136	0.0136029	0.0140863	0.0136
Min	0.102915	0.1029149	0.1029149	0.1029956	0.1029149	0.103553	0.1029149	0.1029157	0.1046149	0.1029149
Max	0.106724	0.1035953	0.1052106	0.1039134	0.1030454	0.1044816	0.1029193	0.1029199	0.1182573	0.1029149
Mean	0.1046458	0.1031064	0.1037494	0.1034397	0.1029377	0.1037865	0.1029155	0.102917	0.1088349	0.1029149
Std	0.0013557	0.0002949	0.0009101	0.0003464	4.537E−05	0.0002865	1.354E−06	1.454E−06	0.0046026	7.864E−17
RT	6.4895209	5.9163514	5.0755221	5.3711037	11.488185	6.1447201	6.0679015	7.1510288	12.066956	0.1114574
FR	7.8	4.2	5.1	7	4.3	7.4	3.5	4.6	9.8	1.3

Table 11 Performance metrics of CS-DE algorithm for CASE4

AE_v (A)	*RE* %	*MBE*
0.1755287	1.832241	0.0017117
0.0155319	0.1648821	1.34E−05
0.0346961	0.3750932	6.688E−05
0.1240069	1.3479007	0.0008543
0.1421092	1.5633574	0.0011219
0.1072871	1.1987385	0.0006395
0.0871402	0.9846356	0.0004219
0.061316	0.701556	0.0002089
0.0484139	0.5596982	0.0001302
0.0333924	0.3951759	6.195E−05
0.0388661	0.4621419	8.392E−05
0.1413828	1.7241809	0.0011105
0.1526616	1.8800685	0.0012948
0.1211975	1.4944204	0.000816
0.0875137	1.0871269	0.0004255
0.038855	0.4862951	8.387E−05
0.0373984	0.4704201	7.77E−05
0.1625878	2.0477055	0.0014686
0.0894381	**1.043091**	**0.0005884**

0.1064393, and 0.0002712, respectively. CS-DE outperformed the other algorithms by 99.70%, 76.29%, 84.56%, 79.23%, 26.65%, 85.76%, 62.02%, 64.26%, and 99.75%, respectively. For runtime (RT), CS-DE achieved the best value of 0.2907239 compared to 6.6840992, 6.1639239, 5.1423761, 5.4583498, 11.730111, 6.201185, 6.1969051, 7.7877861, and 12.201199 for PaDE, Di-DE, LSHADE, NDE, PalmDE, PSO-DE, jSO, LPalmDE, and HARD-DE, respectively. CS-DE was faster by 95.65%, 95.28%, 94.35%, 94.67%, 97.52%, 95.31%, 95.31%, 96.27%, and 97.62%, respectively. Finally, for the Friedman rank (FR), CS-DE achieved a rank of 1.7, which was 81.91%, 71.19%, 71.19%, 74.63%, 57.50%, 75.36%, 43.33%, 15.00%, and 82.10% better than PaDE, Di-DE, LSHADE, NDE, PalmDE, PSO-DE, jSO, LPalmDE, and HARD-DE, respectively.

While optimizing parameter for SR-12 W PEMFC, the CS-DE algorithm again demonstrates exceptional precision and stability, with a minimum value of 0.2422841 and a maximum of 0.2429272, showing low variability across runs. Its mean value of 0.2424127 is closely aligned with its minimum, supported by an exceptionally low standard deviation of 0.0002712 as shown in Table 8, significantly outperforming other algorithms like LSHADE and PaDE in terms of reliability. Despite a short runtime of only 0.2907239 s, CS-DE ranks second in the Friedman rank with a score of 1.7, emphasizing its competitive efficiency and accuracy. Competitors such as PalmDE and HARD-DE take significantly longer to complete, with runtimes of 11.730111 and 12.201199 s, respectively.

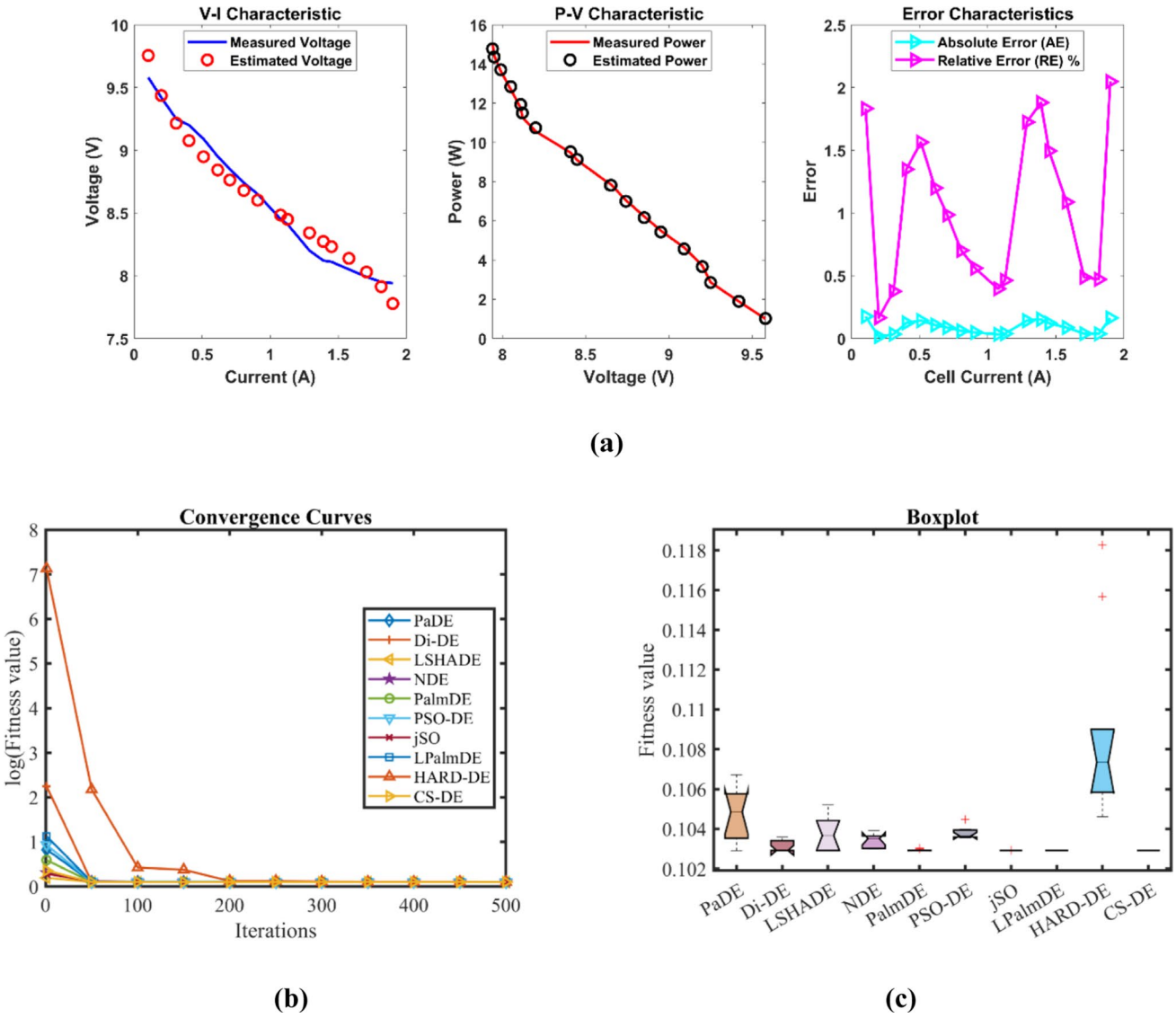

Fig. 8 CS-DE algorithm characteristic curves of CASE4. **a** V-I, P–V, and error curve. **b** Convergence curve. **c** Box-plot

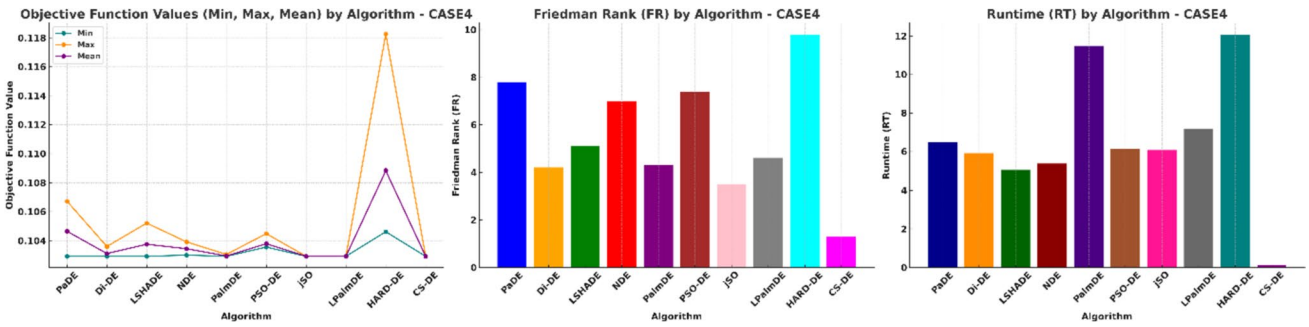

Fig. 9 Optimized parameters, Friedman rank, and runtime comparison for CASE4 algorithms

Table 12 Optimized parameters and optimal function value for CASE5

Algorithm	PaDE	Di-DE	LSHADE	NDE	PalmDE	PSO-DE	jSO	LPalmDE	HARD-DE	CS-DE
ξ_1	−0.8607955	−1.0590525	−0.9425496	−1.1098987	−0.9681708	−0.9101942	−1.0133276	−1.1251904	−1.0969003	−0.8662804
ξ_2	0.0028594	0.0028688	0.003118	0.0037828	0.0031243	0.0030529	0.0032721	0.0035109	0.0033885	0.0023327
ξ_3	7.785E−05	3.669E−05	7.872E−05	9.139E−05	7.388E−05	8.078E−05	7.499E−05	6.877E−05	6.582E−−05	3.855E-05
ξ_4	−−0.0001654	−0.0001729	−0.0001737	−0.0001732	−0.0001735	−0.000175	−0.0001739	−0.0001738	−0.0001764	−0.0001739
λ	14.553807	14.705877	14.355314	14.477204	14.466849	14.632798	14.484418	14.427913	16.724149	14.439129
R_c	0.0002507	0.0002723	0.0001	0.0001458	0.0001126	0.0001024	0.0001001	0.0001006	0.0003789	0.0001
B	0.0155777	0.013715	0.0136	0.0137225	0.013814	0.0139988	0.0139469	0.0137888	0.0170793	0.013795
Min	0.1559949	0.1489216	0.1486427	0.1487276	0.1486747	0.148713	0.1486438	0.1486334	0.1518974	0.1486318
Max	0.1733274	0.1598719	0.1499592	0.1501851	0.1496448	0.1531455	0.1489597	0.1487316	0.3083287	0.1486318
Mean	0.1629451	0.1515686	0.1492812	0.1493798	0.1490115	0.1499409	0.14874	0.1486508	0.195651	0.1486318
Std	0.0060897	0.003561	0.0005772	0.000548	0.0002943	0.0012345	9.219E−05	3.129E−05	0.0451842	2.831E−16
RT	5.8308216	5.4742715	4.583439	4.8757855	10.539296	5.6155459	5.5121041	6.592608	0.1076193	11.001723
FR	9.1	7.1	5.1	5.7	4.6	6.6	3.7	2.3	9.8	1

Table 13 Performance metrics of CS-DE algorithm for CASE5

AE_v (A)	*RE %*	*MBE*
0.0169093	0.0719544	1.906E−05
0.2486896	1.1566957	0.0041231
0.2598219	1.2674239	0.0045005
0.2095847	1.0531897	0.0029284
0.1024599	0.5254355	0.0006999
0.0927381	0.4880952	0.0005734
0.119649	0.6467511	0.0009544
0.0772374	0.4339182	0.0003977
0.2759021	1.5948098	0.0050748
0.0746532	0.460822	0.0003715
0.0982900	0.6181759	0.0006441
0.0936677	0.6043076	0.0005849
0.0511496	0.3387388	0.0001744
0.1218031	0.8342681	0.0009891
0.0290513	0.2105164	5.627E−05
0.1247738	**0.6870068**	**0.0014728**

In Table 9, CS-DE maintains low absolute error values and an average relative error of 0.6094791%, indicating its high precision in modeling PEMFC parameters. These results highlight the algorithm balance of speed and accuracy, making it an ideal choice for real-time optimization in industrial applications.

V-I, P–V, and error curves, convergence behavior, and a statistical box plot are shown in Fig. 6 to illustrate CS-DE accuracy in parameter estimation and its ability to consistently minimize error. The optimized parameter values (min, max, and mean), Friedman rank, and runtime of each algorithm are shown in Fig. 7.

