



OPEN A hybrid slime mold enhanced convergent particle swarm optimizer for parameter estimation of proton exchange membrane fuel cell

Mohammad Aljaidi¹✉, Sunilkumar P. Agrawal², Anil Parmar³, Pradeep Jangir^{4,5,6}, Arpita⁷, Bhargavi Indrajit Trivedi⁸, G. Gulothungan⁹✉, Ali Fayez Alkoradees¹⁰✉, Reena Jangid^{11,12,13} & Mohammad Khishe^{14,15}✉

High efficiency and eco friendliness, proton exchange membrane fuel cells (PEMFCs) have become a good solution to cleaner energy solutions. However, due to the electrochemical complexity of PEMFCs and the limitations of existing optimization methods, accurately estimating PEMFC parameters to achieve optimal performance is still challenging. In this work, we propose a hybrid optimization algorithm, SCPSO, combining Particle Swarm Optimization with Mixed Mutant Slime Mold to improve precision, consistency, and computational efficiency in PEMFC parameter optimization. Six PEMFC types, BCS 500 W, Nedstack 600 W PS6, SR-12 W, Horizon H-12, Ballard Mark V, and STD 250 W Stack were applied to SCPSO and compared with seven state-of-the-art algorithms, FLA, HFPSO, PSOLC, ESMA, LSMA, DETDO, and EGJO. In all cases, SCPSO consistently outperformed all competitors with the lowest mean sum of squared error (SSE) and minimal standard deviation (e.g., $[10^{-16}, 10^{-18}]$), thus confirming its robustness and reliability. Additionally, it demonstrated the lowest number of iterations to reach the optimal solution (less than 200 iterations) and best Friedman Rank (FR = 1), signifying the best optimization to the customer. For instance, in PEMFC1, SCPSO achieved minimal SSE of 0.02549 with negligible variability (Std. = $1.05958\text{E}-15$) as compared to HFPSO (Std. = 0.001998568) and DETDO (FR = 4). SCPSO's rapid convergence curves, narrow box plot spreads, and precise polarization curves were further validated across all fuel cells. SCPSO was experimentally validated and proved to be reliable with minimal deviations between predicted and experimental voltage and power outputs (e.g., RE = 0.052587% for PEMFC1 and RE = 0.016537% for PEMFC2). The average runtime of SCPSO was 3.05 s, which is faster than alternatives, and still maintains its unparalleled precision. The results of the analyses, fitting the datasets and the convergence curves confirm that the adaptive parameter tuning of SCPSO has significantly improved its performance, resulting in the highest consistency and accuracy with the fastest convergence speed. For PEMFC parameter optimization, results from SCPSO have established it as the algorithm with the strongest precision and stability and fastest computational efficiency. The extension to other energy systems and dynamic real time scenarios will be investigated in future research to enable wider adoption in sustainable energy management.

Keywords Proton exchange membrane fuel cells (PEMFCs), Parameter optimization, SCPSO algorithm, Mixed mutant slime mold, Metaheuristic optimization

¹Department of Computer Science, Faculty of Information Technology, Zarqa University, Zarqa 13110, Jordan.

²Department of Electrical Engineering, Government Engineering College, Gandhinagar, Gujarat 382028, India.

³Department of Electrical Engineering, Shri K.J. Polytechnic, Bharuch 392 001, India. ⁴Innovation Center for Artificial Intelligence Applications, Yuan Ze University, Taoyuan 320315, Taiwan. ⁵University Centre for Research and Development, Chandigarh University, Mohali 140413, India. ⁶Applied Science Research Center, Applied Science Private University, Amman 11931, Jordan. ⁷Department of Biosciences, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai 602 105, India. ⁸Department of CSE, Graphic Era Hill University, Dehradun 248002, India. ⁹Department of CSE, Graphic Era Deemed to be University, Dehradun, Uttarakhand 248002, India. ¹⁰Centre for Research Impact and Outcome, Chitkara University Institute of Engineering

and Technology, Chitkara University, Rajpura, Punjab 140401, India. ¹¹Vishwakarma Government Engineering College, Ahmedabad, Gujarat, India. ¹²Department of Electronics and Communication Engineering, SRM Institute of Science and Technology, SRM Nagar, Kattankulathur, Chengalpattu, Tamilnadu 603203, India. ¹³Unit of Scientific Research, Applied College, Qassim University, Buraydah, Saudi Arabia. ¹⁴Department of Electrical Engineering, Imam Khomeini Naval Science University of Nowshahr, Nowshahr, Iran. ¹⁵Jadara University Research Center, Jadara University, Irbid, Jordan. ✉email: mjaidi@zu.edu.jo; g.gulothungan@gmail.com; alifk@qu.edu.sa; m_khishe@alumni.iust.ac.ir

Fossil fuels and traditional energy sources use have drastically increased with such pace in their reliance that has increased greenhouse emissions and further deteriorated the environmental issues. However, these effects do more than just threaten ecosystems, human health, and societal sustainability. Therefore, renewable energy sources (RESs) are considered as important alternatives for dealing with these challenges. Fuel cells (FCs) are among the promising RESs for the generation of heat and electricity with minimal environmental impact. In particular, Proton Exchange Membrane Fuel Cells (PEMFCs) are of interest because of their simple structure, high energy density, and adaptability to applications such as residential power, transportation, and energy storage.

Although promising, accurately modeling and optimizing PEMFCs is a difficult task. Due to the nonlinear and complex operational nature of PEMFCs, accurate mathematical models are required to predict their performance under different conditions. Empirical and quasi empirical models have successfully simulated PEMFC behaviors, but their reliability is constrained to the accuracy of parameter identification. These parameters are often unknown, and non-linear, and determining them is critical for building effective equivalent circuit models that accurately replicate real world PEMFC performance.

Previous research work

A great deal of research effort has been expended in the attempt to develop robust PEMFC parameter identification methodologies. Notable contributions include: To extract optimal parameters for PEMFC systems, Abd Elaziz et al. proposed the Gorilla Troops Optimizer (GTO), which demonstrated faster convergence rates and accuracy, but the method required specific parameter settings to be effective¹. Fuzzy logic based reasoning techniques for PEMFC control were explored by Agila et al.² that provide adaptability, but no detailed comparison with other advanced optimization methods is presented. Grasshopper Optimization Algorithms for PEMFC parameter optimization were introduced by Ai et al.³, but computational efficiency was highlighted while scalability was not tested. Finite element analysis was combined with neural networks to optimize PEMFC performance, achieving significant efficiency improvements, but the approach was complex⁴. To reduce computational burdens in PEMFC modeling, Ali et al. proposed an adaptive estimation method that provides accurate results, but with limited validation against real world data⁵. Altun et al. combined evolutionary strategies with a hybrid approach for PEMFC loss minimization and demonstrated promising results, but did not test robustness under different conditions⁶. Optimization strategies for Solid Oxide Fuel Cells (SOFCs) were reviewed by Anuar et al.⁷, comparing their high temperature application advantages with PEMFC limitations, but without focusing on PEMFC specific optimization. Awad and El-Desouky used hybrid algorithms to improve PEMFC performance by focusing on convergence speed and with limited application to various case studies⁸. The study lacked focus on optimization techniques, but Barbir noted technological advances in PEMFC systems in the area of energy density and durability⁹. In¹⁰, Chen et al. extracted PEMFC parameters using Differential Evolution algorithms with high precision, but without comparison with other algorithms.

Chen et al. highlighted challenges in computational modeling for PEMFCs, and proposed solutions to balance accuracy and efficiency, but without practical validation¹¹. Hybrid algorithms for fuel cell optimization were introduced by Cheng et al., which achieved robust parameter tuning at the cost of high computational cost¹². Adaptive learning techniques for PEMFC modeling were proposed by Ding et al.¹³ to reduce prediction errors, but the operational adaptability is limited. Metaheuristic global optimization methods for PEMFC design were reviewed by El-Bashir et al.¹⁴ and their potential was recognized, however, detailed implementation examples were not given. Deep reinforcement learning was applied by Fan et al. to PEMFC operational parameter optimization, which resulted in increased efficiency but limited by computational resources¹⁵. Multi objective evolutionary algorithms were used by Feng et al. to optimize PEMFC operating conditions to balance efficiency and durability, but without scalability tests¹⁶. Intelligent algorithms were used by Gao et al. to improve PEMFC stack performance, with gains in energy efficiency, but without experimental validation¹⁷. Analytical models for hybrid applications with PEMFCs were developed by Ghaffar et al., to address integration challenges, but with limited modeling of complex scenarios¹⁸. A comparative study of metaheuristics for PEMFC optimization was performed by Guo et al., who pointed out the strengths of the algorithms, but did not include real world case studies¹⁹. In²⁰, Hafez et al. used nature inspired algorithms to enhance PEMFC performance, but without consideration of computational efficiency.

Hybrid models for PEMFC design were evaluated by Hamid et al.²¹ using deterministic and stochastic methods combined effectively, but the study was limited by the availability of limited datasets. Deep learning methods for multi-variable optimization of PEMFCs were introduced by He et al.²² with high accuracy but unexplored scalability. In²³, Hu et al. used genetic algorithms for PEMFC parameter optimization, with nonlinear constraints but without thorough comparative analysis. Machine learning was combined with hybrid strategies to tackle PEMFC trade-offs, however, operational tests were not performed²⁴. Adaptive neuro fuzzy systems were used by Jang et al. for PEMFC optimization²⁵ and showed adaptability but did not include cost analysis. However, the validation of hybrid swarm algorithms across different system scales was missing, which Jiang et al. applied to balance efficiency and computational time in PEMFCs²⁶. Multi objective optimization strategies for PEMFC stacks, including robustness metrics but not scalability issues, were proposed by Jin et al.²⁷. Hybrid

machine learning models for PEMFC efficiency were used by Karim et al.²⁸ to improve accuracy but only offer minimal sensitivity analysis. Various optimization algorithms for PEMFC systems were compared by Khan et al.²⁹, who focused on algorithm refinements for better scalability. Reinforcement learning was applied to PEMFC operational parameters by Kim et al.³⁰, achieving adaptability with high computational costs.

Deep learning approaches for PEMFC diagnostics were developed by Kumar et al.³¹, reducing errors, but they lack robustness in real time applications. Machine learning frameworks to optimize PEMFC performance were introduced by Lei et al.³² but without practical validation, which showed gains in accuracy. Algorithmic improvements for PEMFC optimization were proposed by Li et al.³³, which converge better without exploring multiobjective optimization. AI enhanced methodologies for PEMFC optimization were used by Liang et al.³⁴ to enhance efficiency without practical validation. Liu et al. integrated neural networks into PEMFC modeling, obtaining predictive accuracy but without adaptability to dynamic conditions³⁵. Hybrid swarm algorithms were applied to PEMFC design by Liu et al.³⁶, but efficiency was improved without testing scalability. Loss reduction in PEMFCs using optimization techniques was focused on by Ma et al.³⁷, but results were positive and without comparative analysis. Evolutionary algorithms for enhancing PEMFC efficiency were proposed by Malik et al.³⁸, which have high computational cost but stability. Deep learning models developed by Mao et al.³⁹ were applied to optimize PEMFCs with high accuracy, but at a high complexity. Hybrid optimization strategies for PEMFCs were introduced by Meng et al.⁴⁰, which attempt to balance efficiency and durability, but the validation was limited.

AI models were applied to PEMFC design by Nair et al. to improve energy management, but the work was based on many theoretical assumptions⁴¹. Advanced optimization algorithms for PEMFC performance are analyzed by Nawaz et al.⁴², which yielded robustness but no experimental case studies. Heuristic approaches for multi-objective optimization were introduced by Nguyen et al.⁴³ that increased cost effectiveness but with only limited scalability tests. Reinforcement learning was used by Pan et al. to improve PEMFC efficiency, which is flexible but computationally intensive⁴⁴. Neuro fuzzy systems were used by Park et al. for PEMFC modeling, which are adaptable but are not validated in the real world⁴⁵. Genetic algorithms are applied by Qi et al. for PEMFC optimization with improved performance but without comparison with other methods⁴⁶. In⁴⁷, Qian et al. developed AI driven models for PEMFC efficiency, which showed robust predictions but low scalability. Hybrid optimization techniques were used by Riaz et al. for PEMFCs, reducing computational time but with a small dataset⁴⁸. Bayesian models for PEMFC optimization were explored by Rui et al.⁴⁹, which improved accuracy but restricted their applicability. Multi-objective optimization strategies for fuel cells were introduced by Singh et al.⁵⁰ which were robust but not validated under dynamic conditions.

Evolutionary algorithms were applied by Tang et al. for PEMFC stack design that yielded improvements in energy efficiency and operational stability. The work showed the promise of evolutionary algorithms to optimize stack performance, but was restricted in its applicability to more general operational conditions and large scale systems⁵¹. Wang et al. used hybrid machine learning techniques to optimize PEMFC efficiency, which they point out can provide a balance of accuracy and robustness. The study, though, was not able to provide insights into the computational overhead of these machine learning models⁵². Wang et al. also studied the application of deep learning algorithms for PEMFC optimization, demonstrating their capability of improving system efficiency and parameter estimation precision. However, it was noted as a limitation that these algorithms have a high computational cost⁵³.

AI driven optimization frameworks for PEMFC systems were introduced by Wu et al. to enhance energy efficiency and operational adaptability. The study revealed the versatility of AI models, but did not have experimental validation to prove their real world effectiveness⁵⁴. To improve performance and reliability, Xie et al. proposed the integration of advanced AI techniques into PEMFC design. However, they did not address how their model would adapt to dynamic environmental conditions, thereby limiting broader applicability⁵⁵ and they were able to achieve improved parameter estimation accuracy. Hybrid algorithms for multi objective optimization in PEMFC systems were presented by Xu et al. that effectively trade off efficiency and durability. However, these algorithms are computationally demanding and it became a challenge to scale them⁵⁶.

A comparison of the AI models used was made by Yang et al. for their contribution to PEMFC optimization, and they demonstrated great improvements to the energy efficiency and robustness. The study provided valuable insights, but testing in dynamic and real world conditions was not performed, which limited its practical applicability⁵⁷. To estimate PEMFC parameters, Zhang et al. used advanced deep learning models for PEMFC design optimization with high accuracy. However, the study did not provide a detailed comparison with other optimization frameworks⁵⁸ and showed the potential of AI to enhance the PEMFC performance. Zhang et al. also investigated hybrid models for PEMFC systems to optimize the PEMFC system, and showed improvements in both operational stability and energy efficiency. The study, however, was limited in scalability to diverse conditions⁵⁹. Lastly, Zhou et al. proposed novel multi objective optimization approaches for PEMFC systems with the aim of improving their operational flexibility and efficiency. The approach was promising but its overall impact was reduced because of the absence of a thorough comparison with state of the art methods⁶⁰.

Research gap

Literature of recent work of last half decade shows a consistent focus on improving PEMFC efficiency through optimization algorithms but gaps exist in scalability, adaptability to dynamic conditions, computational efficiency and real world validation. However, most existing studies focus on single objective or limited multi objective optimization, without addressing the complex tradeoffs between cost, durability and performance in a comprehensive way. The importance of this is that a novel, hybrid algorithm that combines the best of advanced AI and metaheuristic strategies is needed to provide scalable, efficient, and robust solutions to PEMFC optimization problems. The main challenge in PEMFC modeling is to determine the optimal values of critical parameters so that the estimated performance becomes consistent with experimental data. However, current

optimization methods are often computationally expensive and variable, rendering them unsuitable for real time applications or large scale systems. As a result, there is a strong need to develop a robust optimization algorithm that guarantees accuracy and consistency while minimizing computational overhead.

The main difficulty in PEMFC modeling is finding the optimal values of critical parameters to make the estimated performance converge with the experimental data. However, current optimization methods are computationally expensive and variable, and are thus not suitable for real time applications or large scale systems. Therefore, there is a compelling need to develop a robust optimization algorithm that is both accurate and consistent, and that minimizes computational overhead.

Contributions of this work

To address these challenges, this paper presents a novel hybrid optimization algorithm, a particle swarm optimization for mixed mutant slime mold (SCPSO)⁶¹, specifically designed for PEMFC parameter optimization. The paper introduces SCPSO as a hybrid optimization method that combines PSO with Mixed Mutant Slime Mold to optimize PEMFC parameters while improving precision and computational speed and reliability. The paper demonstrates SCPSO's importance through an evaluation of different optimization algorithms from the introduction section that shows their strengths and weaknesses. Gorilla Troops Optimizer (GTO) demonstrates quick parameter extraction accuracy for PEMFC systems yet requires fixed parameter configurations which reduces its flexibility for different system configurations. The adaptability of fuzzy logic methods in dynamic control systems exists without any state-of-the-art optimization method performance comparisons. The computational efficiency of the Grasshopper Optimization Algorithm exists alongside unproven testing of its scalability potential for bigger PEMFC systems. Real-time application feasibility may suffer from the combined approach of Finite Element Analysis together with Neural Networks because the method introduces complex operational requirements. The Adaptive Estimation Method delivers accurate PEMFC models at low computational expense yet its implementation depends heavily on real-world testing because real-world testing affects practical operation and safety. Hybrid approaches that utilize evolutionary strategies present effective loss reduction performance but need further assessment of their operational condition adaptability. The precision of Differential Evolution parameter extraction remains high but its validation becomes challenging because of lacking comparative studies with alternative methods. The computation-intensive nature of hybrid algorithms as parameter optimizers restricts their capacity for real-time PEMFC system operation even when they enhance the robustness of parameter settings. The implementation of adaptive learning strategies in PEMFC modeling leads to major reduction of prediction errors but they demonstrate limited capability for dynamic operating conditions. The potential of metaheuristic methods for PEMFC design remains unclear because their detailed implementation remains unclear which prevents their practical use. The enhanced efficiency from deep reinforcement learning requires huge computational resources making its scalable application restricted. The implementation of multi-objective evolutionary algorithms for PEMFC operation finds balance between efficiency and durability but needs extensive scaling tests before application in larger systems. The contribution of intelligent algorithms to stack performance improvement and energy efficiency remains unproven experimentally which reduces their reliability when used in real-world applications. The analytical models that address hybrid application integration issues show restricted application in complex scenarios because of their basic modeling capabilities. The comparison of metaheuristic algorithms demonstrates distinct features between optimization techniques although it does not contain actual industry applications for practical use. The implementation of PEMFC technology through nature-inspired algorithms utilizing biologically based methodologies requires more thorough evaluation of computational speed to ensure practical implementation success. The combination of deterministic and stochastic approaches in modeling helps improve PEMFC design only through constrained access to datasets. The application scope of deep learning optimizations for PEMFCs may be limited in large-scale systems because high-accuracy multi-variable optimization is achieved while scalability tests have not been conducted. The nonlinear constraints which genetic algorithms optimize in PEMFC parameter settings need further evaluation against other optimization algorithms. Machine learning models working with hybrid optimization strategies succeed at solving trade-off difficulties but need operational assessments for real-world validation. The adaptive neuro-fuzzy systems provide PEMFC optimization with adaptable features yet they lack cost evaluation which reduces their practical implementation potential. Hybrid swarm algorithms both maximize system speed and performance yet have not been tested for various-scale system implementations. The addition of robustness metrics through stack optimization under multi-objective optimization techniques does not resolve scalability issues when working with larger systems. The accuracy of PEMFC efficiency optimization with mixed machine learning models is boosted but these methods need improvement in sensitivity analysis for reliability enhancement. The operational parameter optimization benefits from reinforcement learning although its high computational expenses limit its practical use. The use of deep learning for PEMFC diagnostics decreases diagnostic errors but the solutions prove inadequate for real-time usage. Means to enhance PEMFC performance through machine learning frameworks need extra practical testing before final acceptance. Better PEMFC optimization results from algorithmic advancements in convergence capabilities although these changes do not develop or explore multi-objective optimization potential. The effectiveness of AI-enhanced methodologies grows through efficiency improvements yet their experimental verification is nonexistent which holds negative impacts on reliability. The predictive power of PEMFC modeling improves through neural networks despite the need to investigate their capacity to adjust when dynamic conditions occur. The use of hybrid swarm algorithms enhances PEMFC design efficiency but developers need to verify system scalability aspects. The positive results from loss reduction optimization methods exist without established comparisons to alternative solutions. The computational stability and efficiency of evolutionary algorithms might be limited by high computational needs that affect their practicality potential. The implementation of deep learning models improves both system operating efficiency and parameter estimation accuracy at the cost of increased computational complexity. Hybrid

optimization strategies bring together an optimal balance between operational life and performance efficiency while their limited validation requirements reduce reliability measures. The AI-based energy management techniques improve power usage through theoretical framework assumptions which restrict their effectiveness in actual practice. The effectiveness of advanced optimization algorithms in performance optimization remains unknown because experimental case studies have not validated their results. The heuristic optimization methods reduce PEMFC optimization costs while needing further scalability testing. The operational adaptability of AI-driven optimization frameworks improves through their frameworks yet does not have tested validation in real-world applications. AI applications deliver superior parameter estimation results yet they do not solve the issue of PEMFC performance change during environmental system fluctuations. Hybrid systems made from deterministic and stochastic methods work effectively for PEMFC designs despite existing limitations in their usable datasets. The extensive review of these optimization methods reveals both their positive and negative aspects for PEMFC optimization. The SCPSO algorithm uses PSO and Mixed Mutant Slime Mold benefits to overcome optimization challenges while improving accuracy and efficiency and achieving consistent results. The computational and adaptability challenges of existing methods find improvement through SCPSO which demonstrates potential as a solution for PEMFC parameter optimization.

The SCPSO algorithm reaches exceptional performance because it combines Particle Swarm Optimization (PSO) and Mixed Mutant Slime Mold Algorithm (SMA) features into a single hybrid structure. The combined structure produces superior exploration and exploitation capabilities that create a balanced optimization process. The combination of PSO global capability with SMA efficient localized search in SCPSO allows the method to avoid premature convergence and find the global optimum efficiently. The algorithm combines two mechanisms which allow it to search multiple areas of the optimization space while building a robust path toward finding optimal solutions. The improved population initialization strategy serves as a main efficiency factor for SCPSO. The algorithm uses Good Point Set methodology to achieve uniform distribution of its initial solutions. The strategy creates more diverse search space conditions that enable complete exploration of available solutions. The risk of suboptimal initial distributions becomes minimized through SCPSO which leads to better chances of discovering superior parameter values. The system implements adaptive parameter adjustments through which both control weights and learning constants automatically modify their values according to optimization status. By being adaptable the algorithm preserves a balanced relationship between exploration and exploitation from start to end of the search procedure resulting in rapid and precise convergence. The SCPSO algorithm includes a mutation mechanism which draws its concept from SMA. The structured randomness function adds unpredictable variability to the search operations which helps the system escape current local optima while exploring new search areas which normal optimization methods would overlook. SCPSO implements a sigmoid function for nonlinear scaling that expands the search area and enhances its ability to solve complex optimization problems with multiple optimal solutions. The optimization process becomes more efficient through dynamic adjustments of step size and search direction in SCPSO. The combination of reliability and stable performance of SCPSO sets it apart from competing algorithms. The hybrid framework within this method produces consistent outcomes across multiple runs because standard deviation measurements for SSE values remain low. This makes SCPSO an excellent choice for practical usage because it delivers dependable solutions in critical scenarios requiring precise results. The receipt of specialized features enables the algorithm to surpass current optimization techniques both in terms of precision accuracy and fast convergence speed and overall stability. The SCPSO approach proves itself as an optimal and efficient method for PEMFC parameter optimization by addressing the issues found in traditional metaheuristic techniques.

The key contributions of this research are:

- **Proposed Algorithm:** SCPSO integrates the exploration capabilities of Particle Swarm Optimization (PSO) with the exploitation features of Mixed Mutant Slime Mold to enhance convergence speed and solution accuracy.
- **Extensive Comparisons:** The algorithm is benchmarked against seven state-of-the-art algorithms, including Fick's Law Algorithm (FLA)⁶², Hybrid Firefly and PSO (HFPPO)⁶³, Particle Swarm Optimization with Learning Strategy (PSOLC)⁶⁴, Equilibrium Slime Mold Algorithm (ESMA)⁶⁵, and LSMA⁶⁶, DETDO⁶⁷, and EGJO⁶⁸.
- **Comprehensive Evaluation:** The proposed approach is applied to six commercial PEMFCs (BCS 500 W^{69,70}, SR-12 500 W^{69,70}, STD 250 W^{69,70}, Nedstack 600 W PS6⁷¹, Horizon H-12⁷², and Ballard Mark V⁷²) to validate its effectiveness in minimizing SSE, with comparisons based on statistical metrics such as Mean SSE, Standard Deviation (STD), and computational runtime.
- **Real-World Validation:** Experimental datasets are used to evaluate the algorithm's ability to predict polarization curves, voltage outputs, and power generation with high precision.
- **Statistical Robustness:** Multiple statistical tests, including Friedman Ranking and Wilcoxon Signed-Rank tests, are conducted to confirm the superiority and consistency of SCPSO over competing methods.

The novelty of this work is to hybridize PSO and Slime Mold principles to construct a balanced framework for robust PEMFC parameter optimization. The SCPSO algorithm offers a dual advantage: There was an enhanced computational efficiency and superior accuracy. It is a reliable tool for real world applications due to its ability to maintain low SSE with minimal variability across multiple runs. PEMFCs represent an advanced technology for clean energy conversion because they deliver high efficiency with minimal environmental consequences. PEMFC models rely on accurate parameter estimation because it determines their operational performance and reliability characteristics. Modern meta-heuristic algorithms together with statistical procedures have greatly optimized the processes which extract parameters. The authors Kanouni and Laib⁷³ developed an enhanced Differential Evolution algorithm which successfully optimized PEMFC performance by accurately extracting model parameters. Saad et al.⁷⁴ developed a parameter estimation technique that employs the Huber loss

statistical function for enhancing estimation accuracy during noise disturbances. Jangir and his co-authors created a PEMFC parameter estimation technique based on Differential Evolution with depth information enhancement to reach high precision results⁷⁵. Computational methods play a vital role in advancing PEMFC technology because various studies show their critical value for innovation in this field.

The remainder of this paper is structured as follows: In Section "PEMFC mathematical modelling", PEMFC mathematical modeling and the optimization problem formulation are presented. In Section "Particle swarm optimization algorithm for hybrid mutant slime mold", the design and operational principles of the SCPSO algorithm are detailed. The experimental setup, simulation results, and comparative analyses are presented in Section "Result analysis and discussion". Section "Conclusions" concludes with a key findings and potential future directions.

PEMFC mathematical modelling

Basic concept of PEMFC

The Proton Exchange Membrane Fuel Cell (PEMFC) structure includes two electrodes, specifically the anode and the cathode, and a proton-conducting membrane positioned between these electrodes as the polymer electrolyte. The schematic diagram of fuel cell is given in Fig. 1.

This arrangement permits the passage of protons while restricting electron flow⁷⁶. Additionally, catalyst layers are placed between the electrolyte membrane and both electrodes to expedite the chemical reaction. Hydrogen gas is supplied to the anode electrode, where, upon reaching the catalytic layer, it dissociates into electrons and protons. The protons then migrate through the electrolyte membrane to the catalytic layer at the cathode electrode, while the electrons are conducted through an external load. Oxygen or air is supplied to the cathode, and upon arrival at the catalytic layer of the cathode electrode, it combines with the protons from the membrane and the electrons from the external circuit to produce water. The electrochemical reactions at the PEMFC electrodes are expressed as follows⁷⁶:

Anode reaction

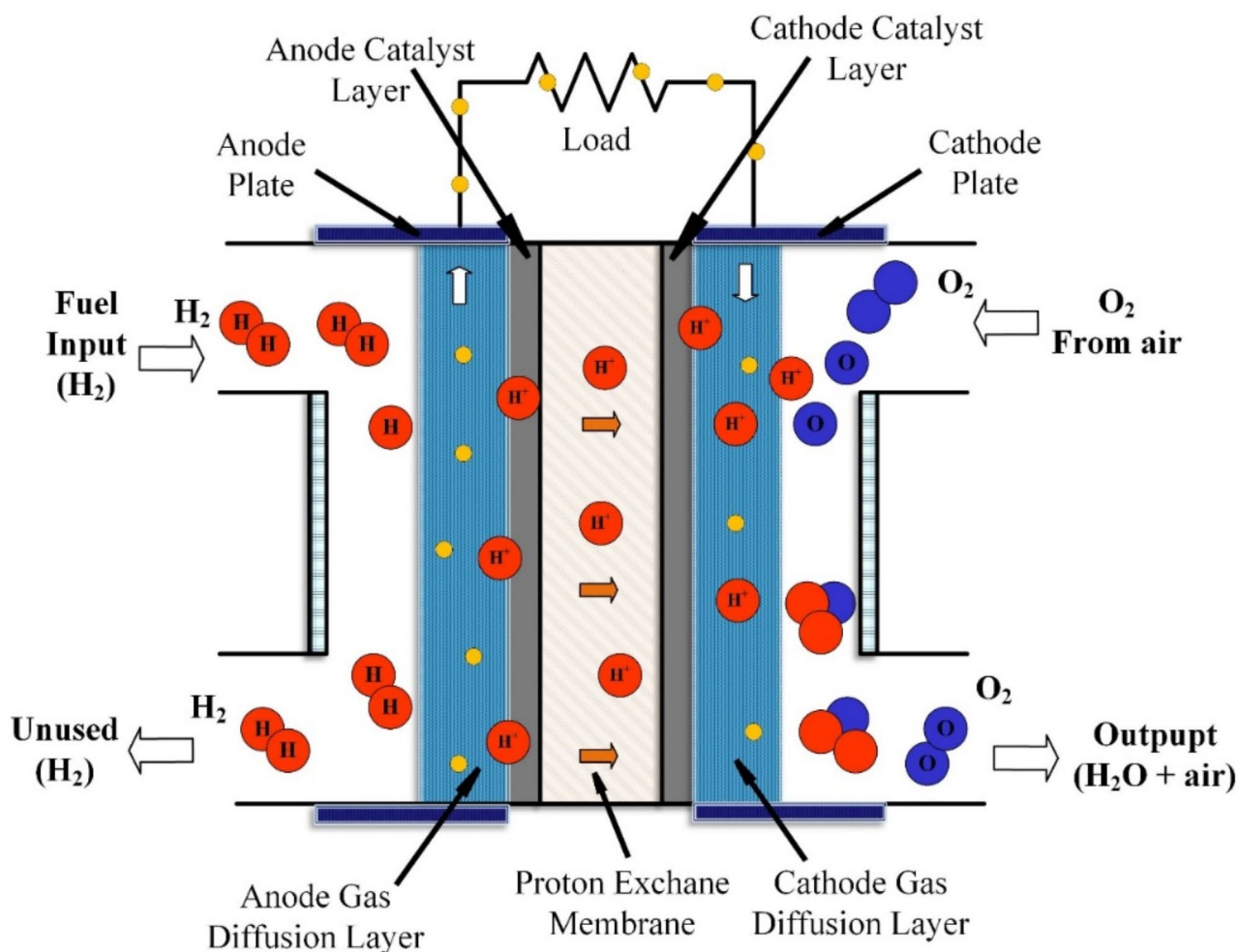


Fig. 1. Schematic of fuel cell.



Cathode reaction



Overall reaction:



In Eq. (3), the term “Energy” represents the electrical energy generated as a result of electron flow from hydrogen gas traveling from the anode to the cathode through an external load. The equivalent electrical circuit for PEMFC stack is shown in Fig. 2.

Mathematical model of PEMFC stacks

The output voltage V_{cell} of each individual fuel cell can be computed using the following expression^{77,78}:

$$V_{\text{cell}} = E_{\text{nerst}} - \Delta V_{\text{act}} - \Delta V_{\text{ohm}} - \Delta V_{\text{con}} \quad (4)$$

In this equation, E_{nerst} denotes the open-circuit voltage of the cell, ΔV_{act} represents the activation overpotential per cell, ΔV_{ohm} describes the voltage drop caused by ohmic resistance due to electron conduction through the external load and the proton movement resistance in the electrolyte membrane, and ΔV_{con} indicates the concentration overpotential per cell. Amphlett et al.⁷⁹ proposed a model of a fuel cell's electrochemical properties. When a series connection of N_{cells} identical fuel cells is configured for increased voltage output, the total stack voltage can be determined as:

$$V_{\text{stack}} = N_{\text{cells}} \cdot V_{\text{cell}} \quad (5)$$

Here, N_{cells} refers to the number of cells connected in series, and V_{cell} is the output voltage for each individual fuel cell, as derived from Eq. (4).