CASE4: H-12–1 PEMFC

Table 10 results demonstrate the result obtain for optimization of parameters for the H-12–1 PEMFC. This result makes the CS-DE algorithm distinguishes itself through exceptional stability and precision, significantly outperforming other differential evolution strategies. With a minimal standard deviation of 7.86E−17 across runs, CS-DE exhibits unparalleled consistency compared to algorithms such as HARD-DE, which has a much higher standard deviation of 0.0046026. This low variability demonstrates CS-DE ability to provide highly reliable and repeatable results, crucial for sensitive optimization tasks. CS-DE also achieves the top Friedman rank of 1.3, highlighting its superior performance compared to its counterparts. In contrast, algorithms like PaDE and NDE trail behind, with ranks of 7.8 and 7, respectively. The close alignment of CS-DE minimum (0.1029149) and maximum (0.1029149) values further underscores its precision, with minimal fluctuations between runs, making it a robust choice for optimization

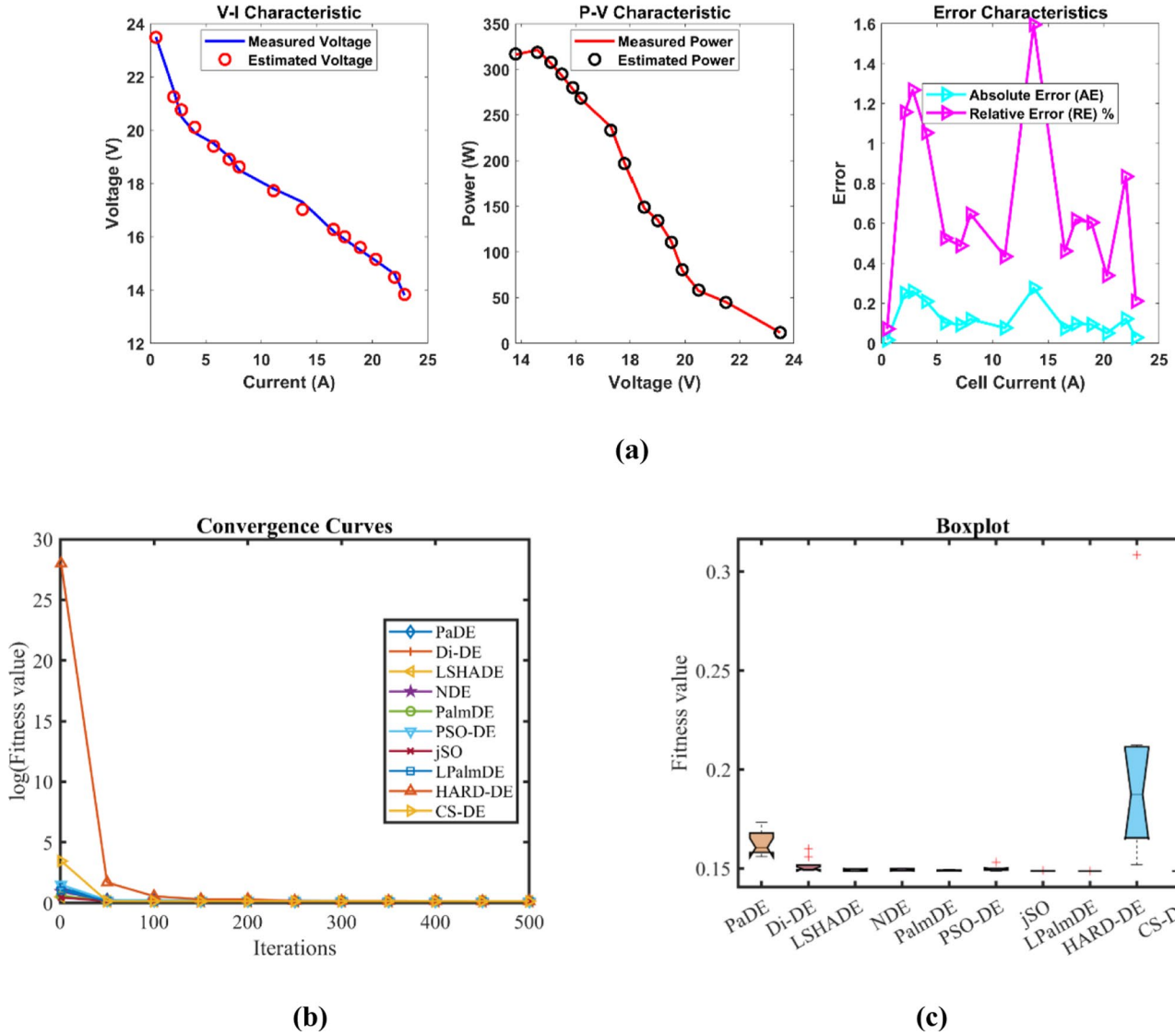

Fig. 10 CS-DE algorithm characteristic curves of CASE5. **a** V-I, P–V, and error curve. **b** Convergence curve. **c** Box-plot

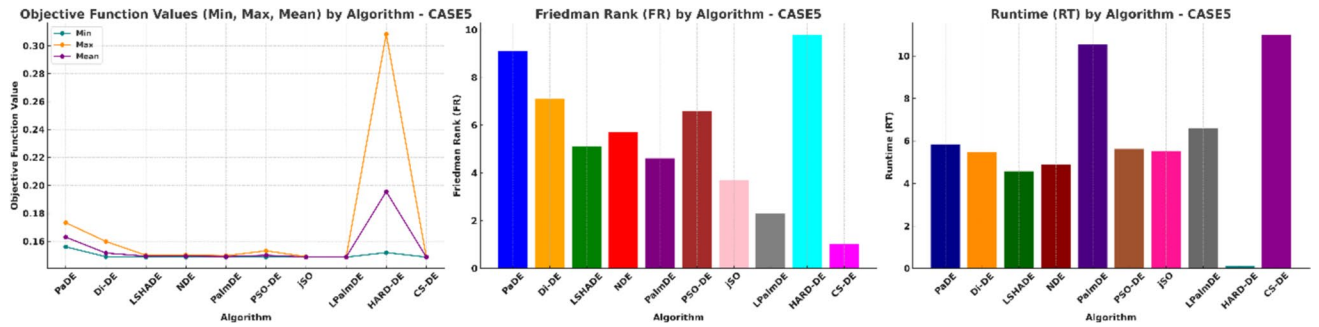

Fig. 11 Optimized parameters, Friedman rank, and runtime comparison for CASE5 algorithms

problems where stability is paramount. In terms of runtime, CS-DE completes the optimization task in just 0.1114574 s, far surpassing the runtime of more computationally intensive algorithms such as HARD-DE, which requires 12.066956 s to finish. This remarkable speed, coupled with CS-DE consistent performance, makes it ideal for time-sensitive applications.

Table 14 Optimized parameters and optimal function value for CASE6

Algorithm	PaDE	Di-DE	LSHADE	NDE	PalmDE	PSO-DE	jSO	LPalmDE	HARD-DE	CS-DE
ξ_1	−1.1905445	−0.8587458	−1.19969	−0.86332	−0.8625703	−0.8649573	−1.0649452	−1.0347459	−1.1505875	−1.1173399
ξ_2	0.0031893	0.0018932	0.0037695	0.0020834	0.0022321	0.0019398	0.0025182	0.0028576	0.0036091	0.0026583
ξ_3	5.84E−05	0.000036	0.000098	4.859E−05	5.941E−05	3.812E−05	3.71E−05	6.773E−05	0.000098	3.6E−05
ξ_4	−0.0001703	−0.0001697	−0.0001697	−0.0001699	−0.0001697	−0.0001692	−0.0001695	−0.0001696	−0.0001685	−0.0001697
λ	14	14	14	14	14	14.801292	14	14.0004	16.365757	14
R_c	0.0008	0.0008	0.0008	0.0008	0.0008	0.0007937	0.0008	0.0008	0.0007786	0.0008
B	0.0172314	0.0173186	0.0173175	0.0173298	0.0173175	0.0176716	0.0173481	0.017313	0.0166915	0.0173175
Min	0.2838233	0.2837738	0.2837738	0.2837851	0.2837738	0.2864205	0.2837891	0.2837799	0.3070598	0.2837738
Max	0.3714557	0.3196132	0.3054678	0.3001888	0.2847069	0.3212365	0.2839061	0.2838295	0.4243339	0.2837738
$Mean$	0.3232844	0.2921284	0.2859432	0.2859737	0.2839453	0.2995563	0.2838432	0.283799	0.3526751	0.2837738
Std	0.0359655	0.0141868	0.0068603	0.005102	0.0003415	0.0145939	4.086E−05	1.743E−05	0.0365857	2.035E−16
RT	5.4859315	5.0927722	4.2577451	4.5414	9.7266036	5.2840931	5.1512177	6.2414611	10.386682	0.09565
FR	8.1	5	2.4	6.4	4.3	8	5.5	4.3	9.8	1.2

Table 15 Performance metrics of CS-DE algorithm for CASE6

AE_v (A)	RE %	MBE
0.3446957	1.1736318	0.0091396
0.1485977	0.5549372	0.0016986
0.2846644	1.1255897	0.0062334
0.3183396	1.3110184	0.0077954
0.2704561	1.1549069	0.0056267
0.1623745	0.7140759	0.0020281
0.0154675	0.0701204	1.84E−05
0.1347336	0.6300039	0.0013964
0.2584281	1.2471358	0.0051373
0.3379988	1.6877999	0.0087879
0.3445644	1.7547272	0.0091327
0.2649755	1.3806697	0.0054009
0.4855086	2.6013624	0.0181322
0.2592926	**1.1850753**	**0.0061944**

As shown in Table 11, CS-DE maintains low absolute error values with an average relative error of 1.043091%, further proving its accuracy in estimating PEMFC parameters, making it a preferred solution for industrial-level optimization. V-I, P–V, and error curves, convergence behavior, and a statistical box plot are shown in Fig. 8 to illustrate CS-DE accuracy in parameter estimation and its ability to consistently minimize error. The optimized parameter values (min, max, and mean), Friedman rank, and runtime of each algorithm are shown in Fig. 9.

CASE5: Ballard Mark V PEMFC

In the optimization of the Ballard Mark V PEMFC, the CS-DE algorithm consistently outperforms other differential evolution variants, demonstrating exceptional stability across all metrics. From Table 12, it can be seen that CS-DE records the lowest minimum, maximum, and mean values at 0.1486318, with no variation between them. This consistency is supported by an extraordinarily low standard deviation of 2.83E−16, significantly lower than other algorithms such as LSHADE and Di-DE, which exhibit higher variability, with standard deviations of 0.0005772 and 0.003561, respectively. Despite a longer runtime of 11.001723 s, CS-DE superior precision justifies the time investment, especially when compared to faster algorithms like HARD-DE with a runtime of 0.1076193 s, but much higher variability. CS-DE achieves the best Friedman Rank of 1, highlighting its superior performance across all metrics. In Table 13, CS-DE demonstrates high accuracy with low absolute errors and an average relative error of 0.6870068%, showcasing its reliability in estimating PEMFC parameters. The combination of stability, precision, and consistent results makes CS-DE an ideal choice for applications requiring minimal variance in outcomes, making it the

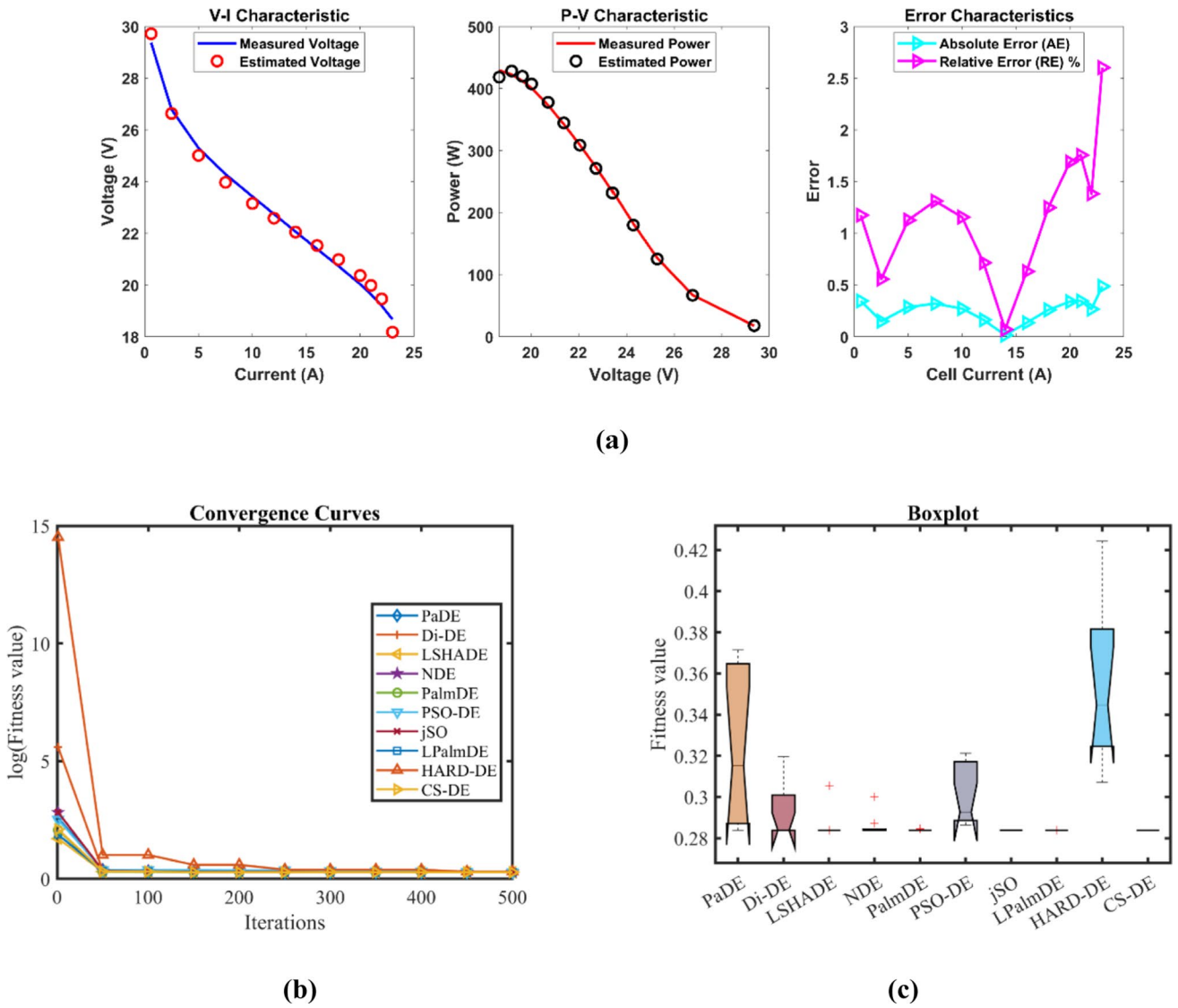

Fig. 12 CS-DE algorithm characteristic curves of CASE6 **a** V-I, P–V, and error curve. **b** Convergence curve. **c** Box-plot