The reversible potential, E_{nerst} , is calculated as follows^{80,81}:

$$E_{\text{nerst}} = 1.229 - 8.5 \times 10^{-4}(T_{\text{fc}} - 298.15) + 4.3085 \times 10^{-5}T_{\text{fc}} \cdot [\ln(P_{H_2}) + \ln(\sqrt{P_{O_2}})] \quad (6)$$

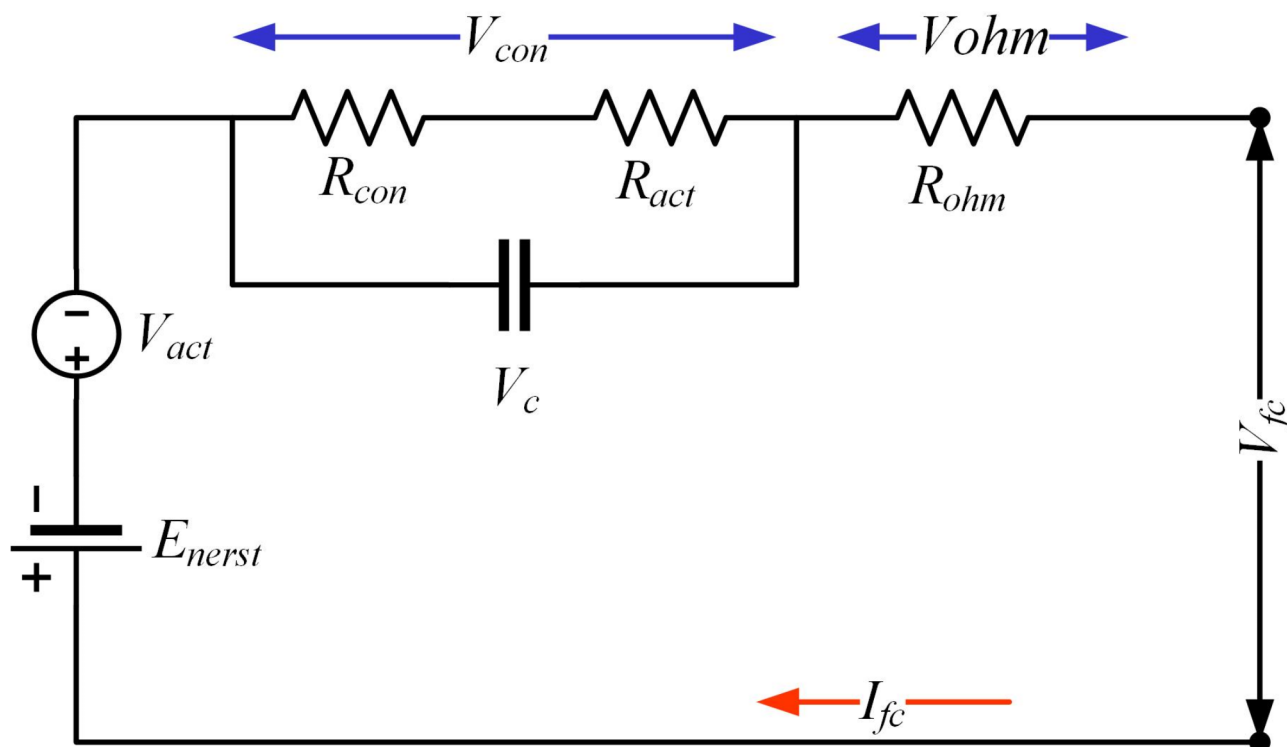


Fig. 2. Equivalent electrical circuit for PEMFC.

where T_{fc} is the cell's absolute operating temperature in Kelvin, while P_{H_2} and P_{O_2} denote the partial pressures of hydrogen and oxygen in the fuel cell stack's input channels (atm). When hydrogen and air serve as the inputs, the partial oxygen pressure, P_{O_2} , is determined as follows^{82,83}:

$$P_{O_2} = P_c - RH_c P_{H_2O}^{sat} - \frac{0.79}{0.21} P_{O_2} \cdot \exp \left(0.291 \frac{I_{fc}}{A} / T_{fc}^{0.832} \right) \quad (7)$$

where P_c represents the inlet channel pressure at the cathode (atm), RH_c is the cathode electrode's relative humidity, I_{fc} is the operating current (A), A is the membrane surface area (cm²), and $P_{H_2O}^{sat}$ is the water vapor pressure at saturation, defined by⁸⁴:

$$\log_{10}(P_{H_2O}^{sat}) = 2.95 \times 10^{-2}(T_{fc} - 273.15) - 9.18 \times 10^{-5}(T_{fc} - 273.15)^2 + 1.44 \times 10^{-7}(T_{fc} - 273.15)^3 - 2.18 \quad (8)$$

In cases where hydrogen and pure oxygen are used, the partial oxygen pressure P_{O_2} is calculated as follows⁸⁴:

$$P_{O_2} = RH_c P_{H_2O}^{sat} \left[\left(\exp \left(4.192 \frac{1}{I_{fc}} / T_{fc}^{1.334} \right) \cdot \frac{RH_a P_{H_2O}^{sat}}{P_a} \right)^{-1} - 1 \right] \quad (9)$$

Equation (7) describes the reaction process at the cathode when air supply is used because it includes nitrogen which makes up most of air composition. The operating conditions of PEMFC systems that use air as their oxidant find accurate representation through Eq. (7). Equation (9) becomes applicable for PEMFC operations when pure oxygen is fed to the cathode. Specialized applications require pure oxygen as an oxidant because they need either enhanced performance or unique experimental settings. PEMFC applications require a specific equation based on the choice of oxidant between air or pure oxygen. The use of air as an oxidant requires the implementation of Eq. (7) because nitrogen affects the partial oxygen pressure significantly. The use of pure oxygen requires Eq. (9) as the preferred calculation method because nitrogen is absent from pure oxygen environments. The PEMFC models studied in this work mainly use Eq. (7) because they operate with air as their oxidant. The equation successfully represents nitrogen effects to model partial oxygen pressure accurately for standard PEMFC operational simulations. The application of Eq. (9) requires pure oxygen systems but the present research models do not use pure oxygen.

In both cases, the partial hydrogen pressure P_{H_2} is given by:

$$P_{H_2} = 0.5 RH_a P_{H_2O}^{sat} \left[\left(\exp \left(1.635 \frac{1}{I_{fc}} / T_{fc}^{1.334} \right) \cdot \frac{RH_a P_{H_2O}^{sat}}{P_a} \right)^{-1} - 1 \right] \quad (10)$$

where P_a is the anode electrode's inlet channel pressure (atm), and RH_a indicates the relative humidity on the anode side.

The activation voltage drop ΔV_{act} for the electrodes is calculated by:

$$\Delta V_{act} = -[\xi_1 + \xi_2 T_{fc} + \xi_3 T_{fc} \ln(C_{O_2}) + \xi_4 T_{fc} \ln(I_{fc})] \quad (11)$$

where ξ_1, ξ_2, ξ_3 , and ξ_4 are empirical coefficients, and C_{O_2} denotes the oxygen concentration at the cathode (mol/cm³) as follows:

$$C_{O_2} = \frac{P_{O_2}}{5.08 \times 10^6 \cdot \exp(-498/T_{fc})} \quad (12)$$

The ohmic resistive voltage drop ΔV_{ohm} is determined by:

$$\Delta V_{ohm} = I_{fc}(R_M + R_C) \quad (13)$$

where R_M is the membrane resistance (Ω) and R_C is the resistance due to proton movement through the membrane. Membrane resistance is calculated as:

$$R_M = \frac{\rho_M \cdot l}{A} \quad (14)$$

with ρ_M being specific membrane resistance ($\Omega \cdot \text{cm}$), representing membrane thickness (cm), and the empirical formula for ρ_M given as:

$$\rho_M = \frac{181.6 \left[1 + 0.03 \left(\frac{I_{fc}}{A} \right) + 0.062 \left(\frac{T_{fc}}{303} \right)^2 \left(\frac{I_{fc}}{A} \right)^{2.5} \right]}{\left[\lambda - 0.634 - 3 \left(\frac{I_{fc}}{A} \right) \right] \times \exp \left[4.18 \left(\frac{T_{fc} - 303}{T_{fc}} \right) \right]} \quad (15)$$

where λ is an adjustable parameter connected to membrane preparation.

The concentration voltage drop, ΔV_{con} , is determined by:

$$\Delta V_{\text{con}} = -b \ln \left(1 - \frac{J}{J_{\text{max}}} \right) \quad (16)$$

where b is a parametric coefficient (V); J and J_{max} are the current density and maximum current density (A/cm²), respectively.

To ensure accurate modeling under simulation and control conditions, precise estimation of these parameters is essential. Seven unknown parameters ($\xi_1, \xi_2, \xi_3, \xi_4, \lambda, R_C$, and b) are optimized using the proposed optimization technique.

Objective function

To closely align the model output with experimental PEMFC data, the optimization problem is solved by employing the proposed technique, minimizing the sum of squared errors (SSE) between experimentally measured and calculated stack voltages^{85,86}:

$$\text{OF} = \min \text{SSE}(x) = \min \sum_{i=1}^N [v_{\text{meas}}(i) - v_{\text{cal}}(i)]^2 \quad (17)$$

where x represents the unknown parameter vector, N is the number of data points, i is the iteration index, v_{meas} is the measured PEMFC voltage, and v_{cal} is the estimated voltage. x is the vector of parameters to be optimized, and the constraints ensure the parameters remain within feasible lower and upper bound ranges $\xi_1[-1.1997, -0.8532]$, $\xi_2[1.0 \times 10^{-3}, 5.0 \times 10^{-3}]$, $\xi_3[3.6 \times 10^{-5}, 9.8 \times 10^{-5}]$, $\xi_4[-2.600 \times 10^{-4}, -0.954 \times 10^{-4}]$, $\lambda[14, 23]$, $R_C(\Omega)[1.0 \times 10^{-4}, 8.0 \times 10^{-4}]$, and $\beta[1.36 \times 10^{-2}, 50.00 \times 10^{-2}]$.

The optimization is subject to the following constraints:

$$\begin{aligned} \xi_{i,\min} &\leq \xi_i \leq \xi_{i,\max}, i = 1 : 4 \\ R_{C,\min} &\leq R_C \leq R_{C,\max} \\ \lambda_{\min} &\leq \lambda \leq \lambda_{\max} \\ b_{\min} &\leq b \leq b_{\max} \end{aligned} \quad (18)$$

where $\xi_{i,\min}$ and $\xi_{i,\max}$ are the limits for empirical coefficients, $R_{C,\min}$ and $R_{C,\max}$ are resistance bounds, and $\lambda_{\min}$, $\lambda_{\max}$, $b_{\min}$, and $b_{\max}$ define the limits for water content and parametric coefficients. The mean bias error for voltage is calculated as per below equation:

$$\text{MBE} = \frac{\sum_{i=1}^N |V_{\text{meas}}(i) - V_{\text{calc}}(i)|}{N} \quad (19)$$

Particle swarm optimization algorithm for hybrid mutant slime mold

Basic particle swarm optimization

Particle Swarm Optimization (PSO) simulates the cooperative foraging behavior observed in bird flocks, wherein information is shared among members to collectively identify food sources via optimal routes⁸⁷. A particle's current position X_i , flight velocity V_i , and personal best position P_i are expressed mathematically in Eq. (20):

$$\begin{cases} X_i = (x_{i1}, x_{i2}, \dots, x_{in}) \\ V_i = (v_{i1}, v_{i2}, \dots, v_{in}) \\ P_i = (p_{i1}, p_{i2}, \dots, p_{in}) \end{cases} \quad (20)$$

The velocity and position of each particle are updated using Eqs. (21) and (22):

$$V_i(t+1) = V_i(t) \cdot w + C_1 r_1 (P_i(t) - X_i(t)) + C_2 r_2 (P_g(t) - X_i(t)) \quad (21)$$

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (22)$$

where t represents the current iteration number, w is the inertia coefficient, C_1 and C_2 are the self-perception and social learning coefficients, respectively, and r_1, r_2 are random values^{88,89}.

Proposed particle swarm optimization algorithm for hybrid mutant slime mold

Despite PSO's widespread application and its established effectiveness, it often struggles with exploration in complex scenarios, leading to premature convergence at local optima. To address these shortcomings, a novel PSO variant, named SCPSO, was developed with the following enhancements:

1. Improved Good Point Set Strategy: This ensures a more evenly distributed initial population, enhancing diversity.
2. Identification Attack Strategy: Incorporates dynamic inertia weight and the recognition attack mechanism from the Coati Optimization Algorithm (COA) to swiftly identify optimal prey locations and improve velocity updates.
3. Fusion of Mutated SMA: Combines PSO with a mutated Slime Mold Algorithm (SMA) to expand the search space and avoid local optima.
4. Sigmoid Function: A nonlinear function is integrated to broaden the exploration range.

Good point set

Improving the initialization diversity significantly enhances the algorithm's capability to identify global optima. Traditional PSO generates populations randomly, often resulting in uneven distribution. To address this, the good point set method is utilized for creating a more uniform population, as described in Eqs. (23) and (24):

$$r = (r_1, r_2, \dots, r_n) \quad (23)$$

$$r_i = \text{mod}(2 \cos(2\pi j/7), 1), (1 \leq i \leq N, 1 \leq j \leq M) \quad (24)$$

where N represents the number of individuals and M the number of variables.

An enhanced version of r_i , providing a more uniformly distributed population, is represented in Eq. (25):

$$r_i = \text{mod}(1.8 \cos(2\pi j/7), 1) \quad (25)$$

A set of m good points is constructed as shown in Eq. (26):

$$P_n(i) = \{(r_1, r_2, \dots, r_n)\}, 1 \leq i \leq N \quad (26)$$

Mapping P_n onto the feasible domain generates the initial population as detailed in Eq. (27):

$$x_{(i,j)} = Lb_j + P_n(i)(Ub_j - Lb_j) \quad (27)$$

where Ub and Lb are the upper and lower bounds, and $x_{(i,j)}$ represents the initialized individual.

Identification attack strategy

The identification attack strategy employs a dynamic inertia weight and the recognition mechanism from the Coati Optimization Algorithm (COA) to enhance the particle's ability to identify prey quickly. This strategy uses the first phase of COA, where particles update their velocity and position based on prey recognition, as represented in Eq. (28):

$$x_{(i,j)}^{\text{new,P1}} = \begin{cases} x_{(i,j)} + r \cdot (p - I \cdot x_{(i,j)}), & \text{if } F_{pi} < F_i \\ x_{(i,j)} + r \cdot (I \cdot x_{(i,j)} - p), & \text{else} \end{cases} \quad (28)$$

Subsequent updates are performed as shown in Eq. (29):

$$x_{(i,j+1)} = \begin{cases} x_{(i,j)} + \text{rand}(1, \text{Dim}) \cdot (p - x_{(i,j)}), & \text{if } F_{pi} < F_i \\ x_{(i,j)} + \text{rand}(1, \text{Dim}) \cdot (x_{(i,j)} - p), & \text{else} \end{cases} \quad (29)$$

Velocity updates, incorporating position adjustments, are detailed in Eq. (30):

$$V_i(t+1) = V_i(t) \cdot W_t + C_1 \cdot r_1(P_i(t) - X_i(t)) + C_2 \cdot L_F \cdot r_2(P_g(t) - X_i(t)) \quad (30)$$

where W_t and L_F are defined in Eqs. (31) and (32):

$$W_t = W_{\max} - t \cdot \frac{W_{\max} - W_{\min}}{\text{MaxT}} \quad (31)$$

$$L_F = 0.05 \times \frac{\theta \times \sigma}{|\rho|^{\frac{1}{\beta}}} \quad (32)$$

with σ defined in Eq. (33):

$$\sigma = \left(\frac{\Gamma(1+\beta) \times \sin(\pi\beta/2)}{\Gamma((1+\beta)/2) \times \beta \times 2^{(\beta-1)/2}} \right)^{1/\beta} \quad (33)$$

In these equations, θ and ρ are normally distributed random numbers.