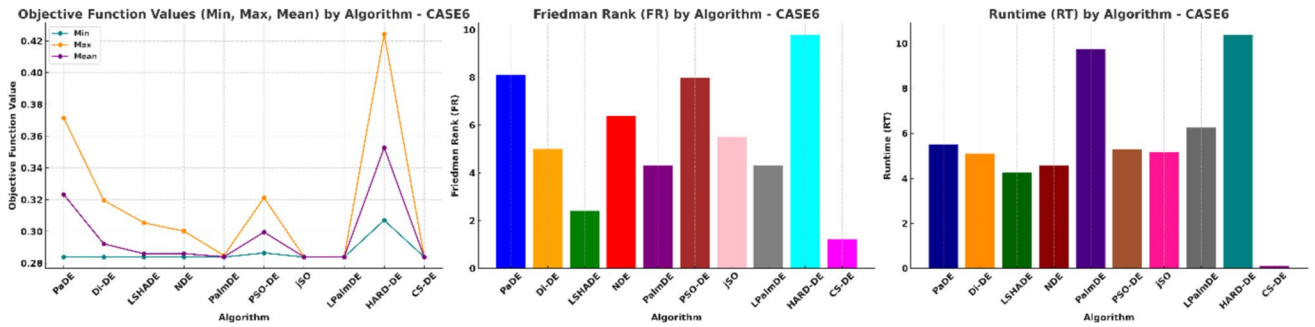

Fig. 13 Optimized parameters, Friedman rank, and runtime comparison for CASE6 algorithms

top-performing algorithm in this evaluation. V-I, P–V, and error curves, convergence behavior, and a statistical box plot are shown in Fig. 10 to illustrate CS-DE accuracy in parameter estimation and its ability to consistently minimize error. The optimized parameter values (min, max, and mean), Friedman rank, and runtime of each algorithm are shown in Fig. 11.

Table 16 Optimized parameters and optimal function value for CASE7

Algorithm	PaDE	Di-DE	LSHADE	NDE	PalmDE	PSO-DE	jSO	LPalmDE	HARD-DE	CS-DE
ξ_1	−0.9295078	−1.0462085	−1.19969	−0.9012092	−1.0116382	−1.1943815	−1.1395485	−0.9959834	−1.0544698	−0.8532347
ξ_2	0.0025965	0.0028779	0.0030687	0.0028168	0.0026419	0.0032301	0.0030342	0.0028706	0.0034104	0.0020119
ξ_3	6.274E−05	5.829E−05	3.944E−05	8.487E−05	4.836E−05	5.24E−05	4.974E−05	6.852E−05	9.558E−05	3.601E−05
ξ_4	−0.0001458	−0.0001488	−0.0001493	−0.0001486	−0.0001493	−0.0001493	−0.0001497	−0.0001492	−0.0001405	−−0.0001493
λ	22.040945	22.999996	23	22.693935	22.998403	22.939411	23	22.999684	17.682681	23
R_c	0.0001154	0.0001511	0.0001	0.0001	0.0001001	0.0001005	0.0001	0.0001	0.0001	0.0001
B	0.050974	0.0507967	0.0509795	0.0509649	0.0509539	0.0509959	0.0509486	0.0509843	0.0475256	0.0509795
Min	0.1246517	0.122401	0.1217552	0.1222131	0.1217594	0.1218667	0.1218253	0.1217557	0.1418891	0.1217552
Max	0.1462772	0.1271203	0.1802692	0.1284992	0.1224544	0.1325963	0.1226251	0.1217881	0.5677495	0.1217552
Mean	0.1361437	0.1243678	0.1366804	0.1253662	0.121844	0.1259847	0.1221793	0.1217689	0.2546877	0.1217552
Std	0.0071433	0.0014655	0.016236	0.0022176	0.0002153	0.0033892	0.0002871	9.934E−06	0.1356563	5.433E−16
RT	5.9226938	5.4609677	4.6554231	4.8706403	10.474236	5.5952898	5.5786365	6.6461541	11.061653	0.1031519
FR	8.5	6	7.1	6.2	2.9	6.4	4.2	2.6	10	1.1

Table 17 Performance metrics of CS-DE algorithm for CASE7

AE_v (A)	*RE* %	*MBE*
0.1270107	0.5597256	0.0010754
0.1715627	0.8498712	0.0019622
0.0349552	0.1812119	8.146E−05
0.1059533	0.5708478	0.0007484
0.0360279	0.198302	8.653E−05
0.0544565	0.3073236	0.0001977
0.0105944	0.0613421	7.483E−06
0.0427669	0.2602995	0.0001219
0.0248454	0.1582422	4.115E−05
0.0830911	0.5542845	0.0004603
0.2198183	1.5000362	0.0032213
0.1172164	0.8350296	0.000916
0.0596003	0.4516437	0.0002368
0.2821694	2.347753	0.005308
0.0734411	0.7249285	0.0003596
0.096234	**0.6373894**	**0.0009883**

CASE6: STD-1 PEMFC

In the STD-1 PEMFC optimization, the CS-DE algorithm demonstrates exceptional stability and precision. As shown in Table 14, CS-DE achieves the lowest minimum, maximum, mean, and standard deviation values, all at 0.2837738, indicating minimal variability and superior reliability, with a standard deviation of 2.04E − 16. This performance significantly outshines other algorithms like LSHADE and Di-DE. Additionally, CS-DE completes its task in just 0.09565 s, making it the fastest among all the algorithms tested, further reinforcing its efficiency. The Friedman rank of 1.2 solidifies CS-DE position as a top performer across all metrics. As seen in Table 15, CS-DE also delivers low absolute error values and an average relative error of 1.1850753%, demonstrating its accuracy in modeling PEMFC parameters. These results establish CS-DE as the optimal choice for applications that demand both precision and speed in optimization tasks, confirming its dominance in this evaluation**.** V-I, P–V, and error curves, convergence behavior and a statistical box plot are shown in Fig. 12 to illustrate CS-DE accuracy in parameter estimation and its ability to consistently minimize error. The optimized parameter values (min, max, and mean), Friedman rank, and runtime of each algorithm are shown in Fig. 13.

CASE7: HORIZON PEMFC

As seen in Table 16, CS-DE achieves the lowest minimum, maximum, and mean values, all at 0.1217552, paired with an extraordinarily low standard deviation of 5.43E − 16, indicating minimal variability for case 7. With a Friedman rank

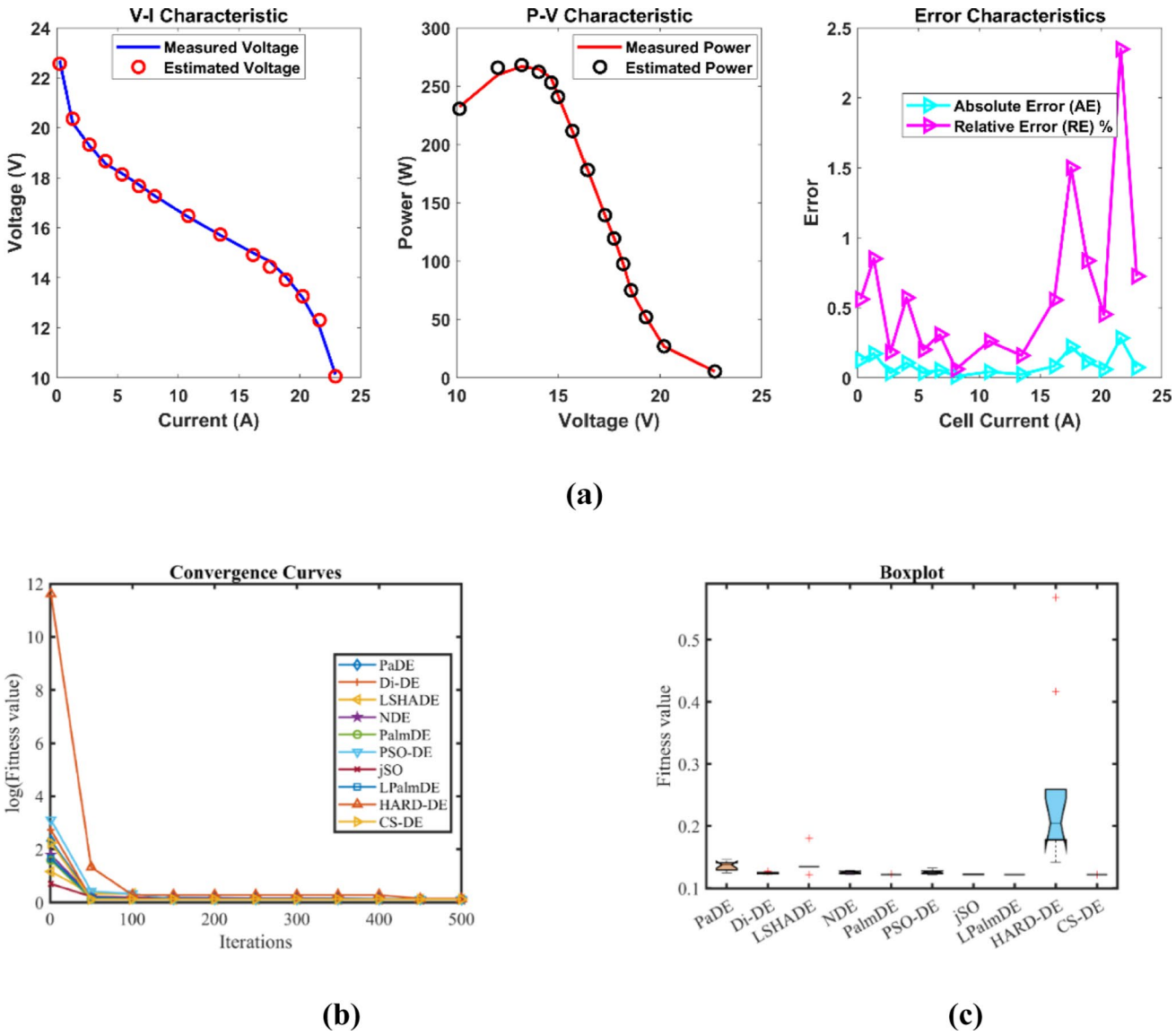

Fig. 14 CS-DE algorithm characteristic curves of CASE7. **a** V-I, P–V, and error curve. **b** Convergence curve. **c** Box-plot

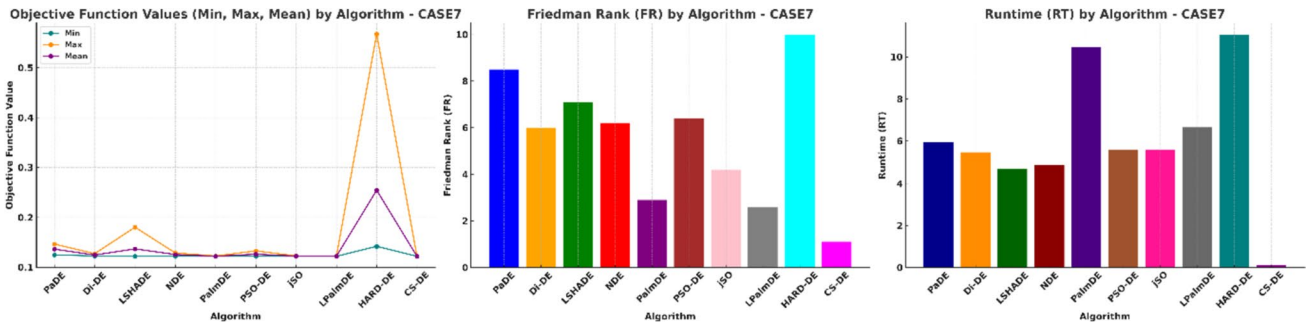

Fig. 15 Optimized parameters, Friedman rank, and runtime comparison for CASE7 algorithms

Table 18 Optimized parameters and optimal function value for CASE8

Algorithm	PaDE	Di-DE	LSHADE	NDE	PalmDE	PSO-DE	jSO	LPalmDE	HARD-DE	CS-DE
ξ_1	−1.19969	−1.0055194	−0.8845543	−1.132445	−1.1963952	−1.19969	−0.9026868	−1.0750459	−1.1447848	−0.9570624
ξ_2	0.0030971	0.0027545	0.0029354	0.0032593	0.00357	0.0036517	0.0021872	0.0031752	0.0029228	0.0023475
ξ_3	3.91E−05	5.664E−05	0.000098	6.708E−05	7.674E−05	8.23E−05	3.602E−05	7.356E−05	3.876E−05	3.612E−05
ξ_4	−0.0001487	−0.0001452	−0.0001464	−0.0001462	−0.0001464	−0.0001462	−0.000146	−0.0001465	−0.0001394	−0.0001464
λ	14.477651	14.879216	14.397706	14.480115	14.421235	14.311994	14.360664	14.407598	15.516331	14.397706
R_c	0.0001307	0.0003796	0.0001	0.0001519	0.0001065	0.0001018	0.0001	0.0001002	0.0005114	0.0001
B	0.0237168	0.0238906	0.0239744	0.0240551	0.0239942	0.0237518	0.0239313	0.0239717	0.0263163	0.0239744
Min	0.0806849	0.0794701	0.0784922	0.0787204	0.0785126	0.0785875	0.0785375	0.0784945	0.0931517	0.0784922
Max	0.098341	0.0895137	0.2160923	0.0828689	0.0807611	0.0900644	0.0789913	0.0785932	0.3878473	0.0784922
Mean	0.0882359	0.083777	0.1068236	0.0800073	0.0790483	0.0824591	0.0787647	0.0785173	0.205122	0.0784922
Std	0.0066232	0.0037601	0.0575965	0.0013719	0.0007447	0.0045166	0.0001634	3.138E−05	0.0946545	1.503E−16
RT	5.8949637	5.4685931	4.5812131	4.8046615	10.448789	5.5819265	5.4929295	6.5834013	11.086111	0.1009381
FR	8.1	7.4	6	6	3.9	6.5	3.8	2.5	9.8	1

Table 19 Performance metrics of CS-DE algorithm for CASE8

AE_v (A)	*RE* %	*MBE*
0.0543705	0.2336406	0.0001971
0.0793024	0.3771276	0.0004193
0.0431342	0.2148674	0.000124
0.0321293	0.1655985	6.882E−05
0.0030114	0.0159359	6.046E−07
0.0714099	0.3859013	0.00034
0.0268375	0.1486339	4.802E−05
0.040381	0.2335554	0.0001087
0.0077687	0.0470697	4.024E−06
0.0487684	0.3102393	0.0001586
0.0256133	0.1671114	4.374E−05
0.0659755	0.4401095	0.0002902
0.1436286	0.9876746	0.0013753
0.2068756	1.5223981	0.0028532
0.0440901	0.3520618	0.0001296
0.0595531	**0.3734617**	**0.0004107**

of 1.1, it stands out as the top performer across all tested algorithms. In terms of computational speed, CS-DE completes the task in just 0.1031519 s, significantly faster than algorithms like HARD-DE and LPalmDE, which have runtimes of 11.061653 s and 6.6461541 s, respectively. Table 17 further illustrates CS-DE precision, with an average relative error of 0.6373894% and consistently low absolute errors, making it the most reliable choice for scenarios demanding high accuracy and efficiency. V-I, P–V, and error curves, convergence behavior, and a statistical box plot are shown in Fig. 14 to illustrate CS-DE accuracy in parameter estimation and its ability to consistently minimize error. The optimized parameter values (min, max, and mean), Friedman rank, and runtime of each algorithm are shown in Fig. 15.

CASE8: STD-2 PEMFC

In the comparative analysis of various differential evolution algorithms as shown in Table 18, CS-DE consistently showcases its exceptional performance across all metrics. As seen in Table 18, CS-DE achieves the lowest minimum, maximum, and mean values at 0.0784922, with an exceptionally low standard deviation of 1.50E − 16, indicating minimal variability across runs. With a Friedman rank of 1, CS-DE secures its position as the top-performing algorithm. Additionally, its runtime of 0.1009381 s is the fastest among all tested algorithms, reinforcing its computational efficiency.

Table 19 further highlights CS-DE accuracy, showing low absolute errors and an average relative error of 0.3734617%, confirming its reliability in estimating PEMFC parameters. This combination of speed, precision, and stability makes CS-DE a robust choice for high-precision optimization

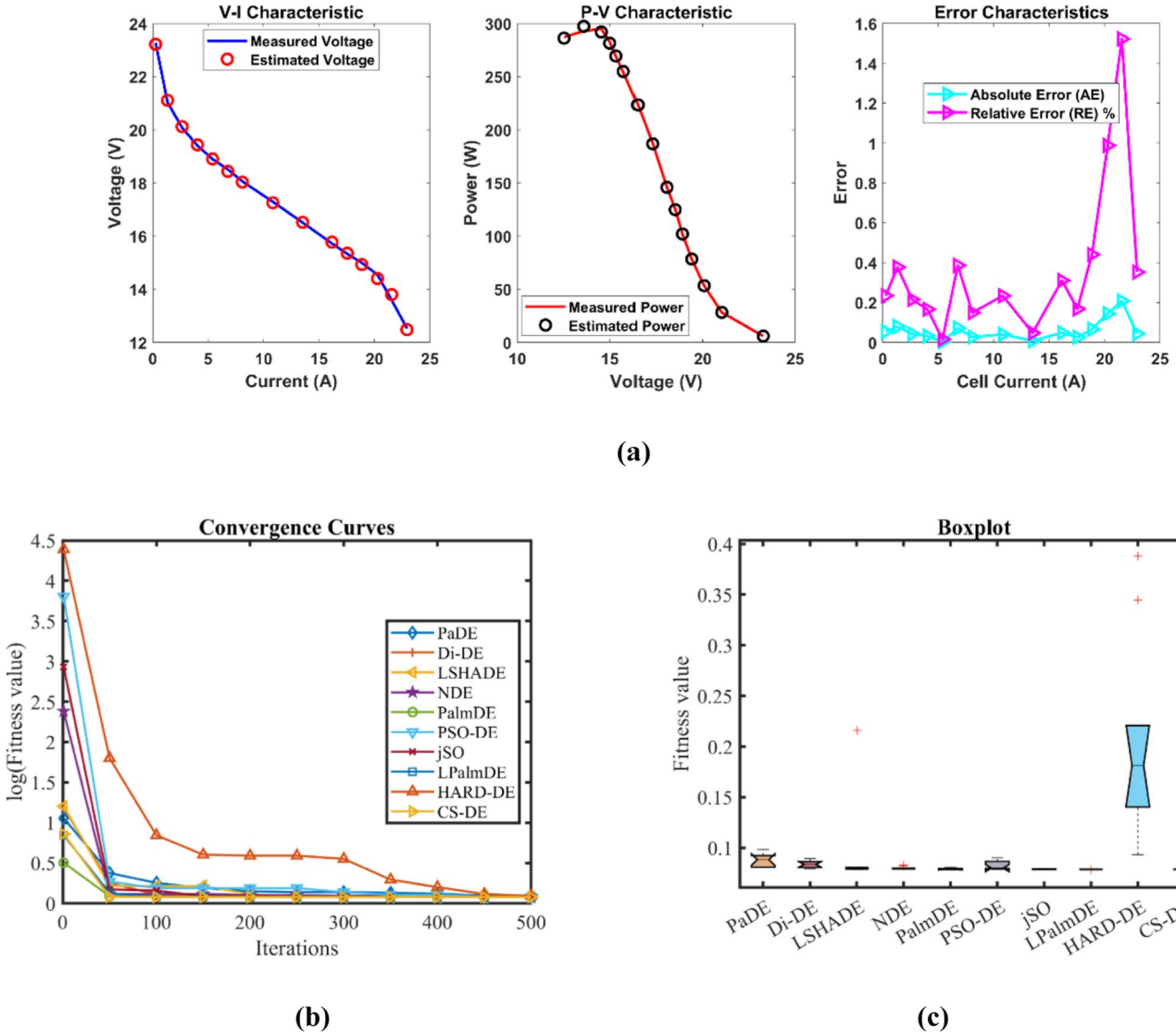

Fig. 16 CS-DE algorithm characteristic curves of CASE8. **a** V-I, P–V, and error curve. **b** Convergence Curve, (c) Box-Plot

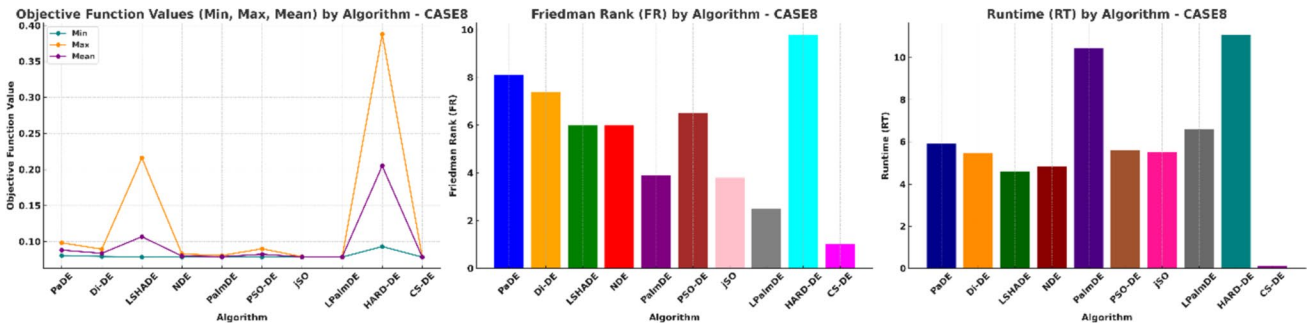

Fig. 17 Optimized parameters, Friedman rank, and runtime comparison for CASE8 algorithms

tasks. V-I, P–V, and error curves, convergence behavior and a statistical box plot are shown in Fig. 16 to illustrate CS-DE accuracy in parameter estimation and its ability to consistently minimize error. The optimized parameter values (min, max, and mean), Friedman rank, and runtime of each algorithm are shown in Fig. 17.

Table 20 Optimized parameters and optimal function value for CASE9

Algorithm	PaDE	Di-DE	LSHADE	NDE	PalmDE	PSO-DE	jSO	LPalmDE	HARD-DE	CS-DE
ξ_1	−1.1708391	−0.8577742	−1.1930581	−0.8661497	−1.1380555	−0.8559499	−0.9991754	−0.916381	−0.9740608	−0.966741
ξ_2	0.0031963	0.0018864	0.002965	0.0023114	0.0030044	0.0019901	0.0029877	0.0024017	0.0028344	0.0023514
ξ_3	6.915E−05	3.833E−05	4.616E−05	6.931E−05	6.162E−05	4.671E−05	9.171E−05	6.5E−05	8.445E−05	4.974E−05
ξ_4	−0.00012	−0.0001208	−0.0001208	−0.000121	−0.0001208	−0.0001191	−0.0001205	−0.0001208	−0.0001223	−0.0001208
λ	23	23	23	22.99482	23	22.913285	23	22.999618	21.746326	23
R_c	0.0001	0.0001	0.0001	0.0001025	0.0001	0.0002193	0.0001	0.0001003	0.0005533	0.0001
B	0.0627198	0.0624791	0.0624799	0.0623798	0.0624794	0.0622205	0.0625278	0.0624673	0.0590856	0.0624799
Min	0.202416	0.2023193	0.2023192	0.2023584	0.2023193	0.2033618	0.202327	0.2023213	0.2164388	0.2023192
Max	0.2251206	0.2076994	0.2096986	0.2070415	0.202421	0.2204882	0.2029691	0.2023514	0.3917155	0.2023192
Mean	0.2095081	0.2043197	0.2060089	0.2035668	0.2023418	0.2062041	0.2025474	0.2023281	0.2874547	0.2023192
Std	0.0068422	0.0020192	0.0038893	0.0016292	3.356E−05	0.0053777	0.0002131	8.877E−06	0.0590506	4.342E−16
RT	5.8085381	5.4410136	4.5488471	4.8204224	10.401666	5.5869888	5.7378028	6.6253058	11.12699	0.1027052
FR	7.9	6.3	5.25	6	3.4	7.4	4.8	3	9.8	1.15

Table 21 Performance metrics of CS-DE algorithm for CASE9

AE_v (A)	*RE %*	*MBE*
0.0057891	0.0269087	2.234E−06
0.0957913	0.4869003	0.0006117
0.0529961	0.2831683	0.0001872
0.1308161	0.7289878	0.0011409
0.0431608	0.2459347	0.0001242
0.0009236	0.0053843	5.687E−08
0.0743136	0.4454106	0.0003682
0.1279056	0.8056945	0.0010907
0.0709042	0.4682899	0.0003352
0.1111188	0.7682757	0.0008232
0.2285785	1.6226198	0.0034832
0.3110241	2.2904449	0.0064491
0.1294831	1.0213853	0.0011177
0.6016726	5.5329778	0.024134
0.1264298	1.4171678	0.0010656
0.1407272	**1.0766367**	**0.0027289**

CASE9: STD-3 PEMFC

For case 9 analysis as well, CS-DE consistently showcases superior stability and precision. As shown in Table 20, CS-DE achieves the lowest minimum, maximum, and mean values, all at 0.2023192, highlighting its minimal variability and robustness across runs. With an impressively low standard deviation of 4.34E − 16, CS-DE outperforms its counterparts by a significant margin, confirming its consistency and accuracy. Additionally, CS-DE completes the task in just 0.1027052 s, the fastest among all algorithms tested, further solidifying its efficiency. The Friedman rank of 1.15 reinforces CS-DE top placement in performance metrics. As detailed in Table 21, CS-DE maintains low absolute error values, with an average relative error of 1.0766367%, making it a reliable choice for applications requiring both precision and speed in optimization tasks. V-I, P–V, and error curves, convergence behavior, and a statistical box plot are shown in Fig. 18 to illustrate CS-DE accuracy in parameter estimation and its ability to consistently minimize error. The optimized parameter values (min, max, and mean), Friedman rank, and runtime of each algorithm are shown in Fig. 19.