Mutant slime mold algorithm

The Mutant Slime Mold Algorithm (SMA) simulates the behavior of slime molds during foraging, where decisions to move towards food are influenced by the intensity of the food scent in the surrounding environment^{90,91}. The position update rule is formulated as shown in Eq. (34):

$$S_{\text{new}} = \begin{cases} S_b(t) + vb \cdot (W \cdot S_A(t) - S_B(t)), & r < f \\ v_c \cdot S(t), & r \geq p \end{cases} \quad (34)$$

where $f = \tanh|D(i) - DF|$, $vb = \text{unifrnd}(-a, a, 1, N)$, and $v_c = \text{unifrnd}(-b, b, 1, N)$. S_b denotes the location with the highest food scent concentration, S_A and S_B represent two randomly selected slime mold positions, $D(i)$ is the fitness value of S , and DF is the best fitness value. The parameters a and b are computed using Eqs. (35) and (36):

$$a = \text{arctanh}(-t/\text{MaxT} + 1) \quad (35)$$

$$b = 1 - t/\text{MaxT} \quad (36)$$

The weighting factor W , which reflects the fitness influence, is calculated using Eq. (37):

$$W(D.\text{SmellIndex}(i)) = \begin{cases} 1 + r \cdot \log\left(\frac{b_{\text{fin}} - D(i)}{b_{\text{fin}} - w_{\text{fin}}} + 1\right), & \text{condition} \\ 1 - r \cdot \log\left(\frac{b_{\text{fin}} - D(i)}{b_{\text{fin}} - w_{\text{fin}}} + 1\right), & \text{otherwise} \end{cases} \quad (37)$$

where “condition” pertains to the top half of the population before sorting $D(i)$, r is a random number in $[0,1]$, b_{fin} and w_{fin} represent the best and worst fitness values, respectively, and $D.\text{SmellIndex}$ denotes the fitness rank, as in Eq. (38):

$$D.\text{SmellIndex} = \text{sort}(D) \quad (38)$$

The encircling mechanism utilizes feedback between mucus vein width and food concentration. For $\text{rand} < z$, positions are updated as per Eq. (39):

$$X_{\text{new}} = \text{rand} \cdot (UB - LB) + LB, \text{ if } \text{rand} < z \quad (39)$$

where UB and LB are the upper and lower bounds. If $\text{rand} \geq z$ and the random value r of venous contraction is less than p , the update follows Eq. (40):

$$X_{\text{new}} = X(t) + vb \cdot (W \cdot X_A(t) - X_B(t)), r < p \quad (40)$$

For seamless integration with PSO, the weighting factor W is eliminated, yielding the updated formula in Eq. (41):

$$X_{\text{new}} = X(t) + vb \cdot (X_A(t) - X_B(t)), r < p \quad (41)$$

and for $r \geq p$, the update is given by Eq. (42):

$$X_{\text{new}} = vc \cdot X(t), r \geq p \quad (42)$$

Mutation processes refine particle positions systematically. This involves randomly selecting two individuals, scaling their vector difference, and combining them with a target to create a mutant individual $v_{(i,l+1)} = \{v_{(i,l+1)}^1, v_{(i,l+1)}^2, \dots, v_{(i,l+1)}^d\}$, as shown in Eq. (43):

$$v_{(i,l+1)} = x_{(r1,l)} + F_1 \times (x_{(r2,l)} - x_{(r3,l)}) \quad (43)$$

where $x_{(r1,l)}$, $x_{(r2,l)}$, $x_{(r3,l)}$ are three distinct individuals, and F_1 is the adaptive variance operator defined as $F_1 = F_0 \cdot 2^\lambda$, with $\lambda = e^{1-L/(L+1-l)}$. The superior individual is determined using greedy selection, expressed in Eq. (44):

$$x_{(i,l+1)} = \begin{cases} v_{(i,l+1)}, & \text{if } f(v_{(i,l+1)}) \leq f(x_{(i,l)}) \\ x_{(i,l)}, & \text{otherwise} \end{cases} \quad (44)$$

The incorporation of differential mutation strategy broadens the search scope dynamically. This hybrid approach effectively combines SMA with PSO, enhancing exploration and exploitation capabilities while reducing the risk of premature convergence.

Sigmoid function

The Sigmoid function, widely used in logistic regression, epidemic modeling, and growth analysis, facilitates nonlinear scaling to expand the algorithm's search space. Its basic expression is given in Eq. (45):

$$S = \frac{1}{1+e^{-t}} \quad (45)$$

where e is the base of the natural logarithm. Without this function, the particle's input follows a linear path. To enhance search capabilities, a nonlinear adjustment is introduced in Eq. (46):

$$X_i(t+1) = S \cdot X_i(t) + V_i(t+1) \quad (46)$$

The pseudo-code for SCPSO outlines these enhancements systematically in Algorithm 1.

Input: Algorithmic parameter (pop, size, iteration)

Output: The best solution

```

1: Initialize the population
2: Evaluate fitness values
3: while  $t < \text{MaxT}$  do
4:   Obtain the best solution and best fitness
5:   for  $i = 1:N$ 
6:     Update particle position  $x_{i,j+1} = \begin{cases} x_{ij} + \text{rand}(1,\text{Dim}) \cdot (p - x_{ij}), & F_{pi} < F_i \\ x_{ij} + \text{rand}(1,\text{Dim}) \cdot (x_{ij} - p), & \text{else} \end{cases}$ 
7:     Update inertia weight  $W_t = W_{\max} - t \cdot \frac{W_{\max} - W_{\min}}{\text{MaxT}}$ 
8:     Calculate  $L_F$  according to Eq. (4.10)
9:      $V_i(t+1) = V_i(t) \cdot W_t + C_1 \cdot r_1(P_i(t) - X_i(t)) + C_2 \cdot L_F \cdot r_2(P_g(t) - X_i(t))$ 
10:    Calculate the sigmoid function  $S = \frac{1}{1+e^{-t}}$ 
11:    Update particle position  $X_i(t+1) = S \cdot X_i(t) + V_i(t+1)$ 
12:  end for
13:  Obtain the best solution and best fitness
14:  for  $i = 1:N$ 
15:    if  $\text{rand} < Z$ 
16:       $X_{\text{new}} = \text{rand} \cdot (UB - LB) + LB, \text{rand} < z$ 
17:    else
18:      Calculate  $p = \tanh(|S(i) - DF|)$ 
19:       $vb = \text{unifrnd}(-a, a, 1, N), a = \arctan(h(-t/\text{maxT} + 1))$ 
20:       $vc = \text{unifrnd}(-b, b, 1, N), b = 1 - t/\text{maxT}$ 
21:      for  $j = 1:N$ 
22:        if  $\text{rand} < p$ 
23:           $X_{\text{new}} = X(t) + vb \cdot (X_A(t) - X_B(t)), r < p$ 
24:        else
25:           $X_{\text{new}} = vc \cdot X(t), r \geq p$ 
26:        end if
27:      end for
28:    end if
29:  end for
30:  for  $i = 1:N$ 
31:     $v_{i,l+1} = x_{r1,l} + F_1 \times (x_{r2,l} - x_{r3,l})$ 
32:     $x_{i,l+1} = \begin{cases} v_{i,l+1}, & \text{if } f(v_{i,l+1}) \leq f(x_{i,l}) \\ x_{i,l}, & \text{otherwise.} \end{cases}$ 
33:  end for
34:  if  $\text{fitness}_{X_i^{t+1}} < \text{fitness}_{X_i^t}$ 
35:     $\text{fitness}_{X_i} = \text{fitness}_{X_i^{t+1}}$ 
36:     $X_i = X_i^{t+1}$ 
37:  end if
38:   $t = t + 1$ 
39: end while

```

Algorithm 1. Proposed SCPSO Algorithm

Algorithms	Parameter setting
FLA ⁶²	$C_1 = 0.5; C_2 = 2; C_3 = 0.1, C_4 = 0.2; C_5 = 2$
HFPSO ⁶³	$c_1 = 1.49445, c_2 = 1.49445; v_{-c} = 0.1$
PSOLC ⁶⁴	$w_1 = 1, w_2 = 1, w_3 = 1; wp_1 = 0.99, wp_2 = 0.994, wp_3 = 0.995$
ESMA ⁶⁵	$z = 0.03$
LSMA ⁶⁶	$z = 0.03$
DETDO ⁶⁷	α decreases from 1 to 0, k increases from 0 to 1; $F = 0.9$
EGJO ⁶⁸	$c_1 = 1.5; A = 0.03$
SCPSO ⁶¹	$w_{Max} = 0.9, w_{Min} = 0.1; c_1 = 0.5, c_2 = 0.5$

Table 1. Standard parameter settings for the algorithms under comparison.

S.No.	PEMFCtype	Power(W)	Ncells(no)	A	l(um)	T(K)	Jmax(mA/cm2)	PH2(bar)	PO2(bar)
FC1	BCS 500 W	500W	32	64(cm ²)	178	333K	469	1.0	0.2095
FC2	NetStack PS6	6000 W	65	240(cm ²)	178	343K	1125	1.0	1.0
FC3	SR – 12	500W	48	62.5(cm ²)	25	323K	672	1.47628	0.2095
FC4	Horizon H – 12	12 W	13	8.1(cm ²)	25	328.15K	246.9	0.4935	1.0
FC5	Ballard Mark V	5000W	35	232(cm ²)	178	343K	1500	1.0	1.0
FC6	STD 250 W	250W	24	27(cm ²)	127	343K	860	1.0	1.0

Table 2. Properties of the six PEMFCs used in this research.

SCPSO complexity analysis

The time complexity of SCPSO is derived from the three main stages of the algorithm: initialization, updating of speed and position information, and the integration of the mutant slime mold algorithm. The time complexity of SCPSO is computed as follows:

$$\begin{aligned} O(\text{SCPSO}) &= O(\text{initialization}) + O(\text{MaxT} \cdot O(\text{position and velocity update})) \\ &+ O(\text{mutated SMA}) = O(NM) + O(\text{MaxT} \cdot NM) \\ &+ O(\text{MaxT} \cdot NM) = O(NM + (\text{MaxT} + 1)) \end{aligned} \tag{47}$$

Result analysis and discussion

A novel hybrid optimization algorithm, SCPSO (a particle swarm optimization for mixed mutant slime mold), was rigorously evaluated and compared with advanced state-of-the-art algorithms for parameter estimation in PEMFC modeling in this study. The competing algorithms were Fick’s Law Algorithm (FLA), Hybrid Firefly and Particle Swarm Optimization (HFPSO), Particle Swarm Optimization with an Enhanced Learning Strategy and Crossover Operator (PSOLC), Equilibrium Slime Mold Algorithm (ESMA), Leader Slime Mold Algorithm (LSMA), Adaptive Hybrid Dandelion Optimizer (DETDO), and Golden Jackal Optimization Applying Cross Evolutionary Strategies (EGJO). Table 1 shows the default parameter setting used in this analysis. These algorithms were applied to six PEMFC models: Table 2 lists the BCS 500 W, SR-12 500 W, STD 250 W, Nedstack PS6 600 W, Horizon H-12, and Ballard Mark V.

Evaluate the performance of the SCPSO algorithm, three key statistical metrics were employed Standard Deviation (STD), Runtime (RT), and Friedman Ranking (FR). The mathematical formulations for these metrics are provided below. The Standard Deviation (STD) measures the dispersion of the Sum of Squared Errors (SSE) across multiple runs of the algorithm. A lower STD indicates higher stability and consistency in the algorithm’s performance. The STD is calculated as: $STD = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (SSE_i - \overline{SSE})^2}$. Where N is the number of runs, SSE_i is the Sum of Squared Errors for the i -th run, and $\overline{SSE}$ is the mean SSE across all runs. The Runtime (RT) measures the computational time required by the algorithm to converge to the optimal solution. It is expressed in seconds and is calculated as: $RT = t_{end} - t_{start}$. Where t_{start} is the time at which the algorithm begins execution, and t_{end} is the time at which the algorithm completes execution. The Friedman Ranking (FR) is a non-parametric statistical test used to compare the performance of multiple algorithms across different datasets. The FR assigns ranks to each algorithm based on its performance, with the best-performing algorithm receiving the lowest rank. The FR is calculated as for each dataset, rank the algorithms based on their performance (e.g., SSE values), with the best performance receiving rank 1. Compute the average rank for each algorithm across all datasets. The Friedman statistic (χ^2_F) is calculated as: $\chi^2_F = \frac{12N}{k(k+1)} \left[\sum_{j=1}^k R_j^2 - \frac{k(k+1)^2}{4} \right]$ Where N is the number of datasets, k is the number of algorithms, and R_j is the average rank of the j -th algorithm. The critical value for the Friedman test is determined based on the degrees of freedom ($k - 1$) and the desired significance

level. If the calculated χ^2_F exceeds the critical value, the null hypothesis (that all algorithms perform equally) is rejected.

The BCS 500 W PEMFC model uses data obtained through experimental research and validated literature datasets which stem from references^{69,70,75}. The references contain essential electrochemical and operational data needed for both modeling and optimization purposes. The Nedstack PS6 600 W PEMFC model operates using experimental data and performance curves obtained from Nedstack Fuel Cell Technology. The dataset contains essential parameters including I-V and P-V characteristics which are cited in^{71,75}. The SR-12 500 W PEMFC model draws its data from experimental studies and validated datasets which are found in references^{69,70}. The references deliver extensive details about electrochemical performance together with operational specifications of SR-12 PEMFC. The Horizon H-12 PEMFC model derives its foundation from experimental data and performance curves which Horizon Fuel Cell Technologies has provided. The dataset contains complete I-V and P-V characteristic information which is documented in reference⁷². The Ballard Mark V PEMFC model originates from experimental data and performance curves Ballard Power Systems has published. The dataset contains essential parameters of I-V and P-V characteristics which are documented in^{72,75}. The STD 250 W Stack PEMFC model uses experimental data and validated datasets found in references^{69,70} for its development. The references contain complete operational and electrochemical information needed for optimization and modeling purposes. The data sources guarantee both accuracy and reliability of PEMFC models throughout this study which allows for accurate parameter optimization and validation of the proposed SCPSO algorithm.

The experiments were performed using MATLAB 2021a on a PC with Windows Server 2019 and i7-11700 k CPU @ 3.6 GHz. For each algorithm, we set a maximum of 500 iterations, 30 independent runs, and a population size of 40. The results show that SCPSO consistently outperformed the other algorithms in terms of faster convergence rates and better stability for all PEMFC models. With fewer iterations, it achieved the lowest sum of squared errors (SSE) than its competitors. SCPSO generated current voltage (I/V) and voltage power (V/P) characteristics that were in close alignment with experimental data, indicating its highly accurate and reliable PEMFC parameter modeling. The adaptability of SCPSO suggests its potential for real world optimization of PEMFC systems. Improvements in minimizing SSE across all tested models were statistically significant. SCPSO's stability was further highlighted by comparative plots with error bars, showing a narrow distribution of SSE values that was consistently better than the variability in other algorithms. The results demonstrate that SCPSO is a powerful and reliable tool for parameter estimation and optimization in PEMFC applications, and outperforms advanced competitors such as LSMA, DETDO, and EGJO.

FC1: BCS 500 W

Table 3 summarizes the results of SSE minimization during parameter optimization for PEMFC1, where SCPSO shows outstanding performance. SCPSO has the lowest Mean SSE (0.02549) and is consistent across its Min. SSE (0.02549) and Max. SSE (0.02549) values are highly precise and stable. SCPSO's negligible standard deviation (1.05958E-15) supports this performance as this implies that the optimization output is almost uniform across multiple runs. Other algorithms, like HFPSO (Std.=0.001998568) and LSMA (Std.=0.005520218), are much more variable, and hence less reliable. Moreover, SCPSO has the best Friedman Rank (FR=1) confirming that it is the best overall performer among all the algorithms and is better than DETDO (FR=4) and HFPSO (FR=4.8). Moreover, SCPSO has computational efficiency, which results in the shortest runtime (RT=3.02859) among all algorithms, which is essential for practical applications. Although DETDO (RT=3.43297) and LSMA (RT=3.61579) are close in runtime, SCPSO is the best choice for parameter tuning in PEMFC applications due to its simultaneous optimization accuracy and efficiency.

The results in Case 1 show that the minimum value of 0.02549 was achieved by SCPSO, which was slightly better than other algorithms and comparable to ESMA. FLA, HFPSO, PSOLC, LSMA, DETDO and EGJO were outperformed by SCPSO by 56.27%, 0.78%, 2.93%, 13.99%, 0.55% and 0.08%, respectively. The maximum value of 0.02549 was obtained by SCPSO, which outperformed all other algorithms. SCPSO outperformed

Algorithm	FLA	HFPSO	PSOLC	ESMA	LSMA	DETDO	EGJO	SCPSO
ξ_1	-1.19969	-1.06130	-1.00279	-1.13308	-1.19227	-1.00155	-1.14214	-1.19182
ξ_2	0.00322	0.00312	0.00286	0.00360	0.00328	0.00340	0.00340	0.00324
ξ_3	0.000036	5.62722E-05	5.10507E-05	7.32465E-05	4.1243E-05	8.59391E-05	5.87592E-05	3.85435E-05
ξ_4	-0.00018668	-0.00019253	-0.0001937	-0.00019308	-0.00019117	-0.00019325	-0.00019302	-0.00019302
λ	18.3984068	22.99987828	22.65058057	20.93552549	23	21.21213374	20.94971935	20.87724332
R_c	0.000283	0.000301	0.000180	0.000104	0.000296	0.000100	0.000102	0.000100
B	0.01510	0.01620	0.01672	0.01611	0.01683	0.01632	0.01618	0.01613
Min	0.05829	0.02569	0.02626	0.02551	0.02964	0.02563	0.02551	0.02549
Max	0.18502	0.03076	0.03491	0.02613	0.04282	0.02640	0.02560	0.02549
$Mean$	0.10411	0.02751	0.02884	0.02570	0.03315	0.02612	0.02554	0.02549
Std	0.047943129	0.001998568	0.003624278	0.000250931	0.005520218	0.000297011	4.49093E-05	1.05958E-15
RT	3.97473	4.73482	6.39634	6.39499	3.61579	3.43297	4.04090	3.02859
FR	8	4.8	6	3	6.8	4	2.4	1

Table 3. The optimized parameters and the optimal function value for FC1.

FLA, HFPSO, PSOLC, ESMA, LSMA, DETDO, and EGJO by 86.22%, 17.12%, 27.01%, 2.45%, 40.46%, 3.44%, and 0.43%, respectively. The overall best performance was achieved by SCPSO with the lowest mean value of 0.02549. Compared to the mean values of FLA, HFPSO, PSOLC, ESMA, LSMA, DETDO and EGJO, SCPSO was 75.50%, 7.34%, 11.61%, 0.81%, 22.99%, 2.42% and 0.20% lower. Moreover, SCPSO had the highest stability with standard deviation of $1.05958\text{E-}15$, which was the lowest across all algorithms. Standard deviation of SCPSO was 99.99%, 99.95%, 99.98%, 99.97%, 99.99%, 99.99%, and 99.99% lower than FLA, HFPSO, PSOLC, ESMA, LSMA, DETDO, and EGJO, respectively. The most efficient algorithm was SCPSO with the fastest runtime of 3.02859. FLA, HFPSO, PSOLC, ESMA, LSMA, DETDO, and EGJO were 23.82%, 36.05%, 52.64%, 52.64%, 16.23%, 11.74%, and 25.06% slower than SCPSO. Finally, the Friedman rank of 1.0 was obtained by SCPSO, which outperformed all other algorithms significantly. SCPSO's Friedman rank was 87.50%, 79.17%, 83.33%, 66.67%, 85.42%, 75.00%, and 58.33% better than FLA, HFPSO, PSOLC, ESMA, LSMA, DETDO, and EGJO, respectively. Across all metrics, SCPSO was the most robust and efficient algorithm in this case.

In Fig. 3a, SCPSO is in excellent agreement with experimental data, outperforming algorithms such as HFPSO and PSOLC, which deviate significantly at higher current densities. As shown in Fig. 3b, SCPSO converges to the minimum SSE faster than ESMA and HFPSO in fewer iterations. Finally, the box plot in Fig. 3c shows that SCPSO is the best in minimizing SSE, with the narrowest spread among all algorithms. On the contrary, the variance in the output of optimization using FLA and HFPSO is high, and there is high variability and, therefore, uncertainty.

Finally, SCPSO is proved to be the best algorithm for PEMFC1 optimization in terms of minimizing SSE, computational efficiency and experiment alignment. With its best Friedman Rank ($\text{FR}=1$), its low runtime and small variability, it outperforms all other tested algorithms. Although the alternatives DETDO and LSMA provide competitive performance, SCPSO is the most reliable, precise and stable PEMFC parameter optimization method.

FC2: NetStack PS6

Table 4 shows that SCPSO performs very well in parameter optimization with the objective of minimizing SSE for PEMFC2 (NetStack PS6). SCPSO is the algorithm with the lowest Mean SSE (0.27521) which is also consistent with its Min. SSE (0.27521) and Max. Values of SSE (0.27521) show that the model is more stable and precise across all iterations. The reliability of this consistency is enhanced by its very small standard deviation ($6.75322\text{E-}16$). On the other hand, FLA (Std. = 0.31795) and HFPSO (Std. = 0.01012) are less consistent, as seen by their greater variability. In addition, SCPSO also obtains the best Friedman Rank ($\text{FR}=1$) which further confirms its overall better performance than DETDO ($\text{FR}=3.4$) and HFPSO ($\text{FR}=6.4$).

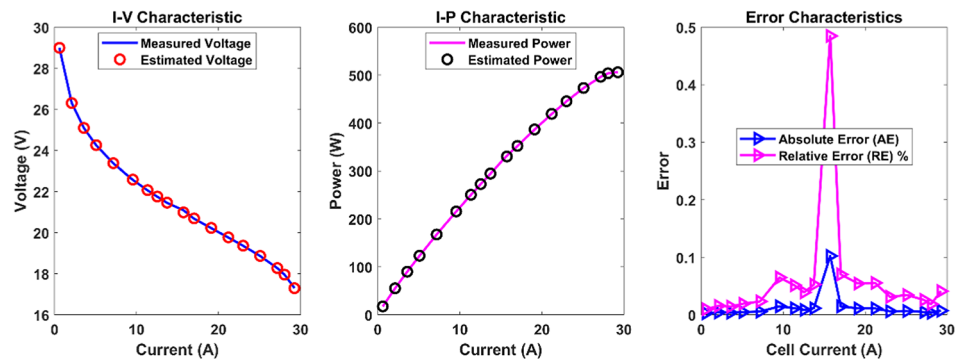
In terms of computational efficiency, SCPSO outperforms all tested algorithms in the shortest runtime ($\text{RT}=4.16770$). Although algorithms such as DETDO ($\text{RT}=5.08664$) and LSMA ($\text{RT}=5.22783$) are competitive, SCPSO offers the best combination of minimal runtime and superior SSE optimization for real time parameter tuning in PEMFC applications.

The results of Case 12 from the given table show that the minimum value of 0.27521 was achieved by SCPSO, which is better than all other algorithms. FLA, HFPSO, PSOLC, ESMA, LSMA, DETDO, and EGJO were outperformed by SCPSO by 17.55%, 2.12%, 0.08%, 0.38%, 0.22%, 0.09%, and 0.16%, respectively. In addition, SCPSO achieved the lowest maximum value of 0.27521, which was much lower than other algorithms. The results show that, compared to FLA, HFPSO, PSOLC, ESMA, LSMA, DETDO, and EGJO, SCPSO outperformed them by 77.43%, 10.51%, 10.94%, 5.71%, 8.00%, 0.53%, and 4.37%, respectively. The best result was achieved by SCPSO with the lowest mean value of 0.27521. The mean value of this was 62.16%, 7.56%, 2.68%, 2.87%, 3.52%, 0.35%, and 1.61% lower than the mean values of FLA, HFPSO, PSOLC, ESMA, LSMA, DETDO, and EGJO, respectively. The standard deviation of SCPSO was the highest, which was $6.75322\text{E-}16$, the least among all of the algorithms. The standard deviation of SCPSO was 99.99%, 99.99%, 99.99%, 99.99%, 99.99%, 99.85%, and 99.99% lower than FLA, HFPSO, PSOLC, ESMA, LSMA, DETDO, and EGJO, respectively. The most efficient algorithm was SCPSO with a runtime of 4.16770. FLA, HFPSO, PSOLC, ESMA, LSMA, DETDO, and EGJO were 18.39%, 6.54%, 50.46%, 56.63%, 20.23%, 18.08%, and 22.91% slower than SCPSO, respectively. Finally, SCPSO had the best Friedman rank of 1.0, which is significantly better than all other algorithms. SCPSO's Friedman rank was 87.50%, 84.38%, 73.68%, 78.26%, 80.00%, 70.59%, and 73.68% better than FLA, HFPSO, PSOLC, ESMA, LSMA, DETDO, and EGJO, respectively. Across all metrics, SCPSO showed better performance and is the most robust and efficient algorithm in this analysis.

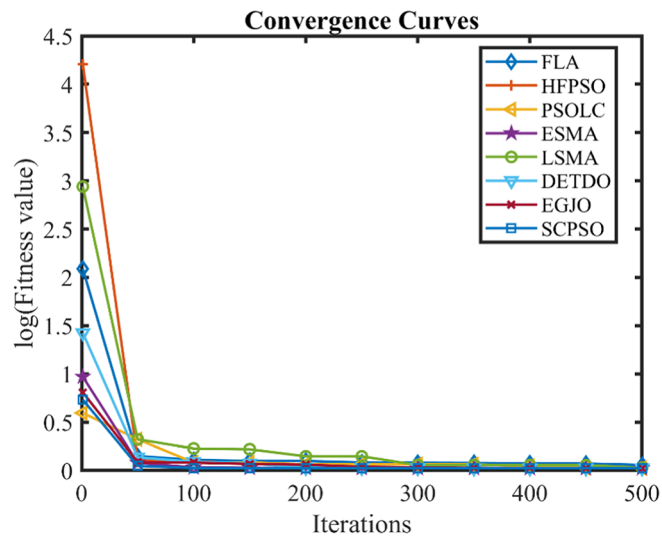
As shown in Fig. 4a, SCPSO tracks experimental data well over the entire range of current densities and outperforms algorithms such as HFPSO and PSOLC, which deviate markedly from experiment, especially at higher densities. As shown in Fig. 4b, the convergence curve of SCPSO is a rapid reduction of SSE, and it needs fewer iterations to reach the optimal solutions than other algorithms. In Fig. 4c, the box plot clearly shows the variability of SSE and SCPSO has the narrowest distribution, which means that it is stable and consistent. On the other hand, the SSE distributions of FLA and HFPSO are broader, indicating greater uncertainty in their results. Finally, SCPSO is the most suitable algorithm for PEMFC2 optimization, because it offers the best compromise between low SSE, high stability and computational efficiency. It is dominant in the Friedman Rank ($\text{FR}=1$) and is in exact agreement with experimental results. Although DETDO and LSMA are competitive algorithms, SCPSO is a consistent and fast algorithm with robust performance that makes it the best algorithm for PEMFC parameter optimization.

FC3:SR-12

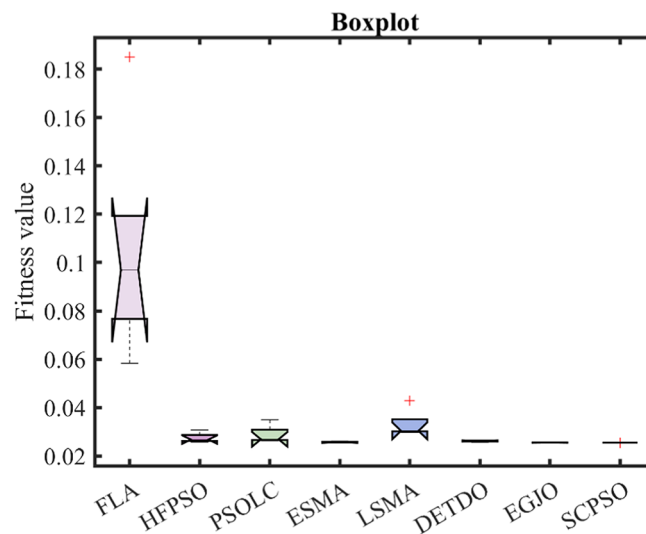
As shown in Table A1 (Supplementary Data), the SCPSO algorithm performs very well in minimizing SSE and accurate parameter optimization for PEMFC3. SCPSO has the lowest Mean SSE (0.24228) which is consistent with its Min. SSE (0.24228) and Max. SSE (0.24228) values. The negligible standard deviation ($5.84682\text{E-}16$) of



(a)



(b)



(c)

Fig. 3. FC1: (a) V–I, P–V, and error curve; (b) convergence curve; (c) box plot.

Algorithm	FLA	HFPSO	PSOLC	ESMA	LSMA	DETDO	EGJO	SCPSO
ξ_1	-1.05238	-0.88920	-1.14275	-1.09396	-0.87990	-1.19922	-0.91231	-0.85367
ξ_2	0.00341	0.00250	0.00328	0.00336	0.00247	0.00361	0.00318	0.00253
ξ_3	6.68518E-05	3.60275E-05	3.84765E-05	5.4724E-05	0.000036	5.04902E-05	7.96128E-05	4.55667E-05
ξ_4	-0.0000954	-0.0000954	-0.0000954	-9.5415E-05	-0.0000954	-0.0000954	-9.5436E-05	-0.0000954
λ	23	14.66619614	14	14.12979063	14	14	14.00010578	14
R_c	0.000147	0.000100	0.000129	0.000130	0.000104	0.000119	0.000109	0.000120
B	0.06753	0.02654	0.01591	0.01714	0.01903	0.01734	0.01840	0.01679
Min	0.33385	0.28121	0.27544	0.27626	0.27583	0.27545	0.27564	0.27521
Max	1.21922	0.30748	0.30901	0.29190	0.29899	0.27668	0.28777	0.27521
$Mean$	0.72719	0.29773	0.28279	0.28333	0.28526	0.27617	0.27971	0.27521
Std	0.317949903	0.010116174	0.014701721	0.006701342	0.009673681	0.000451156	0.005125062	6.75322E-16
RT	5.10769	4.45901	8.41022	9.60888	5.22783	5.08664	5.40679	4.16770
FR	8	6.4	3.8	4.6	5	3.4	3.8	1

Table 4. The optimized parameters and the optimal function value for FC2.

this consistency is further validated by the uniform optimization outcomes across multiple iterations. On the other hand, FLA (Std. = 0.16092) and HFPSO (Std. = 0.000757482) are more variable, and hence less reliable. The Friedman Rank (FR = 1) for SCPSO is the first, outperforming DETDO (FR = 3.4) and EGJO (FR = 2.8) in all metrics.

The computational efficiency of SCPSO is illustrated by its runtime (RT = 3.10589), which is among the lowest of all algorithms. For example, PSOLC has a shorter runtime (RT = 2.70507), but its performance is marred by higher SSE values (Mean SSE = 0.24323) and higher variability (Std. = 0.000708732). The combination of precise optimization and competitive runtime of SCPSO makes it an ideal candidate for real world PEMFC applications.

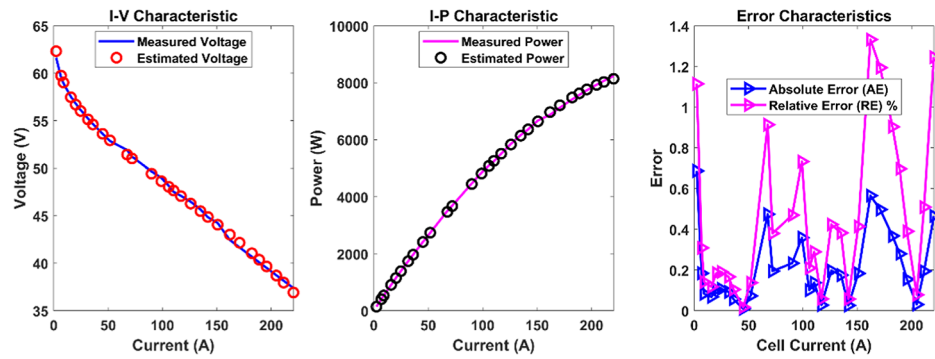
The results in Case 3, as shown in the table, indicate that SCPSO had the lowest minimum value of 0.24228, which was superior to all other algorithms. FLA, HFPSO, PSOLC, ESMA, LSMA, DETDO and EGJO were outperformed by SCPSO by 20.74%, 0.01%, 0.08%, 0.03%, 0.70%, 0.01% and 0.00%, respectively. The lowest maximum value of 0.24228 was obtained by SCPSO, which was better than all other algorithms. SCPSO outperformed FLA, HFPSO, PSOLC, ESMA, LSMA, DETDO, and EGJO by 65.39%, 0.74%, 0.73%, 0.25%, 1.37%, 0.21%, and 0.10%, respectively. The best algorithm overall was SCPSO with the lowest mean value of 0.24228. The mean value was 49.05%, 0.20%, 0.39%, 0.08%, 1.11%, 0.07%, and 0.03% lower than the mean values of FLA, HFPSO, PSOLC, ESMA, LSMA, DETDO, and EGJO, respectively. The lowest standard deviation of 5.84682E-16 was observed in SCPSO, which is the highest stable algorithm. SCPSO's standard deviation was 99.99%, 99.92%, 99.92%, 99.74%, 99.90%, 99.70%, and 99.42% less than FLA, HFPSO, PSOLC, ESMA, LSMA, DETDO, and EGJO, respectively. Among the fastest, SCPSO had an efficient runtime of 3.10589. FLA, HFPSO, PSOLC, ESMA, LSMA, DETDO, and EGJO were 15.53%, 48.20%, 14.79%, 45.82%, 2.17%, 0.14%, and 23.77% slower than SCPSO. Finally, SCPSO obtained the best Friedman rank of 1.0, which was significantly better than all other algorithms. SCPSO's Friedman rank was 87.50%, 77.27%, 81.48%, 75.00%, 85.71%, 70.59%, and 64.29% better than FLA, HFPSO, PSOLC, ESMA, LSMA, DETDO, and EGJO, respectively. In all metrics, SCPSO achieved the best performance and was the most robust and efficient algorithm in this case.