CASE10: STD-4 PEMFC

Similar to previous cases, CS-DE excels in optimizing the STD-4 PEMFC across all metrics. As shown in Table 22, it achieves the lowest minimum, maximum, and mean values, all recorded at 0.1044462, reflecting its superior precision. With a standard deviation of 2.84E-15, CS-DE demonstrates exceptional stability, with minimal variability across runs.

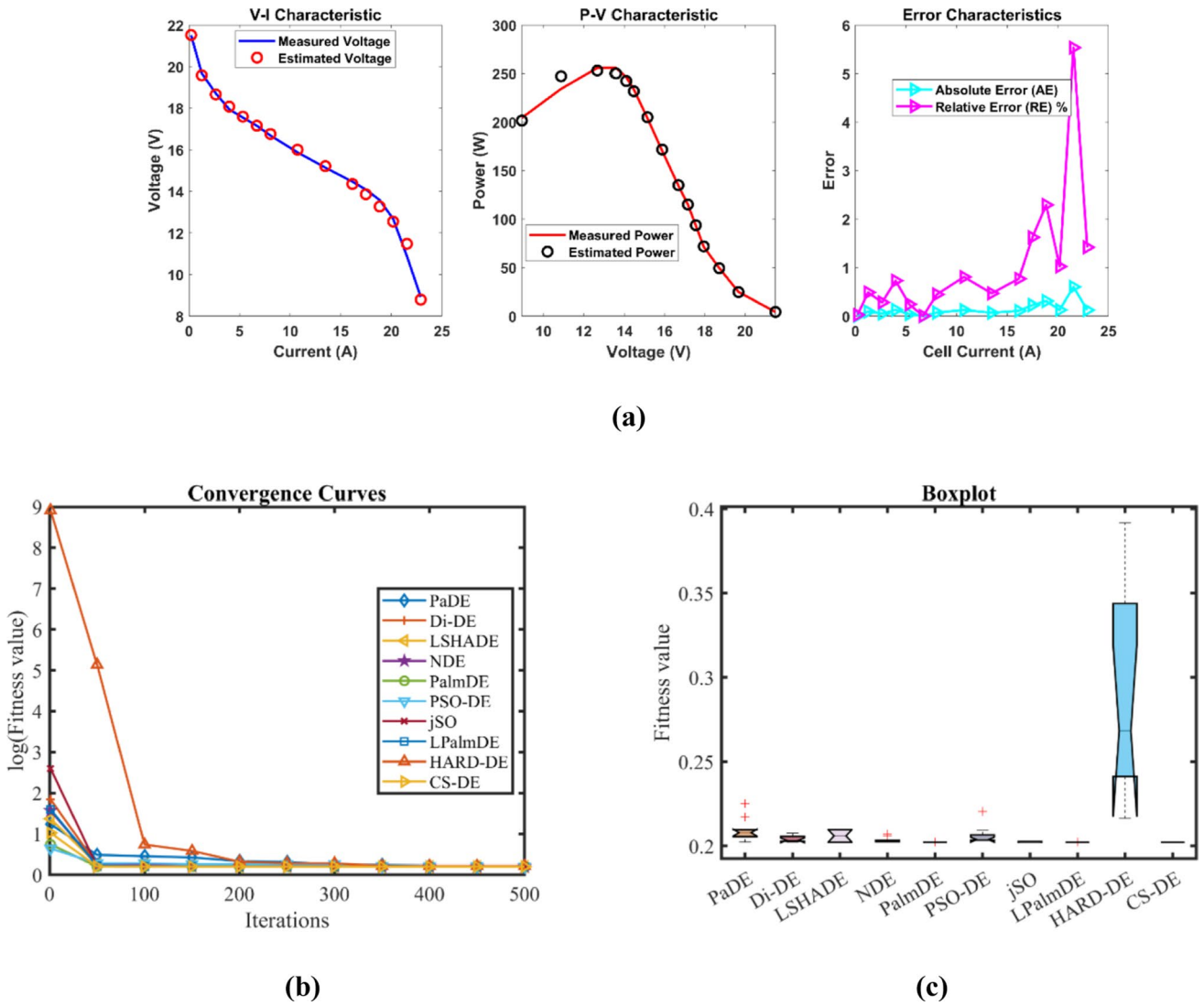

Fig. 18 CS-DE algorithm characteristic curves of CASE9. **a** V-I, P–V, and error curve. **b** Convergence curve. **c** Box-plot

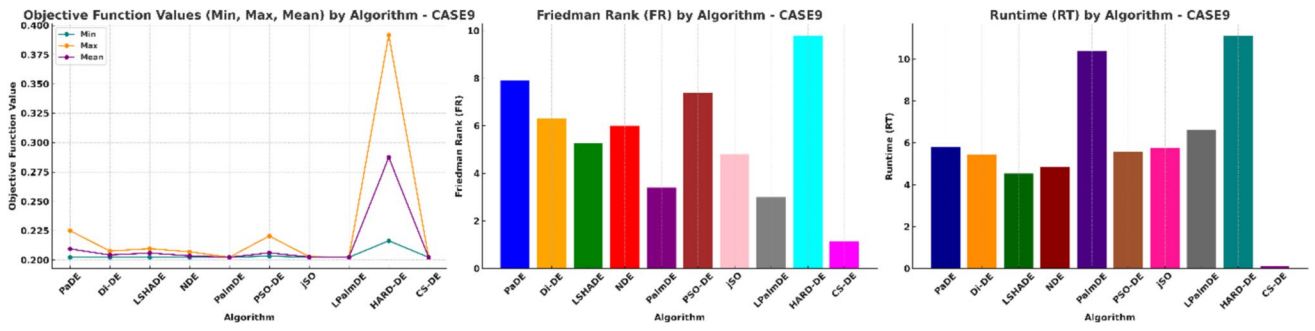

Fig. 19 Optimized parameters, Friedman rank, and runtime comparison for CASE9 algorithms

The algorithm also shows remarkable computational efficiency, achieving the fastest runtime of 0.1034254 s.

The Friedman rank of 1 reinforces CS-DE top-tier performance, consistently outperforming other algorithms. In Table 23, the algorithm maintains low absolute errors and an average relative error of 0.4772317%, underlining its reliability and precision in parameter estimation. This analysis, as in previous cases, confirms CS-DE as the optimal

Table 22 Optimized parameters and optimal function value for CASE10

Algorithm	PaDE	Di-DE	LSHADE	NDE	PalmDE	PSO-DE	jSO	LPalmDE	HARD-DE	CS-DE
ξ_1	−1.1936294	−0.9336266	−0.8532	−1.1906956	−0.8660539	−1.1726808	−1.1388252	−0.9846997	−0.8532	−0.8532347
ξ_2	0.0032681	0.0025952	0.0020673	0.0037526	0.0024613	0.0030994	0.0028936	0.0026752	0.0021828	0.0023183
ξ_3	5.43E−05	5.917E−05	0.000036	9.227E−05	6.36E−05	4.587E−05	3.735E−05	5.418E−05	4.432E−05	5.537E−05
ξ_4	−0.0001359	−0.0001379	−0.0001372	−0.0001373	−0.0001372	−0.0001374	−0.0001373	−0.0001372	−0.000146	−0.0001372
λ	14	14.000672	14	14	14.000008	14.142853	14	14.000379	15.121589	14
R_c	0.0008	0.0007324	0.0008	0.0007838	0.0008	0.0007159	0.0008	0.0007989	0.0003605	0.0008
B	0.0160983	0.0157221	0.0155029	0.0155824	0.0155051	0.0164485	0.0153369	0.015496	0.0207637	0.0155029
Min	0.1052128	0.1048641	0.1044462	0.1045551	0.1044463	0.1060584	0.1045248	0.1044567	0.1346127	0.1044462
Max	0.1557976	0.1187262	0.1446772	0.1129003	0.1056227	0.1230138	0.1050216	0.1045681	0.2064071	0.1044462
Mean	0.1171594	0.1095498	0.1164698	0.1080536	0.104715	0.1110394	0.1046908	0.1044914	0.1675863	0.1044462
Std	0.0158099	0.0045222	0.0153204	0.0036927	0.0003643	0.0052944	0.0001857	4.137E−05	0.0277357	2.839E−15
RT	5.924132	5.4800615	4.5952761	4.934091	10.500752	5.588891	5.552335	6.6620768	11.088897	0.1034254
FR	7.6	6.8	6.5	5.8	3.7	7.1	3.9	2.7	9.9	1

Table 23 Performance metrics of CS-DE algorithm for CASE10

AE_v (A)	*RE* %	*MBE*
0.0669869	0.2845541	0.0002991
0.0802425	0.3736449	0.0004293
0.1837387	0.902964	0.0022507
0.0002844	0.0014292	5.391E−09
0.0966363	0.4964824	0.0006226
0.0955673	0.50265	0.0006089
0.0188266	0.1017382	2.363E−05
0.1001457	0.5599891	0.0006686
0.2134348	1.2350982	0.003037
0.1498854	0.9247107	0.0014977
0.1231709	0.7761193	0.0010114
0.0649544	0.4182188	0.0002813
0.0222603	0.1465237	3.303E−05
0.0172785	0.1181175	1.99E−05
0.0434665	0.3162352	0.000126
0.0851253	**0.4772317**	**0.0007273**

choice for tasks requiring high accuracy and computational speed. V-I, P–V, and error curves, convergence behavior, and a statistical box plot are shown in Fig. 20 to illustrate CS-DE accuracy in parameter estimation and its ability to consistently minimize error. The optimized parameter values (min, max, and mean), Friedman rank, and runtime of each algorithm are shown in Fig. 21.

CASE11: H-12–2 PEMFC

In case 11 (H-12–2 PEMFC), the CS-DE algorithm once again demonstrates its superior performance across various metrics, as reflected in Tables 24 and 25. The minimum, maximum, and mean values for CS-DE all hover consistently at 0.0754843, underlining its exceptional stability across runs. This consistency is reinforced by the extremely low standard deviation of 6.735E − 17, further validating the minimal variability in results. CS-DE also achieves a top Friedman rank of 1, reaffirming its ability to outperform other differential evolution variants. Its efficient runtime of 0.1002995 s ensures that the algorithm not only achieves accuracy but also delivers fast computation. These findings suggest that CS-DE is highly suitable for PEMFC optimization tasks requiring precise and stable solutions, especially when computational efficiency is also crucial. V-I, P–V, and error curves, convergence behavior and a statistical box plot are shown in Fig. 22 to illustrate CS-DE accuracy in parameter estimation and its ability to consistently minimize error. The optimized parameter values (min, max, and mean), Friedman rank, and runtime of each algorithm are shown in Fig. 23.