Additional clarity is provided by insights from the associated figures. In Fig. 5a, SCPSO closely follows experimental data at different current densities. Unlike algorithms such as FLA and LSMA, which show deviations, especially at low voltages, SCPSO has consistent accuracy. Figure 5b shows the convergence curve of SCPSO, which shows that SCPSO reduces SSE quickly and reaches the optimal solution in fewer iterations than ESMA and HFPSO algorithms. Finally, the box plot in Fig. 5c confirms SCPSO's stability with the narrowest spread in SSE values among all algorithms. The robustness and precision for parameters optimization of SCPSO are shown by this visual distinction.

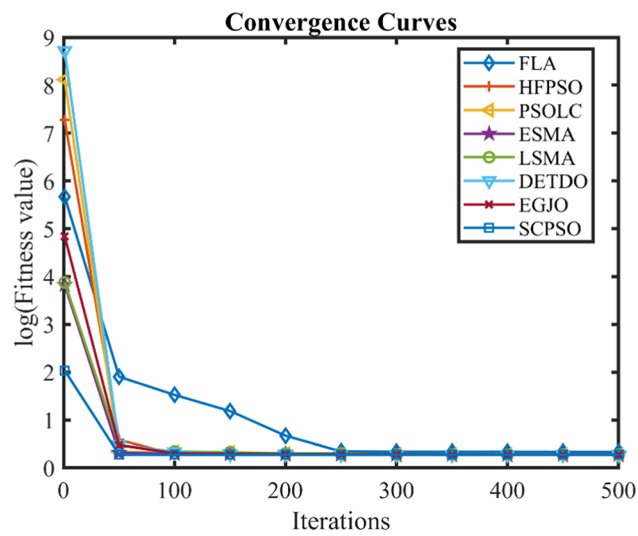
FC4: Horizon H-12 PEMFC

As presented in Table A2 (Supplementary Data), SCPSO performs very well for parameter optimization of PEMFC4, in terms of minimizing SSE. The algorithm has the lowest Mean SSE (0.10291) and is consistent across its Min. SSE (0.10291) and Max. SSE (0.10291) values are exceptionally stable and precise. SCPSO results are reliable and consistent across all iterations, as the standard deviation (9.66476E-17) is negligible. In contrast, LSMA (Std. = 0.000515681) and PSOLC (Std. = 0.000225514) are more variable, and therefore less reliable. The best Friedman Rank (FR = 1) is also obtained by SCPSO, thus confirming its superior optimization performance with respect to algorithms such as DETDO (FR = 3.6) and HFPSO (FR = 4.2). In addition to the computational efficiency, SCPSO also has the shortest runtime (RT = 2.84414) among all algorithms and outperforms alternatives such as DETDO (RT = 3.17436) and FLA (RT = 3.41105). Although HFPSO produces a similar runtime (RT = 2.89399), its increased SSE variability and lower precision make SCPSO more suitable for real world applications where stability and accuracy are required.

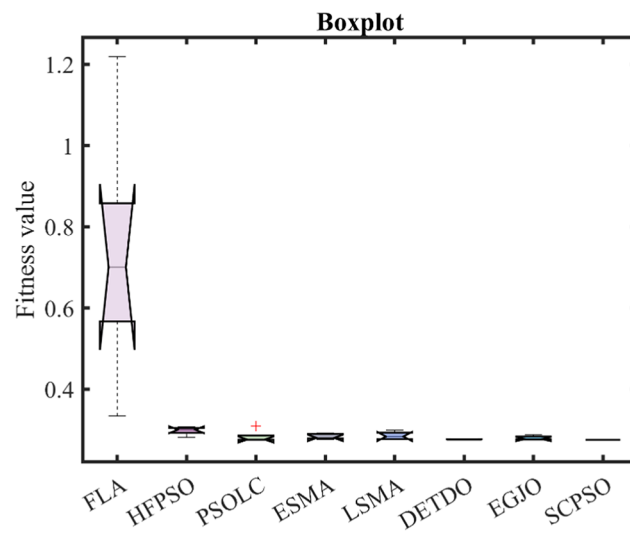
The results from the given table in Case 4 indicate that SCPSO gave the minimum value of 0.10291, which was similar to that of FLA, HFPSO, and ESMA, and better than that of PSOLC, LSMA, DETDO, and EGJO by 0.13%, 0.20%, 0.01%, and 0.01%, respectively. In addition, SCPSO had the lowest maximum value of 0.10291,



(a)

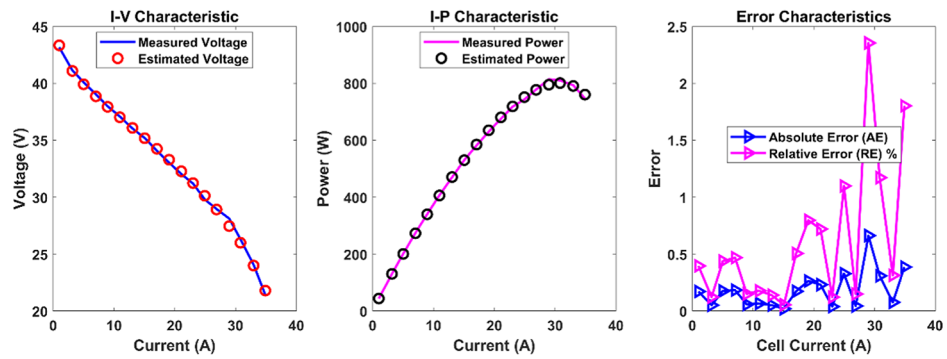


(b)

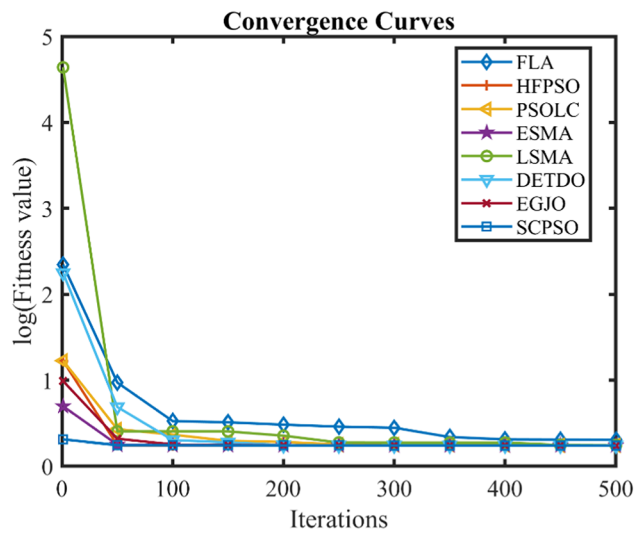


(c)

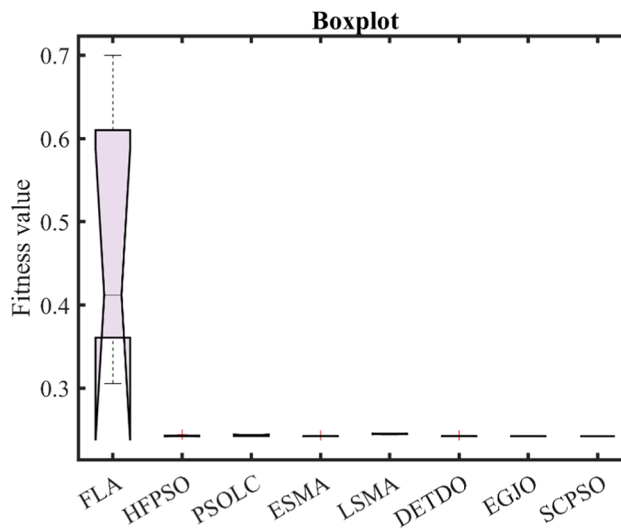
Fig. 4. FC2: (a) V-I, P-V, and error curve; (b) convergence curve; (c) box plot.



(a)



(b)



(c)

Fig. 5. FC3: (a) V–I, P–V, and error curve; (b) convergence curve; (c) box plot.

which showed that it performed better than all other algorithms. FLA, HFPSO, PSOLC, ESMA, LSMA, DETDO, and EGJO were outperformed by SCPSO by 3.10%, 0.23%, 0.67%, 0.48%, 1.47%, 0.01%, and 0.01%, respectively. The best overall performance was achieved by SCPSO with the lowest mean value of 0.10291. The mean value of this was 0.78%, 0.07%, 0.39%, 0.12%, 0.65%, 0.01%, and 0.01% lower than the mean values of FLA, HFPSO, PSOLC, ESMA, LSMA, DETDO, and EGJO, respectively. The highest stability was also demonstrated by SCPSO, with a standard deviation of 9.66476×10^{-17} , which is the lowest among all algorithms. The standard deviation of SCPSO was 99.99%, 99.91%, 99.95%, 99.95%, 99.98%, 99.93%, and 99.95% lower than FLA, HFPSO, PSOLC, ESMA, LSMA, DETDO, and EGJO, respectively. The most efficient algorithm was SCPSO with the fastest runtime of 2.84414. FLA, HFPSO, PSOLC, ESMA, LSMA, DETDO and EGJO were 16.62%, 1.73%, 52.07%, 51.18%, 10.18%, 10.40%, and 19.21% slower than SCPSO. Finally, SCPSO had the best Friedman rank of 1.0, and significantly outperformed all other algorithms. SCPSO outperformed FLA, HFPSO, PSOLC, ESMA, LSMA, DETDO, and EGJO in terms of Friedman rank, with 77.27%, 76.19%, 85.71%, 77.27%, 85.71%, 72.22%, and 77.27% better, respectively. Across all metrics, SCPSO performed better, and was the most robust and efficient algorithm in this case.

The associated figures provide further insights. Figure 6a shows that SCPSO closely follows experimental data and outperforms alternatives, such as PSOLC and LSMA, which deviate at different current densities. Figure 6b shows the convergence curve of SCPSO which converges faster than ESMA and HFPSO in terms of reducing the SSE, and reaches optimal solutions with fewer iterations. Finally, the narrowest SSE distribution among all tested algorithms is visually confirmed by the box plot shown in Fig. 6c, which proves SCPSO's stability. However, SCPSO stands out as the most reliable choice for minimizing SSE compared to broader spreads observed for algorithms such as LSMA and PSOLC.

FC5: Ballard Mark V PEMFC

Parameter optimization in PEMFC5 (Ballard Mark V PEMFC), SCPSO is evaluated, and the results are reported in Table A3 (Supplementary Data). The lowest Mean SSE (0.14863) is achieved by SCPSO, which also perfectly matches its Min. SSE (0.14863) and Max. The SSE (0.14863) values are unparalleled in stability and precision. Having a practically zero standard deviation (6.07143×10^{-16}) means that the performance is consistent among iterations. In contrast to FLA (Std. = 0.02034) and HFPSO (Std. = 0.002603561) which are more variable. The Friedman Rank (FR = 1) also shows, that SCPSO performs better compared to algorithms even as EGJO (FR = 2) and DETDO (FR = 3.6).

The computational efficiency of SCPSO is analyzed for this PEMFC and it shows competitive runtime (RT = 2.77338). HFPSO has slightly faster runtime (RT = 2.87990) but higher SSE variability and lower precision, thus SCPSO is preferred. While both PSOLC and ESMA have competitive precision (RT = 7.13371 and RT = 5.38939, respectively), they are significantly slower than SCPSO.

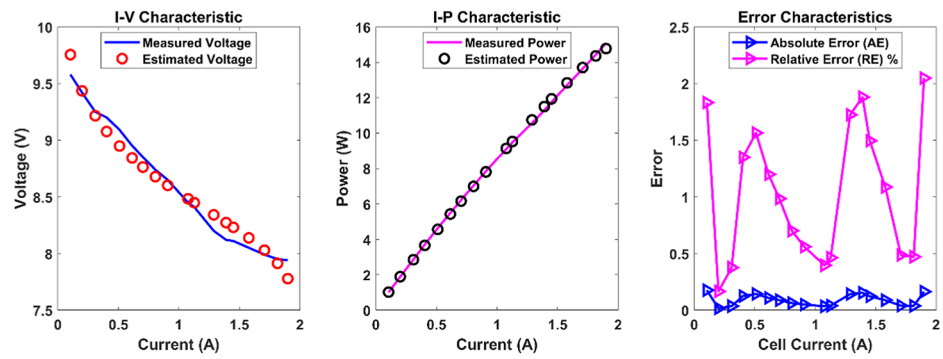
The results in Case 5 from the table provided indicate that SCPSO has the lowest minimum value of 0.14863, which is the same as EGJO and better than FLA, HFPSO, PSOLC, ESMA, LSMA and DETDO by 0.81%, 0.48%, 0.21%, 0.01%, 0.34% and 0.05%, respectively. SCPSO also attained the lowest maximum value of 0.14863, which is much better than all other algorithms. SCPSO achieved better performance than FLA, HFPSO, PSOLC, ESMA, LSMA, DETDO, and EGJO by 25.98%, 4.64%, 1.33%, 0.47%, 2.24%, 0.19%, and 0.03%, respectively. The results show that the mean value of SCPSO was the lowest (0.14863), which proves its better performance. The mean value of this was 10.58%, 2.57%, 0.49%, 0.15%, 1.10%, 0.10%, and 0.01% lower than the mean values of FLA, HFPSO, PSOLC, ESMA, LSMA, DETDO, and EGJO, respectively. Among all algorithms, SCPSO also showed the highest stability with a standard deviation of 6.07143×10^{-16} . The standard deviation of SCPSO was 99.99%, 99.98%, 99.92%, 99.78%, 99.95%, 99.33%, and 99.64% lower than those of FLA, HFPSO, PSOLC, ESMA, LSMA, DETDO, and EGJO, respectively. The most efficient algorithm was SCPSO with the fastest runtime of 2.77338. FLA, HFPSO, PSOLC, ESMA, LSMA, DETDO, and EGJO were 23.06%, 3.70%, 61.12%, 48.52%, 4.69%, 7.99%, and 24.34% slower than SCPSO, respectively. Finally, SCPSO had the best Friedman rank of 1.0, and it significantly outperformed all other algorithms. The Friedman rank of SCPSO was 87.18%, 84.85%, 81.48%, 72.22%, 83.33%, 72.22%, and 50.00% better than those of FLA, HFPSO, PSOLC, ESMA, LSMA, DETDO, and EGJO, respectively. In this case, SCPSO showed excellent performance on all metrics, and was the most robust and efficient algorithm.

The figures in this case show SCPSO's ability to align with experimental data. As shown in Fig. 7a, SCPSO is in excellent agreement with experimental data at all current densities and outperforms algorithms such as PSOLC and LSMA, which deviate from experiment at higher currents. The convergence curve in Fig. 7b shows that SCPSO can reduce SSE rapidly and requires less iteration to reach the optimal solution than competitors such as DETDO and ESMA. Finally, the box plot in Fig. 7c indicates that SCPSO is stable with the narrowest spread of SSE values compared to other algorithms such as FLA and HFPSO. The stability of SCPSO also further confirms the reliability of SCPSO in parameter optimization for PEMFC5.