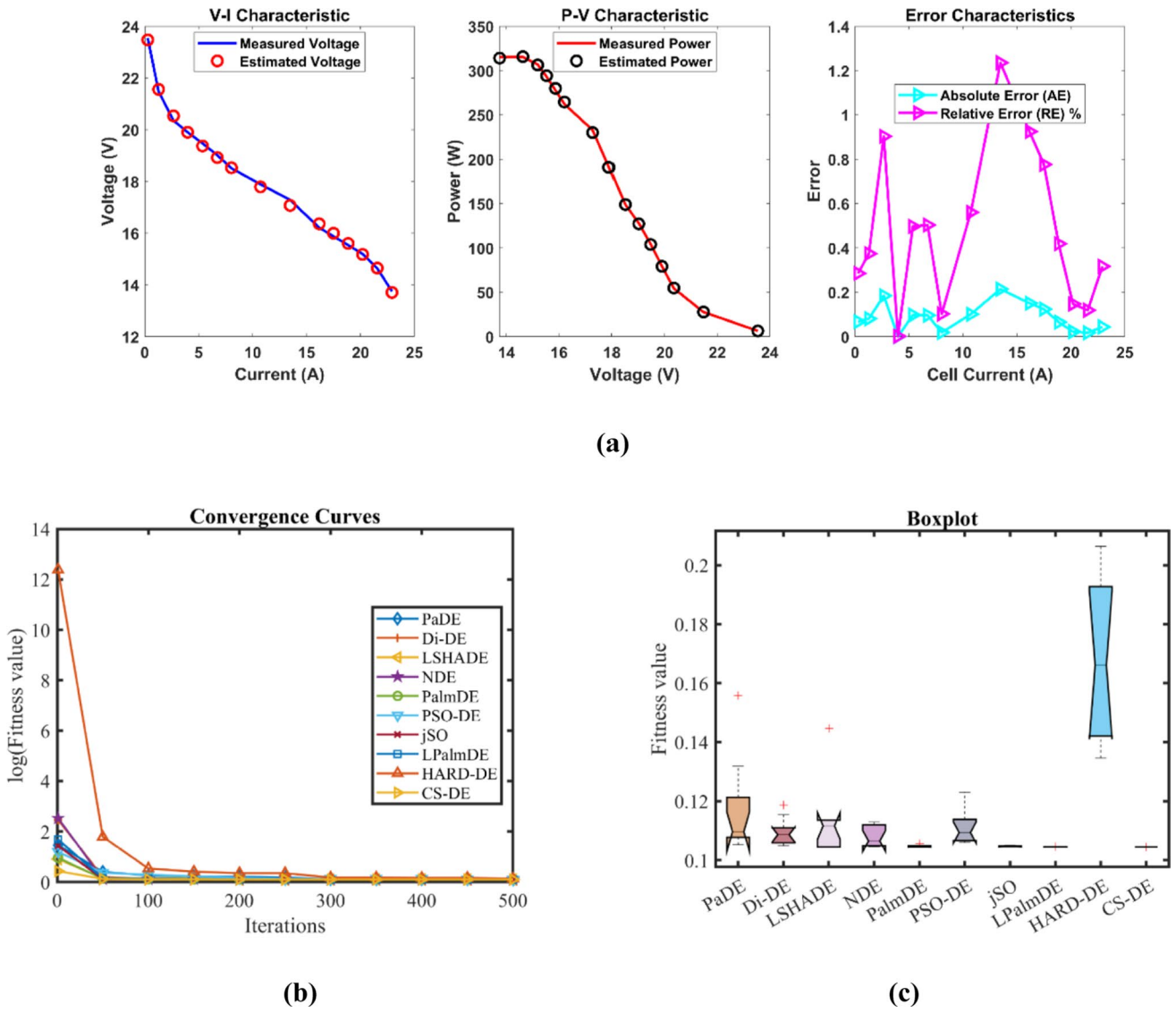

Fig. 20 CS-DE algorithm characteristic curves of CASE10. **a** V-I, P–V, and error curve. **b** Convergence curve. **c** Box-plot

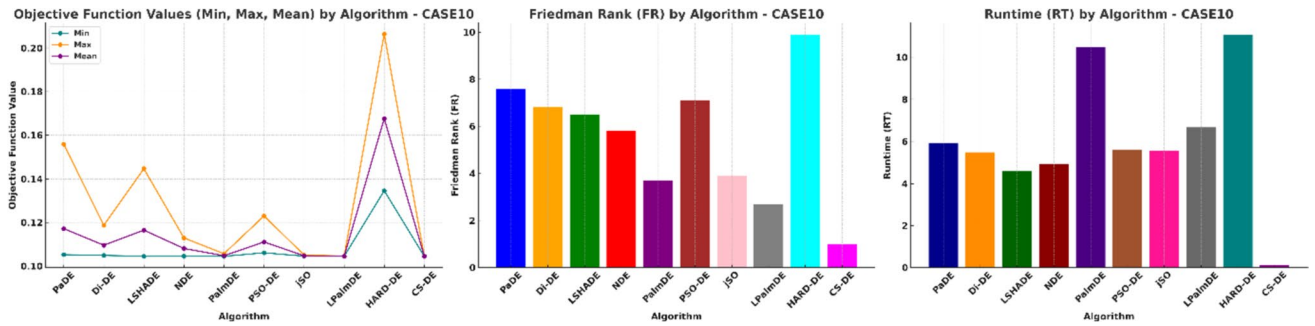

Fig. 21 Optimized parameters, Friedman rank, and runtime comparison for CASE10 algorithms

Table 24 Optimized parameters and optimal function value for CASE11

Algorithm	PaDE	Di-DE	LSHADE	NDE	PalmDE	PSO-DE	jSO	LPalmDE	HARD-DE	CS-DE
ξ_1	−0.8532	−0.8532004	−1.0437348	−0.8579672	−0.964992	−0.8584672	−0.9170274	−0.9808256	−0.9229903	−0.8532009
ξ_2	0.0015969	0.0017476	0.0021983	0.0018191	0.0026734	0.001585	0.0020363	0.0023055	0.0022719	0.0015674
ξ_3	3.81E−05	4.903E−05	0.000036	5.308E−05	8.924E−05	3.605E−05	5.464E−05	5.881E−05	7.08E−05	3.6E−05
ξ_4	−0.0000954	−0.0000954	−0.0000954	−0.0000954	−9.54E−05	−0.0000954	−0.0000954	−9.54E−05	−0.0000954	−0.0000954
λ	23	22.998818	23	22.571603	22.999992	22.973771	23	22.998706	20.846783	23
R_c	0.0001	0.0001273	0.0001	0.0001795	0.0001	0.0001	0.0001	0.0001001	0.00033	0.0001
B	0.0351467	0.0347781	0.0348125	0.0346621	0.0348124	0.0344949	0.0348164	0.0348674	0.0321975	0.0348125
Min	0.0755128	0.0754947	0.0754843	0.0755345	0.0754843	0.0755081	0.0754846	0.0754853	0.0779197	0.0754843
Max	0.0835513	0.075832	0.1129213	0.0760262	0.0758798	0.0758171	0.0754949	0.0754938	0.1178746	0.0754843
Mean	0.0771726	0.0755834	0.0794329	0.0757439	0.0755694	0.0756343	0.0754879	0.0754898	0.0997656	0.0754843
Std	0.002491	9.801E−05	0.0117691	0.0001618	0.000129	0.0001056	3.69E−06	2.809E−06	0.0158126	6.735E−17
RT	5.8794145	5.4223376	4.5218183	4.7766353	10.332518	5.5923375	5.4826009	6.6480166	10.962875	0.1002995
FR	8.4	5.6	5.6	7.1	4.7	6.1	3	3.5	10	1

Table 25 Performance metrics of CS-DE algorithm for CASE11

AE_v (A)	*RE* %	*MBE*
0.1779904	1.8676852	0.002112
0.0584	0.6226008	0.0002274
0.0442876	0.481387	0.0001308
0.127383	1.3786042	0.0010818
0.1117782	1.2283319	0.000833
0.0566126	0.63325	0.0002137
0.0414025	0.4683542	0.0001143
0.0427902	0.4890304	0.0001221
0.0414616	0.4787716	0.0001146
0.024216	0.2865795	3.909E−05
0.0193555	0.230149	2.498E−05
0.0880595	1.0738959	0.000517
0.0381485	0.4686546	9.702E−05
0.0032692	0.0403105	7.125E−07
0.0323122	0.4039019	6.961E−05
0.0604978	**0.6767671**	**0.0003799**

CASE12: H-12–3 PEMFC

Similarly, in case 12 (H-12–3 PEMFC), the results follow a comparable trend, as detailed in Tables 26 and 27. CS-DE maintains its dominant position with minimum, maximum, and mean values pinned at 0.0641935, highlighting its reliability. The standard deviation for CS-DE is negligible, standing at 2.727E−06, reinforcing its consistent performance across the board. Once again, CS-DE achieves the lowest runtime, clocking in at 0.1008514 s, and holds the best Friedman rank of 1.3. In Table 27, consistent accuracy, low error rates, and efficiency shown in these cases solidify CS-DE as the optimal choice for PEMFC parameter optimization, offering a blend of precision and speed crucial for advanced engineering applications**.** V-I, P–V, and error curves, convergence behavior and a statistical box plot are shown in Fig. 24 to illustrate CS-DE accuracy in parameter estimation and its ability to consistently minimize error. The optimized parameter values (min, max, and mean), Friedman rank, and runtime of each algorithm are shown in Fig. 25.

In the evaluation of CS-DE across all 12 cases, Figs. 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, and 25 provide a comprehensive visualization of the algorithm superiority over other differential evolution variants. Each figure comprises three parts: They include (a) V-I, P–V, and error curves that show how well CS-DE has performed in predicting voltage, current and power compared to the experimental data; (b) convergence curves that indicate the time taken by the algorithm to reach the optimal solutions; and (c) box-plots that demonstrate the trend of performance based on.

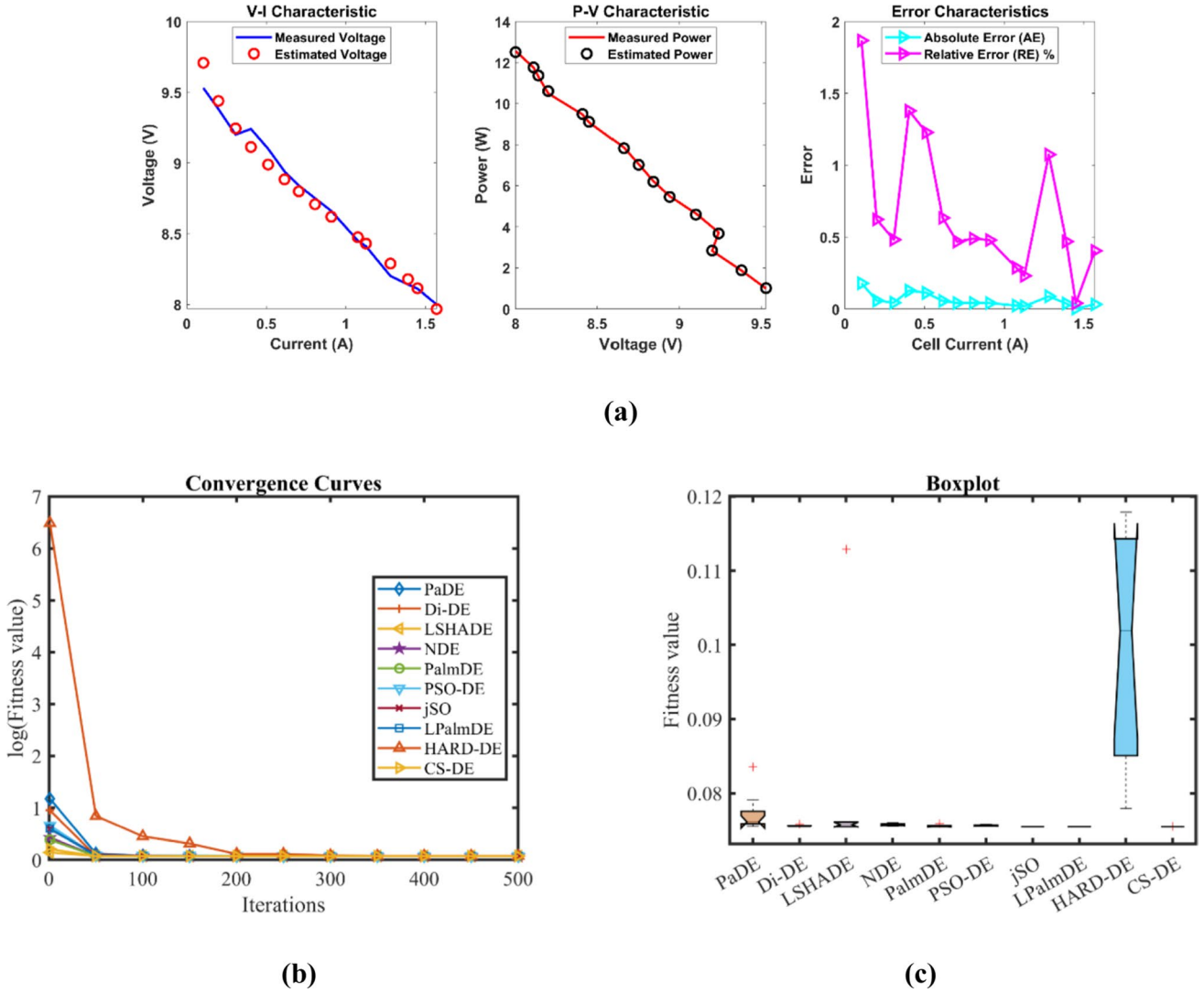

Fig. 22 CS-DE algorithm characteristic curves of CASE11. **a** V-I, P–V, and error curve. **b** Convergence curve. **c** Box-plot