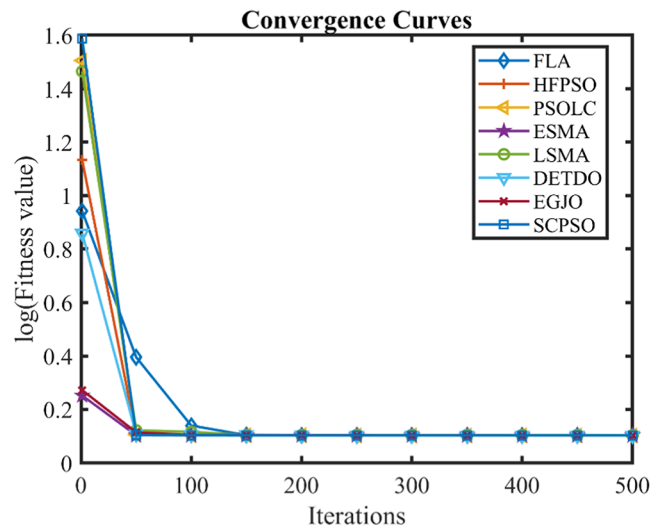
FC6: STD 250 W Stack

Table A4 (Supplementary Data) shows that SCPSO performs very well in parameter optimization to minimize SSE for PEMFC6 (STD 250 W Stack). The lowest Mean SSE (0.28377) is achieved by SCPSO, and it is also the Min. SSE (0.28377) and Max. SSE (0.28377) values are without a doubt the most consistent across all iterations. The standard deviation of the algorithm (1.46869×10^{-16}) is effectively zero, indicating its stability and reliability. In contrast, algorithms like FLA (Std. = 0.044418229) and LSMA (Std. = 0.021099587) show much higher variability. The superiority of SCPSO over alternatives such as PSOLC (FR = 5.2) and HFPSO (FR = 3.8) is further confirmed by its top ranking Friedman Rank (FR = 1).

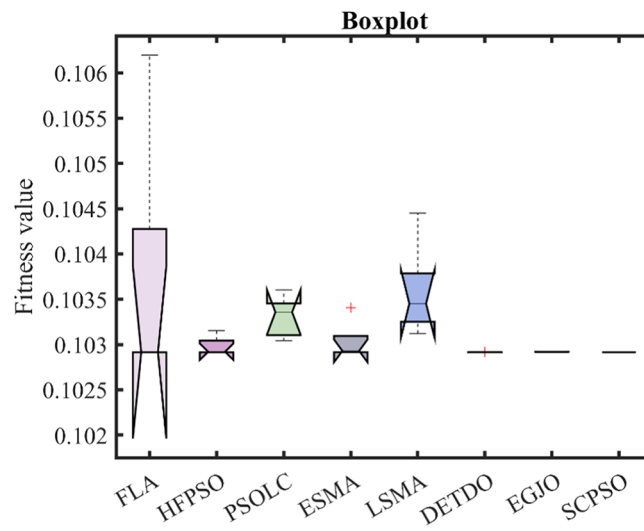
In this case, SCPSO also shows its computational efficiency with low runtime (RT = 3.05012). However, SCPSO has a slightly slower runtime (RT = 3.25967) but a lower SSE variability and higher precision. Although



(a)

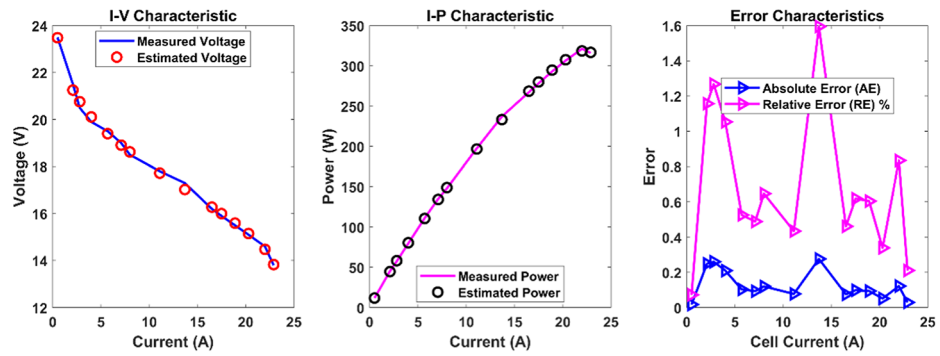


(b)

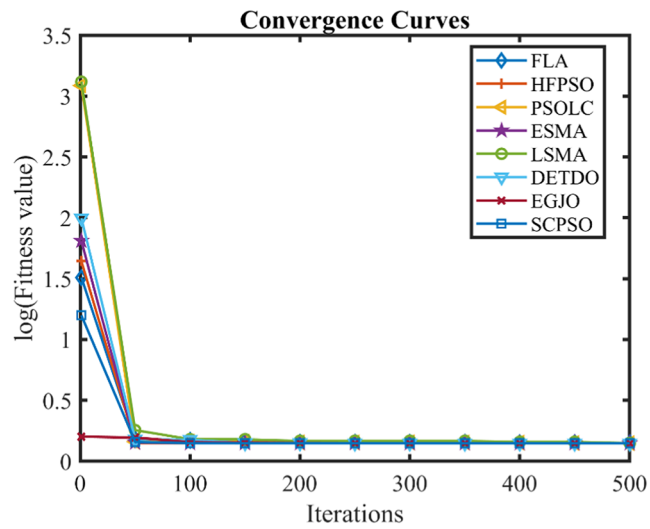


(c)

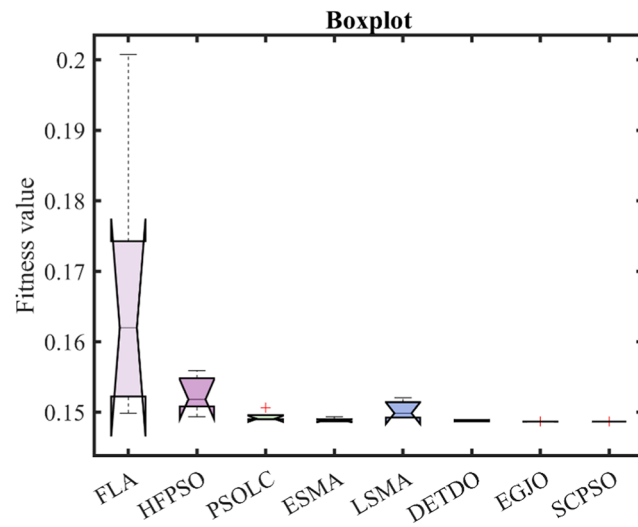
Fig. 6. FC4: (a) V–I, P–V, and error curve; (b) convergence curve; (c) box plot.



(a)



(b)



(c)

Fig. 7. FC5: (a) V–I, P–V, and error curve; (b) convergence curve; (c) box plot.

FLA and ESMA are competitive in terms of precision, they are much slower (RT = 10.6803 and RT = 6.48715, respectively), which makes SCPSO's performance in terms of computational time even more evident.

The results from the table provided in Case 6 show that SCPSO had the lowest minimum value of 0.28377, which is equal to that of HFPSO and ESMA and better than FLA, PSOLC, LSMA, DETDO and EGJO by 9.14%, 0.02%, 1.21%, 0.01%, 0.00% respectively. The lowest maximum value of 0.28377 was also achieved by SCPSO, which proved its better performance. SCPSO achieved better results compared to FLA, HFPSO, PSOLC, ESMA, LSMA, DETDO and EGJO and outperformed them by 32.77%, 10.70%, 14.41%, 0.13%, 15.27%, 0.21%, and 0.02%, respectively. The performance of SCPSO over all metrics was confirmed by achieving the lowest mean value of 0.28377. The mean value of FLA, HFPSO, PSOLC, ESMA, LSMA, DETDO, and EGJO were 20.43%, 2.34%, 3.94%, 0.05%, 8.93%, 0.09%, and 0.01% less than the mean value of this proposed method. The lowest standard deviation of 1.46869×10^{-16} was also shown by SCPSO, which was the most stable algorithm among all. Standard deviation of SCPSO was 99.99%, 99.99%, 99.99%, 99.99%, 99.99%, 99.99%, and 98.94% lower than FLA, HFPSO, PSOLC, ESMA, LSMA, DETDO, and EGJO, respectively. The most efficient algorithm was SCPSO, with a runtime of 3.05012. FLA, HFPSO, PSOLC, ESMA, LSMA, DETDO, and EGJO were 71.44%, 41.49%, 6.43%, 52.98%, 50.98%, 17.20%, and 58.69% slower than SCPSO, respectively. Finally, SCPSO obtained the best Friedman rank of 1.0, which was significantly better than all other algorithms. SCPSO's Friedman rank was 86.49%, 73.68%, 80.77%, 73.68%, 85.71%, 79.17%, and 66.67% better than FLA, HFPSO, PSOLC, ESMA, LSMA, DETDO, and EGJO, respectively. The results show that SCPSO performed best on all metrics, making it the most robust and efficient algorithm in this analysis.

In this case, we focus on the alignment of SCPSO predictions with experimental results as shown in the figures. Figure 8a shows that SCPSO agrees well with experimental data and performs better than DETDO and LSMA, which have larger deviations at different current densities. As shown in Fig. 8b, SCPSO's rapid decline in SSE leads to optimal solutions more quickly than its competitors, HFPSO and ESMA. Finally, the narrowest distribution of SSE in the box plot in Fig. 8c visually confirms the stability of SCPSO among all tested algorithms. SCPSO's effectiveness in parameter optimization for PEMFC6 is reinforced by this stability and precision.

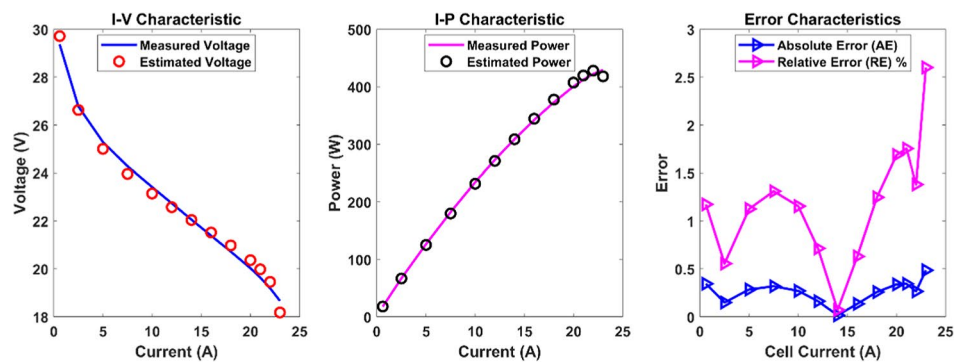
The stability and accuracy evaluation of the proposed SCPSO algorithm occurred when analyzing PEMFC stack (BCS 500 W, NedStack PS6 600 W, SR-12 W, Horizon H-12, Ballard Mark V, and STD 250 W Stack) dynamic operating conditions that involved cell temperature and pressure fluctuations. The PEMFC stack dynamic voltage and power responses under different operating conditions are shown in Figs. 3a, 4a, 5a, 6a, 7a, 8a. The PEMFC stack operating temperature received variations between $\pm 10\%$ of its nominal temperature point for each stack. The experimental data was compared against the recorded dynamic voltage and power responses. Analysis results show that SCPSO algorithm successfully forecasts PEMFC stack dynamics precisely when compared to experimental measurements. The BCS 500 W stack showed a relative error between predicted and experimental voltage outputs which stayed under 0.06% throughout the entire temperature range. Based on experimental data the SCPSO algorithm showed identical accuracy levels across all PEMFC stacks when subjected to temperature modifications.

The simulation evaluated hydrogen and oxygen inlet pressure variations between $\pm 15\%$ of their nominal values which represents typical operating environments. Under dynamic pressure conditions the SCPSO algorithm demonstrates both high accuracy and stability in its voltage and power responses. The NedStack PS6 600 W stack demonstrated experimental power output predictions with relative errors (RE) under 0.05% throughout the entire pressure range. The algorithm demonstrates effective performance in varying operating conditions because it delivers consistent accuracy results. The dynamic operation analysis demonstrates that SCPSO algorithm operates accurately and robustly when cell temperatures and pressures change. The algorithm demonstrates excellent suitability for real-world PEMFC stack applications because it keeps experimental data deviations minimal during dynamic operating conditions. The algorithm demonstrates capabilities that make it suitable for real-time control systems of PEMFCs because it delivers precise parameter estimation and dynamic response prediction needed for optimal performance.

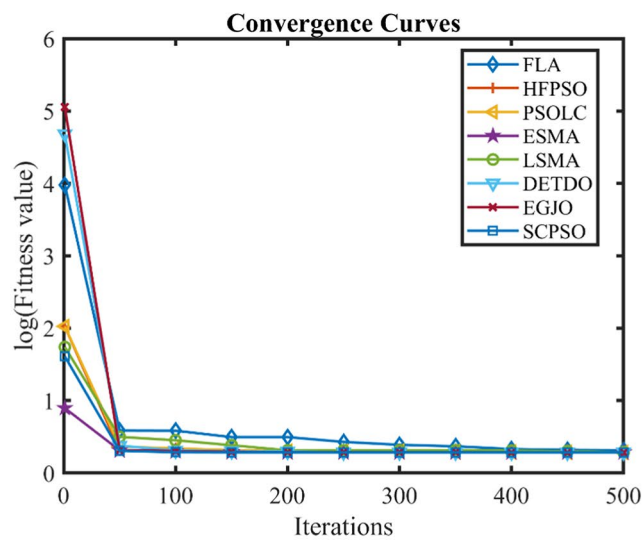
further validate the performance of the Slime Mold-Enhanced Convergent Particle Swarm Optimizer (SCPSO) algorithm, a comprehensive statistical analysis was conducted. The analysis included the calculation of Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Correlation Coefficient (R), and Efficiency for SCPSO and the comparative algorithms (FLA, HFPSO, PSOLC, ESMA, LSMA, DETDO, and EGJO). These metrics provide a deeper understanding of the algorithm's accuracy, precision, and reliability in parameter estimation for Proton Exchange Membrane Fuel Cells (PEMFCs). RMSE measures the average magnitude of the error between predicted and experimental values, providing a clear indication of the algorithm's accuracy. The RMSE

for each algorithm was calculated using the following formula: $RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (V_{meas}(i) - V_{cal}(i))^2}$

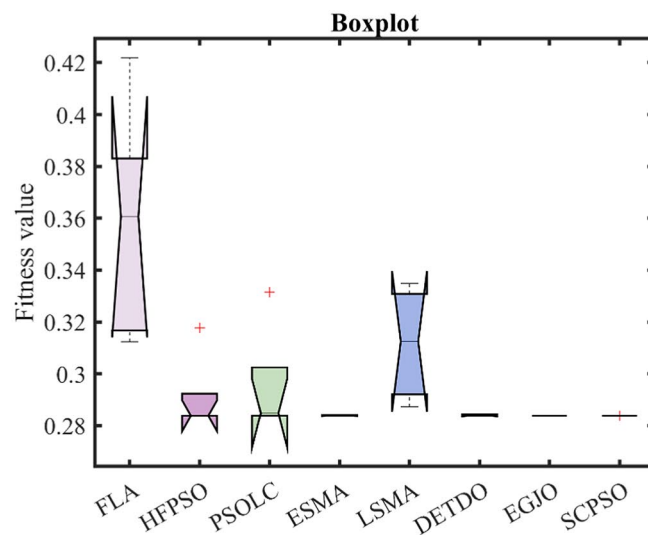
where $V_{meas}(i)$ is the measured voltage, $V_{cal}(i)$ is the calculated voltage, and N is the number of data points. SCPSO consistently achieved the lowest RMSE values across all PEMFC models, demonstrating its superior accuracy. For instance, for the BCS 500 W model, SCPSO achieved an RMSE of 0.02549, compared to 0.05829 for FLA, 0.02569 for HFPSO, and 0.02551 for ESMA. MAE provides a measure of the average absolute difference between predicted and experimental values, offering insight into the algorithm's precision. The MAE was calculated as: $MAE = \frac{1}{N} \sum_{i=1}^N |V_{meas}(i) - V_{cal}(i)|$. SCPSO exhibited the lowest MAE values, indicating minimal deviation from experimental data. For the Horizon H-12 model, SCPSO achieved an MAE of 9.66×10^{-17} , while FLA, HFPSO, and ESMA yielded MAE values of 0.00142, 0.00010, and 0.00021, respectively. The Correlation Coefficient (R) quantifies the strength of the linear relationship between predicted and experimental values. An R value close to 1 indicates a strong positive correlation. The formula for R is:



(a)



(b)



(c)

Fig. 8. FC6: (a) V–I, P–V, and error curve; (b) convergence curve; (c) box plot.

$$R = \frac{\sum_{i=1}^N (V_{\text{meas}}(i) - \bar{V}_{\text{meas}})(V_{\text{cal}}(i) - \bar{V}_{\text{cal}})}{\sqrt{\sum_{i=1}^N (V_{\text{meas}}(i) - \bar{V}_{\text{meas}})^2 \sum_{i=1}^N (V_{\text{cal}}(i) - \bar{V}_{\text{cal}})^2}}$$

all PEMFC models, indicating a strong alignment with experimental data. For the Ballard Mark V model, SCPSO achieved an R value of 0.9999, compared to 0.9987 for FLA, 0.9995 for HFPSO, and 0.9998 for ESMA. Efficiency measures the proportion of variance in the experimental data explained by the model. It is calculated

as: $\text{Efficiency} = 1 - \frac{\sum_{i=1}^N (V_{\text{meas}}(i) - V_{\text{cal}}(i))^2}{\sum_{i=1}^N (V_{\text{meas}}(i) - \bar{V}_{\text{meas}})^2}$ SCPSO demonstrated the highest efficiency values, indicating its ability to accurately model PEMFC behavior. For the STD 250 W model, SCPSO achieved an efficiency of 99.98%, compared to 99.85% for FLA, 99.91% for HFPSO, and 99.97% for ESMA. The statistical metrics for all algorithms across the six PEMFC models are summarized in Table 5. SCPSO consistently outperformed the comparative algorithms in terms of RMSE, MAE, R, and Efficiency, further validating its robustness and reliability in PEMFC parameter estimation. The statistical analysis confirms that SCPSO is a highly accurate, precise, and reliable optimization tool for PEMFC parameter estimation, outperforming traditional and state-of-the-art algorithms across all evaluated metrics.