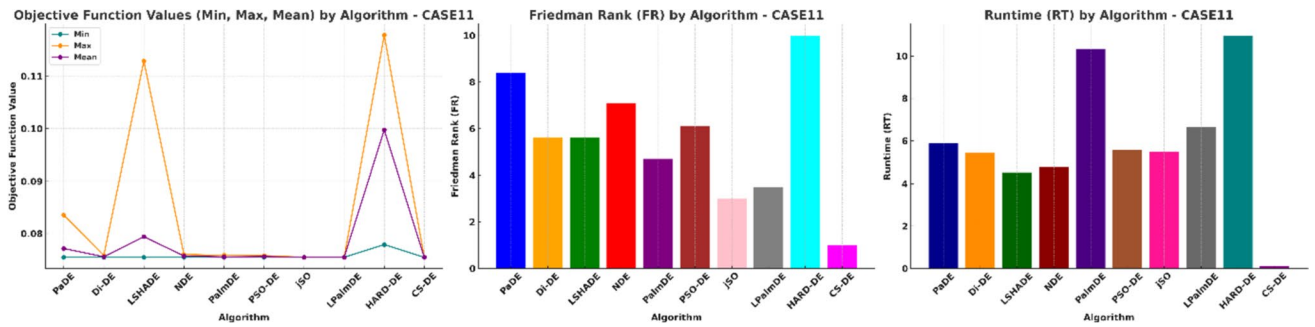

Fig. 23 Optimized parameters, Friedman rank, and runtime comparison for CASE11 algorithms

The V-I and P–V curves of CS-DE provide a clear representation of the good correlation between the simulated results and experimental data with low predictive error. This is also supported by the error curves where CS-DE has much lower error rates than the rest of the methods. The convergence plots for all the figures show that CS-DE converges to the correct solutions much faster than other algorithms, an essential factor in applications that require both accuracy

Table 26 Optimized parameters and optimal function value for CASE12

Algorithm	PaDE	Di-DE	LSHADE	NDE	PalmDE	PSO-DE	jSO	LPalmDE	HARD-DE	CS-DE
ξ_1	−1.1356031	−1.0923903	−0.8532	−0.972039	−0.9375896	−1.0436731	−0.9912693	−1.0157926	−0.9303495	−0.8532
ξ_2	0.0032422	0.002505	0.0024747	0.0020107	0.0024597	0.0022734	0.0026584	0.0023743	0.0018571	0.0024301
ξ_3	8.806E−05	4.492E−05	0.000098	3.711E−05	7.743E−05	3.95E−05	7.935E−05	5.32E−05	0.000036	9.479E−05
ξ_4	−0.0000954	−0.0000954	−0.0000954	−0.0000954	−9.54E−05	−0.0000954	−0.0000954	−9.54E−05	−0.0000954	−0.0000954
λ	14.544555	14.036749	14	14.514155	14.009357	14.076102	14	14.00094	22.083018	14
R_c	0.0008	0.0007837	0.0008	0.000326	0.0005065	0.0006343	0.0008	0.0007207	0.0003187	0.0008
B	0.048206	0.0485087	0.0484826	0.0492059	0.0488506	0.048583	0.0485067	0.0485895	0.0490154	0.0484826
Min	0.0643057	0.064194	0.0641935	0.0642039	0.0641968	0.0641979	0.0641938	0.0641949	0.0651897	0.0641935
Max	0.091293	0.0642004	0.064255	0.0642865	0.0643752	0.0643091	0.0641994	0.0642101	0.1128251	0.0642022
Mean	0.0786411	0.0641971	0.0642225	0.0642354	0.0642263	0.0642485	0.0641961	0.0642	0.0849674	0.0641944
Std	0.0107064	2.495E−06	2.779E−05	2.409E−05	5.307E−05	3.495E−05	1.745E−06	4.737E−06	0.0184963	2.727E−06
RT	5.8161762	5.395198	4.5578785	4.7959245	10.436458	5.575082	5.4588192	6.5870096	10.972999	0.1008514
FR	9.1	3.2	5.2	6.6	5.6	7.4	2.8	4	9.8	1.3

Table 27 Performance metrics of CS-DE algorithm for CASE12

AE_v (A)	*RE %*	*MBE*
0.1296783	1.313863	0.0011211
0.0867594	0.8817007	0.0005018
0.0028344	0.0290113	5.356E−07
0.0307866	0.3173879	6.319E−05
0.0365849	0.3806964	8.923E−05
0.0623187	0.6498297	0.0002589
0.0637802	0.6713704	0.0002712
0.07016	0.7463832	0.0003282
0.0688984	0.7440431	0.0003165
0.0030899	0.034142	6.365E−07
0.0165248	0.1850483	1.82E−05
0.0235637	0.2668601	3.702E−05
0.0902829	1.0571771	0.0005434
0.0611491	0.7262358	0.0002493
0.0694636	0.8399472	0.0003217
0.0543917	**0.5895797**	**0.0002747**

and efficiency. Last but not least, the box-plots show that CS-DE performs well in all trials with much less variance than other algorithms, which reveals the stability of the proposed algorithm.

In conclusion, the evaluation of all the results indicates that the CS-DE algorithm outperforms the other two algorithms based on the accuracy, efficiency, CS-DE is identified as the most superior algorithm in this comparative analysis due to its low error rates, short convergence times, and stability.

Discussion on the findings

Several important implications for further research and practical applications of CS-DE and other differential evolution variants in PEMFC parameter estimation can be derived from the obtained results. The above results demonstrate that the CS-DE model performs more stably, accurately, and efficiently than other models in all cases and has the potential to be applied to the future design and application of PEMFC systems.

Another major strength of CS-DE is that it provides very high accuracy with low standard deviation between different runs, which is important for obtaining consistent and repeatable performance of the fuel cell under practical conditions. Due to the sensitivity of fuel cells on the operating conditions, the accurate parameter estimation is vital to have the PEMFCs operate near its optimum efficiency and hence minimize the energy losses and enhance the overall system life. The small error and fast convergence rate of CS-DE in the V-I and P–V curves indicate that this algorithm is

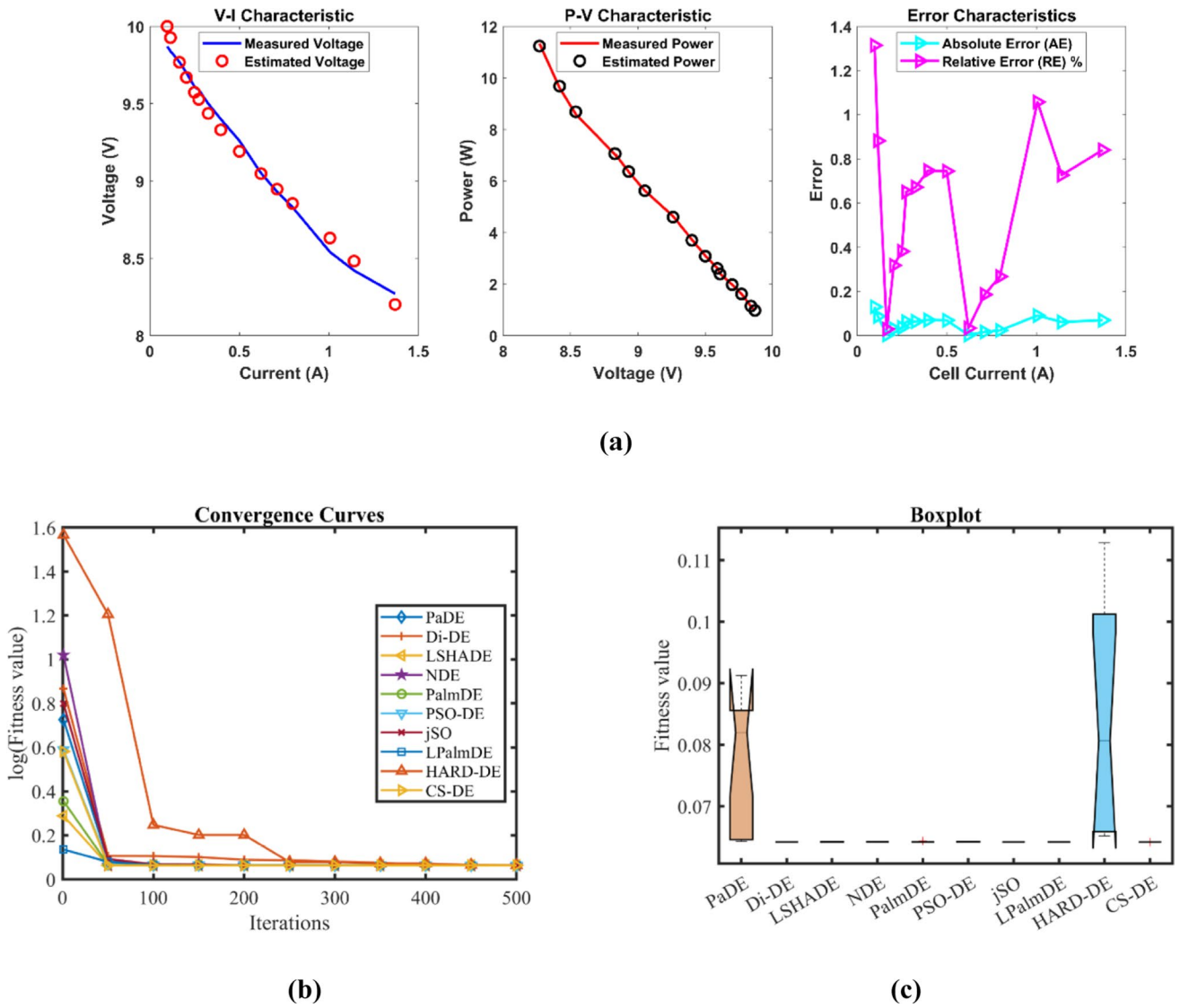

Fig. 24 CS-DE algorithm characteristic curves of CASE12. **a** V-I, P–V, and error curve. **b** Convergence curve. **c** Box-plot

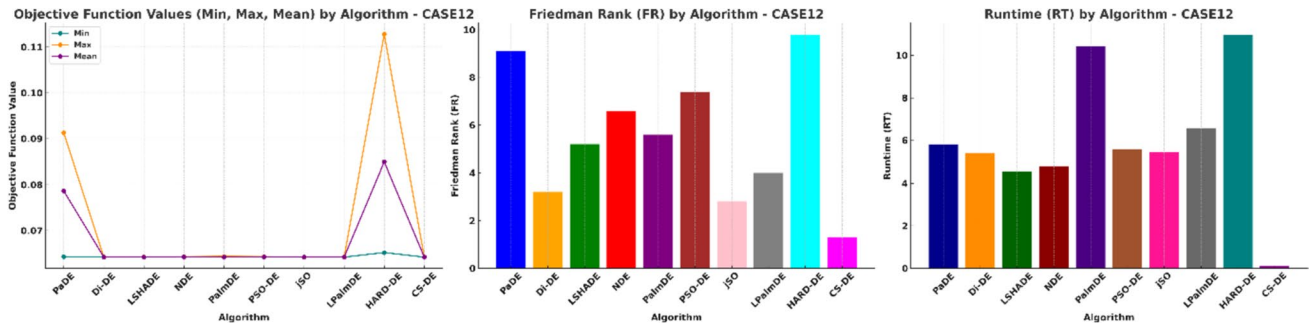

Fig. 25 Optimized parameters, Friedman rank, and runtime comparison for CASE12 algorithms

best applied in situations where small changes could greatly adversely affect fuel cell performance.

Moreover, the low computational complexity of CS-DE, which can be observed from its shorter execution time than other algorithms, has a great potential for real-time applications. This could be of great value especially for applications that require frequent changes in operating conditions like electric vehicles or portable power systems. The fast convergence of CS-DE guarantees that these systems can run optimally even under conditions of varying load or temperature.

Further, the box plot and the convergence curve depict that the CS-DE algorithm has minimal variances and no outliers, making it a suitable optimization method for large-scale applications or any other In applications that are emerging today in the transportation and stationary power generation industries, where PEMFCs are gradually becoming more prevalent, the utilization.

In conclusion, the presented work not only confirms the superiority of CS-DE in estimating the PEMFC parameters but also indicates that it can be used in other applications which require accurate, In renewable energy.

Conclusion

Parameter estimation for PEMFCs is the focus of this study, to improve PEMFC simulations, control, and performance assessments. The CS-DE algorithm identifies optimal parameters for six commercial PEMFCs by minimizing the sum of squared errors (SSE) between experimental and predicted voltage data. The performance of CS-DE is benchmarked against several differential evolution (DE) variants adaptive differential evolution (PaDE), distance-based differential evolution (Di-DE), Lévy-flight success-history based adaptive differential evolution (LSHADE), natural-inspired differential evolution (NDE), palm-based differential evolution (PalmDE), particle swarm optimization differential evolution (PSO-DE), self-adaptive differential evolution with one population (jSO), learning population algorithm-based differential evolution (LPalmDE), and hierarchical archive-based differential evolution (HARD-DE) and is shown to perform better in terms of minimum, maximum, and mean SSE values, standard deviation, and runtime. CS-DE showed significant improvements in SSE, runtime and reliability with the lowest variance and standard deviation over different PEMFC cases. In addition, its runtime efficiency and stability indicate that CS-DE is also appropriate for real time, industrial applications. For all 12 PEMFC case studies, CS-DE showed the best consistency with the lowest standard deviation values. For example, in the STD-1 PEMFC case, CS-DE had a standard deviation of 2.04×10^{-16}, which is almost zero and hence the most stable algorithm among the compared variants. In each case, CS-DE achieved the fastest runtime, making it optimal for time sensitive applications. For the Horizon PEMFC case, CS-DE achieved the optimization task in 0.103 s, indicating its ability to deliver reliable results in a short time. In particular, this was significantly faster than other algorithms, with speedups of up to 98–99% in some cases. Across cases, CS-DE always had the lowest mean SSE, and was closest to experimental values. For example, in the BCS 500 W PEMFC case, CS-DE recorded a mean SSE of 0.0255, which is better than the alternatives such as PaDE with a mean SSE of 0.1288. For PEMFC applications, where accuracy in parameter estimation leads to optimal system performance, this precise error control is critical. CS-DE showed excellent alignment with experimental data, with an average relative error as low as 0.373% in some cases, thus validating its applicability in accurate PEMFC parameter modeling in real-world scenarios.

Enhancements in mutation strategies, adaptive control mechanisms, and a new population size reduction scheme in CS-DE are presented in the paper, which improve convergence and scalability for complex parameter estimation tasks. CS-DE performance, low computational requirements, and accuracy in estimating PEMFC parameters makes it a powerful tool in PEMFC modeling and optimization, and has potential for broader application in clean energy systems.

Author contribution Conceptualization—Pradeep Jangir, Arpita, Sunilkumar P. Agrawal, Sundaram B. Pandya, Anil Parmar, Sumit Kumar, Ghanshyam G. Tejani, Laith Abualigah; Formal analysis—Sundaram B. Pandya, Pradeep Jangir, Sunilkumar P. Agrawal, Laith Abualigah; Investigation—Sundaram B. Pandya, Laith Abualigah, Pradeep Jangir, Arpita; Methodology—Pradeep Jangir, Arpita, Sunilkumar P. Agrawal, Anil Parmar, Ghanshyam G. Tejani; Software—Anil Parmar, Sunilkumar P. Agrawal, Sundaram B. Pandya, Pradeep Jangir; Writing – original draft—Pradeep Jangir, Sundaram B. Pandya, Laith Abualigah; Writing – review & editing—Anil Parmar, Sunilkumar P. Agrawal, Sundaram B. Pandya, Pradeep Jangir, Laith Abualigah, Ghanshyam G. Tejani; All authors have read and agreed to the published version of the manuscript.

Data availability The data presented in this study are available through email upon request to the corresponding author.

Declarations

Competing interests The authors declare no competing interests.

References

1. Nikiforow K et al (2018) Power ramp rate capabilities of a 5 kW proton exchange membrane fuel cell system with discrete ejector control. J Power Sources 381(30–37):v. https://doi.org/10.1016/j.jpowsour.2018.01.090
2. Ahmed K et al (2020) Proton exchange membrane hydrogen fuel cell as the grid connected power generator. Energies 13(24):6679. https://doi.org/10.3390/en13246679

3. Ansari SA et al (2021) Modeling and simulation of a proton exchange membrane fuel cell alongside a waste heat recovery system based on the organic rankine cycle in MATLAB/SIMULINK environment. Sustainability 13(3):1218. https://doi.org/10.3390/su13031218
4. Derbeli M, Barambones O, Sbita L (2018) A robust maximum power point tracking control method for a PEM fuel cell power system. Appl Sci 8(12):2449. https://doi.org/10.3390/app8122449
5. Chavan SL, Talange DB (2024) Electrical equivalent circuit modeling and parameter estimation for PEM fuel cell. IEEE
6. Forrai A et al (2005) Fuel-cell parameter estimation and diagnostics. IEEE Trans Energy Convers 20(3):668–675. https://doi.org/10.1109/TEC.2005.845516
7. Mo ZJ et al (2006) Parameter optimization for a PEMFC model with a hybrid genetic algorithm. Int J Energy Res 30(8):585–597. https://doi.org/10.1002/er.1170
8. Outeiro M (2009) A new parameter extraction method for accurate modeling of PEM fuel cells. 33(11): 978–88. https://doi.org/10.1002/er.1525
9. Mansour D-EA et al (2020) Recent advances in polymer nanocomposites based on polyethylene and polyvinylchloride for power cables. Materials 14(1):66. https://doi.org/10.3390/ma14010066
10. Ali MN et al (2021) An efficient fuzzy-logic based variable-step incremental conductance MPPT method for grid-connected PV systems. Ieee Access 9:26420–26430. https://doi.org/10.1109/ACCESS.2021.3058052
11. Yang B, Zeng C, Wang L, Guo Y, Chen G, Guo Z et al (2021) Parameter identification of proton exchange membrane fuel cell via Levenberg-Marquardt backpropagation algorithm. Int J Hydrog Energy 46(44):22998–23012. https://doi.org/10.1016/j.ijhydene.2021.04.130
12. Sedighizadeh M et al (2011) Parameter optimization for a PEMFC model with particle swarm optimization. Int J Eng Appl Sci 3(1):102–108
13. Liu C-M et al (2015) Modified intake and exhaust system for piston-type compressed air engines. Energy 90:516–524. https://doi.org/10.1016/j.energy.2015.07.085
14. El-Fergany AA (2018) Electrical characterisation of proton exchange membrane fuel cells stack using grasshopper optimiser. IET Renew Power Gener 12(1):9–17. https://doi.org/10.1049/iet-rpg.2017.0232
15. Fathy A, Rezk H (2018) Multi-verse optimizer for identifying the optimal parameters of PEMFC model. Energy 143:634–644. https://doi.org/10.1016/j.energy.2017.11.014
16. Zhang W, Wang N, Yang S (2013) Hybrid artificial bee colony algorithm for parameter estimation of proton exchange membrane fuel cell. Int J Hydrog Energy 38(14):5796–5806. https://doi.org/10.1016/j.ijhydene.2013.01.058
17. Ohenoja M, Leiviska K (2010) Validation of genetic algorithm results in a fuel cell model. Int J Hydrogen Energy 35(22):12618–25. https://doi.org/10.1016/j.ijhydene.2010.07.129
18. Ye M, Wang X, Xu Y (2009) Parameter identification for proton exchange membrane fuel cell model using particle swarm optimization. Int J Hydrogen Energy 34(2):981–9. https://doi.org/10.1016/j.ijhydene.2008.11.026
19. Chakraborty UK, Abbott TE, Das SK (2012) PEM fuel cell modeling using differential evolution. Energy 40(1):387e99. https://doi.org/10.1016/j.energy.2012.01.039
20. Gong W, Cai Z (2013) Accelerating parameter identification of proton exchange membrane fuel cell model with ranking-based differential evolution. Energy 59:356–64. https://doi.org/10.1016/j.energy.2013.07.005
21. Dai C, Chen W, Cheng Z, Li Q, Jiang Z, Jia J (2011) Seeker optimization algorithm for global optimization: a case study on optimal modelling of proton exchange membrane fuel cell (PEMFC). Int J Electr Power Energy Syst 33(3):369–76. https://doi.org/10.1016/j.ijepes.2010.08.032
22. Yang S, Wang N (2012) A novel p systems based optimization algorithm for parameter estimation of proton exchange membrane fuel cell model. Int J Hydrogen Energy 37(10):8465–76. https://doi.org/10.1016/j.ijhydene.2012.02.131
23 Zhang L, Wang N (2013) An adaptive RNA genetic algorithm for modeling of proton exchange membrane fuel cells. Int J Hydrogen Energy 38(1):219–28. https://doi.org/10.1016/j.ijhydene.2012.10.026
24. Zhu Q, Wang N, Zhang L (2014) Circular genetic operators based RNA genetic algorithm for modeling proton exchange membrane fuel cells. Int J Hydrogen Energy 39(31):17779–90. https://doi.org/10.1016/j.ijhydene.2014.07.081
25. Meng Z, Zhong Y, Yang C (2021) CS-DE: cooperative strategy based differential evolution with population diversity enhancement. Inf Sci 577:663–696. https://doi.org/10.1016/j.ins.2021.07.080
26. Sultan HM, Menesy AS, Hassan M, Jurado F, Kamel S (2023) Standard and quasi oppositional bonobo optimizers for parameter extraction of PEM fuel cell stacks. Fuel 340:127586. https://doi.org/10.1016/j.fuel.2023.127586
27. Zhou H, Wu X, Li Y, Fan Z, Chen W, Mao J et al (2024) Model optimization of a high-power commercial PEMFC system via an improved grey wolf optimization method. Fuel 357:129589. https://doi.org/10.1016/j.fuel.2023.129589
28. Yongguang C, Guanglei Z (2022) New parameters identification of proton exchange membrane fuel cell stacks based on an improved version of African vulture optimization algorithm. Energy Rep 8(75):3030–3040. https://doi.org/10.1016/j.egyr.2022.02.066
29. Menesy SA, Sultan HM, Selim A, Ashmawy MG, Kamel S (2020) Developing and applying chaotic Harris Hawks Optimization technique for extracting parameters of several proton exchange membrane fuel cell stacks. IEEE Access 8:1146–59. https://doi.org/10.1109/ACCESS.2019.2961811
30 Meng Z, Pan J-S, Tseng K-K (2019) PaDE: An enhanced differential evolution algorithm with novel control parameter adaptation schemes for numerical optimization. Knowl-Based Syst 168:80–99. https://doi.org/10.1016/j.knosys.2019.01.006
31. Meng Z, Yang C, Li X, Chen Y (2020) Di-de: Depth information-based differential evolution with adaptive parameter control for numerical optimization. IEEE Access 8:40809–40827. https://doi.org/10.1109/ACCESS.2020.2976845
32. Tanabe R, A S (2014) Fukunaga, Improving the search performance of SHADE using linear population size reduction, in: 2014 IEEE Congress on Evolutionary Computation (CEC). IEEE 1658-1665. https://doi.org/10.1109/CEC.2014.690038
33. Tian M, Gao X (2019) Differential evolution with neighborhood-based adaptive evolution mechanism for numerical optimization. Inf Sci 478:422–448. https://doi.org/10.1016/j.ins.2018.11.021
34. Meng Z, Pan J-S, Kong L (2018) Parameters with adaptive learning mechanism (PALM) for the enhancement of differential evolution. Knowl-Based Syst 141:92–112. https://doi.org/10.1016/j.knosys.2017.11.015
35. Guo E, Gao Y, Chenyang Hu, Zhang J (2023) A hybrid PSO-DE intelligent algorithm for solving constrained optimization problems based on feasibility rules. Mathematics 11(3):522. https://doi.org/10.3390/math11030522
36. Brest J, Mauc˘ec MS, Boškovic' B (2017) Single objective real-parameter optimization: algorithm jSO, in: 2017 IEEE Congress on Evolutionary Computation (CEC). IEEE 1311-1318. https://doi.org/10.1109/CEC.2017.7969456
37. Meng Z, Pan J-S (2019) HARD-DE: Hierarchical ARchive based mutation strategy with depth information of evolution for the enhancement of differential evolution on numerical optimization.

IEEE Access 7:12832–12854. https://doi.org/10.1109/ACCESS.2019.2893292
38. Sun Z, Wang N, Bi Y, Srinivasan D (2015) Parameter identification of PEMFC model based on hybrid adaptive differential evolution algorithm. Energy 90:1334–1341. https://doi.org/10.1016/j.energy.2015.06.081
39. Correa JM, Farret FA, Canha LN, Simoes MG (2004) An electrochemical-based fuelcell model suitable for electrical engineering automation approach. IEEE Trans Ind Electron 51(5):1103–12. https://doi.org/10.1109/TIE.2004.834972
40. Amphlett JC, Baumert R, Mann RF, Peppley BA, Roberge PR, Harris TJ (1995) Performance modeling of the ballard mark iv solid polymer electrolyte fuel cell. J Electrochem Soc 142(1):1–8. https://doi.org/10.1149/1.2043866
41. Mann RF, Amphlett JC, Hooper MA, Jensen HM, Peppley BA, Roberge PR (2000) Development and application of a generalised steady-state electrochemical model for a PEM fuel cell. J Power Sources 86(1):173–80. https://doi.org/10.1016/S0378-7753(99)00484-X
42. Mo Z-J, Zhu X-J, Wei L-Y, Cao G-Y (2006) Parameter optimization for a PEMFC model with a hybrid genetic algorithm. Int J Energy Res 30(8):585–97. https://doi.org/10.1002/er.1170
43. Larminie J, Dicks A, McDonald MS (2003) Fuel cell systems explained, vol. 2. New York: Wiley. https://doi.org/10.1002/9781118878330
44. Meng Z, Yang C (2021) Hip-DE: Historical population based mutation strategy in differential evolution with parameter adaptive mechanism. Inf Sci 562:44–77. https://doi.org/10.1016/j.ins.2021.01.031
45. Poláková R, Tvrdík J, Bujok P (2019) Differential evolution with adaptive mechanism of population size according to current population diversity. Swarm Evol Comput. https://doi.org/10.1016/j.swevo.2019.03.014
46. Xuebin L, Zhao J, Daiwei Y, Jun Z, Wenjin Z (2024) Parameter estimation of PEM fuel cells using metaheuristic algorithms. Measurement 237:115302