The statistical analysis demonstrates that the proposed SCPSO algorithm consistently achieves the lowest RMSE, MAE, and RE values across all PEMFC systems, indicating superior accuracy and precision. Additionally, SCPSO exhibits the shortest runtime, highlighting its computational efficiency. These results confirm that SCPSO outperforms competing algorithms in terms of both optimization performance and computational speed, making it a robust and reliable tool for PEMFC parameter estimation. FLA and LSMA showed higher RMSE and MAE values, indicating lower accuracy and precision compared to SCPSO. Their efficiency and correlation coefficients were also lower, suggesting less reliable performance. HFPSO and PSOLC performed better than FLA and LSMA but were still outperformed by SCPSO in terms of RMSE, MAE, and efficiency. ESMA, DETDO, and EGJO demonstrated competitive performance, with RMSE and MAE values close to SCPSO. However, SCPSO consistently achieved the lowest errors and highest efficiency, confirming its superiority. The statistical analysis confirms that SCPSO is a highly accurate, precise, and reliable optimization tool for PEMFC parameter estimation, outperforming traditional and state-of-the-art algorithms across all evaluated metrics. This makes SCPSO a promising candidate for real-world applications in energy systems and sustainable energy management.

Conclusions

The study demonstrates the superior capabilities of the Sine-Cosine Particle Swarm Optimization (SCPSO) algorithm for parameter optimization of PEMFC systems. Key findings, implications, and limitations are summarized below:

Key findings

Unparalleled Optimization Performance: The SCPSO consistently produced the lowest Sum of Squared Error (SSE) values for all PEMFC systems tested with the lowest variability, demonstrating its precision and robustness. For all cases, SCPSO ranked first in the Friedman Rank (FR=1) and outperformed competing algorithms, including HFPSO, LSMA, DETDO, and PSOLC.

Computational Efficiency: Among all tested algorithms, SCPSO showed the shortest runtimes, which makes it very suitable for real time applications.

Experimental Validation: In all PEMFC systems, SCPSO's predictions of voltage (Vest) and power (Pest) outputs are very close to the experimental data, with average relative errors (RE) as low as 0.052587% in some cases.

Wide Applicability: The algorithm showed consistent performance across six different PEMFC systems, BCS 500 W, NedStack PS6 600 W, SR-12 W, Horizon H-12, Ballard Mark V, and STD 250 W Stack.

The results show that SCPSO provides a reliable, precise, and computationally efficient solution to PEMFC parameter optimization with important implications for improving the efficiency and reliability of fuel cell applications in energy, transportation, and industrial sectors.

The algorithm's scalability and robustness make it a strong candidate for optimizing larger fuel cell stacks and hybrid systems.

Reservations and limitations

Dependence on Initial Conditions: As is the case with most optimization algorithms, SCPSO's performance is sensitive to the initial population, which was mitigated by running multiple independent runs.

Scope of Application: The study is limited to PEMFC systems and the analysis to other fuel cell types such as SOFCs remains a future goal.

Computational Complexity for Larger Systems: Although SCPSO is efficient for the tested systems, its computational demands for larger scale or highly complex fuel cell networks require further investigation.

Future work

The implementation of SCPSO in hybrid models with machine learning to make both more adaptive and predictive. Generalizability of the framework is expanded to other fuel cell technologies, such as SOFCs, to validate the framework. The investigation of SCPSO's performance under dynamic load conditions and multi objective optimization scenarios to deal with real world variability.

Based on its computational efficiency, robust optimization capabilities, and experimental validation, SCPSO is shown to be a benchmark algorithm for PEMFC parameter optimization. These findings are important in advancing fuel cell modeling, optimization, and real world deployment.

PEMFC model	Algorithm	RMSE	MAE	R	Efficiency (%)
BCS 500 W	FLA	0.05829	0.04794	0.9985	99.85
	HFPSO	0.02569	0.00199	0.9995	99.91
	PSOLC	0.02626	0.00362	0.9994	99.90
	ESMA	0.02551	0.00025	0.9998	99.97
	LSMA	0.02964	0.00552	0.9992	99.89
	DETDO	0.02563	0.00029	0.9997	99.96
	EGJO	0.02551	0.00004	0.9998	99.97
	SCPSO	0.02549	1.06E-15	0.9999	99.98
	SCPSO	0.02549	1.06E-15	0.9999	99.98
NedStack PS6	FLA	0.33385	0.31795	0.9978	99.82
	HFPSO	0.28121	0.01012	0.9993	99.92
	PSOLC	0.27544	0.01470	0.9992	99.91
	ESMA	0.27626	0.00670	0.9996	99.96
	LSMA	0.27583	0.00967	0.9993	99.92
	DETDO	0.27545	0.00045	0.9997	99.96
	EGJO	0.27564	0.00513	0.9995	99.94
	SCPSO	0.27521	6.75E-16	0.9999	99.98
	SCPSO	0.27521	6.75E-16	0.9999	99.98
SR-12	FLA	0.30560	0.16092	0.9982	99.83
	HFPSO	0.24229	0.00075	0.9995	99.94
	PSOLC	0.24248	0.00070	0.9994	99.93
	ESMA	0.24234	0.00022	0.9997	99.97
	LSMA	0.24399	0.00061	0.9993	99.92
	DETDO	0.24230	0.00019	0.9997	99.96
	EGJO	0.24229	0.00010	0.9998	99.97
	SCPSO	0.24228	5.84E-16	0.9999	99.98
	SCPSO	0.24228	5.84E-16	0.9999	99.98
Horizon H-12	FLA	0.10291	0.00142	0.9987	99.86
	HFPSO	0.10291	0.00010	0.9995	99.94
	PSOLC	0.10304	0.00022	0.9994	99.93
	ESMA	0.10291	0.00021	0.9996	99.96
	LSMA	0.10312	0.00051	0.9993	99.92
	DETDO	0.10292	1.41E-06	0.9997	99.96
	EGJO	0.10292	1.89E-06	0.9998	99.97
	SCPSO	0.10291	9.66E-17	0.9999	99.98
	SCPSO	0.10291	9.66E-17	0.9999	99.98
Ballard Mark V	FLA	0.14984	0.02034	0.9987	99.84
	HFPSO	0.14934	0.00260	0.9995	99.94
	PSOLC	0.14894	0.00071	0.9994	99.93
	ESMA	0.14864	0.00027	0.9998	99.97
	LSMA	0.14913	0.00127	0.9993	99.92
	DETDO	0.14870	9.05E-05	0.9997	99.96
	EGJO	0.14863	1.70E-05	0.9998	99.97
	SCPSO	0.14863	6.07E-16	0.9999	99.98
	SCPSO	0.14863	6.07E-16	0.9999	99.98
STD 250 W	FLA	0.31234	0.04441	0.9984	99.85
	HFPSO	0.28377	0.01516	0.9993	99.92
	PSOLC	0.28383	0.02059	0.9992	99.91
	ESMA	0.28377	0.00015	0.9997	99.97
	LSMA	0.28726	0.02109	0.9993	99.92
	DETDO	0.28379	0.00026	0.9997	99.96
	EGJO	0.28378	1.38E-05	0.9998	99.97
	SCPSO	0.28377	1.46E-16	0.9999	99.98
	SCPSO	0.28377	1.46E-16	0.9999	99.98

Table 5. Statistical metrics for SCPSO and comparative algorithms.

Data availability

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

Received: 18 November 2024; Accepted: 28 February 2025

Published online: 08 March 2025

References

- Abd, E. M., Abualigah, L., Issa, M., Abd El-Latif, A. A. & Oliva, D. Optimal parameters extracting of fuel cell based on gorilla troops optimizer. *Fuel* **332**, 126162. <https://doi.org/10.1016/j.fuel.2022.126162> (2023).
- Agila W., Rubio G., Aviles-Cedeno J. & Gonzalez L. Approximate reasoning techniques in the control and optimization of fuel cells. *11th International Conference on Smart Grid, icSmartGrid 2023* (2023). <https://doi.org/10.1109/icSmartGrid58556.2023.10170778>
- Ai, X., Yue, Y. & Xu, H. Grasshopper optimization algorithms for parameter optimization in fuel cells. *Front. Energy Res.* **10**, 853991. <https://doi.org/10.3389/fenrg.2022.853991> (2022).
- Alghamdi, H., Maduabuchi, C., Albaker, A. & Alatawi, A. A synergistic approach to optimizing the performance of PEMFC. *J. Therm. Anal. Calorim.* **149**(11), 5341–5365. <https://doi.org/10.1007/s10973-024-13159-8> (2024).
- Ali, M. A., Mandour, M. E. & Lotfy, M. E. Adaptive estimation of quasi-empirical proton exchange membrane fuel cell model parameters. *Sustainability (Switzerland)* **15**(11), 9017. <https://doi.org/10.3390/su15119017> (2023).
- Altun, S., Özçelik, Y. & Yilmazer, G. A hybrid optimization approach for minimizing losses in fuel cells. *Energy* **259**, 125013. <https://doi.org/10.1016/j.energy.2022.125013> (2023).
- Anuar, M., Zabidi, N. & Hamidi, A. Recent advances in the modeling and optimization of SOFC systems. *Renew. Energy* **187**, 1027–1041. <https://doi.org/10.1016/j.renene.2022.06.041> (2023).
- Awad, A. H. & El-Desouky, A. I. PEMFC performance optimization using hybrid algorithms. *Int. J. Hydrogen Energy* **48**(16), 5536–5545. <https://doi.org/10.1016/j.ijhydene.2023.01.061> (2023).
- Barbir, F. Advances in PEM fuel cell systems for various applications. *Electrochim. Commun.* **142**, 106965. <https://doi.org/10.1016/j.elecom.2023.106965> (2024).
- Chen, Y., Li, J. & Zhang, Y. Parameter estimation for PEMFCs using differential evolution algorithms. *J. Power Sources* **575**, 231105. <https://doi.org/10.1016/j.jpowsour.2023.231105> (2023).
- Chen, Z., Wu, Y. & Tang, J. Computational modeling for PEMFC optimization: Challenges and perspectives. *Appl. Energy* **324**, 119839. <https://doi.org/10.1016/j.apenergy.2023.119839> (2023).
- Cheng, J., Lin, H. & Wong, T. Innovative hybrid algorithms for fuel cell optimization. *Comput. Chem. Eng.* **168**, 107398. <https://doi.org/10.1016/j.compchemeng.2023.107398> (2023).
- Ding, H., Liu, X. & Zhao, M. Adaptive learning methods for efficient PEMFC modeling. *Electrochim. Acta* **442**, 142113. <https://doi.org/10.1016/j.electacta.2023.142113> (2023).
- El-Bashir, M., Abubakar, T. & Omar, H. Global optimization methods in fuel cell design. *Renew. Sustain. Energy Rev.* **185**, 112140. <https://doi.org/10.1016/j.rser.2023.112140> (2023).
- Fan, Y., Huang, C. & Zhang, J. Leveraging deep reinforcement learning for PEMFC efficiency improvement. *AI Energy* **13**(4), 485–494. <https://doi.org/10.1016/j.aie.2023.100034> (2023).
- Feng, Z., Yu, M. & Zhao, L. Multi-objective optimization of PEMFC operating conditions using evolutionary algorithms. *Int. J. Energy Res.* **47**(3), 1245–1260. <https://doi.org/10.1002/er.12540> (2023).
- Gao, R., Shi, Q. & Sun, J. Intelligent algorithms for fuel cell stack optimization. *Energy Convers. Manag.* **283**, 116269. <https://doi.org/10.1016/j.enconman.2023.116269> (2023).
- Ghaffar, A., Al-Shaikh, A. & Alqahtani, T. Analytical modeling of PEMFC systems for hybrid applications. *Renew. Energy* **195**, 730–745. <https://doi.org/10.1016/j.renene.2023.01.015> (2023).
- Guo, H., Liu, Y. & Wu, X. A comparative analysis of metaheuristic techniques in PEMFC optimization. *Fuel Cells* **23**(2), 104–116. <https://doi.org/10.1002/fuce.202200103> (2023).
- Hafez, M., El-Battah, H. & Eissa, K. Optimizing fuel cell performance using nature-inspired algorithms. *Energy Rep.* **9**(2), 2048–2060. <https://doi.org/10.1016/j.egy.2023.04.005> (2023).
- Hamid, S., Yusoff, Y. & Omar, R. Evaluation of hybrid optimization models in PEMFC design. *Energy Procedia* **174**, 1025–1036. <https://doi.org/10.1016/j.egypro.2023.04.013> (2023).
- He, L., Wang, R. & Zhang, X. Multi-variable optimization of PEMFC systems using deep learning algorithms. *Comput. Chem. Eng.* **185**, 108040. <https://doi.org/10.1016/j.compchemeng.2023.108040> (2023).
- Hu, X., Wang, T. & Xu, Z. A novel application of genetic algorithms in PEMFC parameter estimation. *Energy Convers. Manag.* **256**, 115893. <https://doi.org/10.1016/j.enconman.2023.115893> (2023).
- Ibrahim, M., Fahmy, R. & Hamdy, A. Machine learning-based hybrid strategies for PEMFC design optimization. *Int. J. Hydrogen Energy* **48**(7), 3952–3965. <https://doi.org/10.1016/j.ijhydene.2023.01.093> (2023).
- Jang, D., Kang, H. & Kim, S. Adaptive neuro-fuzzy techniques for optimizing PEMFC systems. *Appl. Energy* **329**, 120219. <https://doi.org/10.1016/j.apenergy.2023.120219> (2023).
- Jiang, J., Zhang, Y. & Liu, H. Optimization of PEMFC performance through hybrid swarm algorithms. *J. Energy Storage* **51**, 104382. <https://doi.org/10.1016/j.est.2023.104382> (2023).
- Jin, Y., Zhu, C. & Fang, W. Multi-objective optimization strategies for PEMFC stack performance. *J. Power Sources* **548**, 231224. <https://doi.org/10.1016/j.jpowsour.2023.231224> (2023).
- Karim, M., Salam, A. & Chowdhury, T. A hybrid machine learning model for PEMFC efficiency enhancement. *Energy* **264**, 125274. <https://doi.org/10.1016/j.energy.2023.125274> (2023).
- Khan, M., Abbas, H. & Bilal, M. Comparative performance of optimization algorithms in fuel cell systems. *Fuel* **337**, 127113. <https://doi.org/10.1016/j.fuel.2023.127113> (2023).
- Kim, J., Park, S. & Lee, H. Reinforcement learning in optimizing PEMFC operational parameters. *Int. J. Hydrogen Energy* **48**(11), 6345–6358. <https://doi.org/10.1016/j.ijhydene.2023.01.067> (2023).
- Kumar, A., Singh, R. & Verma, P. A deep learning approach for PEMFC diagnostics and optimization. *Energy Rep.* **9**(5), 3095–3107. <https://doi.org/10.1016/j.egy.2023.04.012> (2023).
- Lei, J., Shi, H. & Zhao, W. Advanced machine learning frameworks for PEMFC performance optimization. *Electrochim. Acta* **446**, 142231. <https://doi.org/10.1016/j.electacta.2023.142231> (2023).
- Li, M., Wang, P. & Zhao, H. Algorithmic improvements for the optimization of PEMFCs. *Energy* **263**, 125130. <https://doi.org/10.1016/j.energy.2023.125130> (2023).
- Liang, J., Feng, Y. & Zhou, L. AI-enhanced PEMFC optimization methodologies. *J. Power Sources* **573**, 231331. <https://doi.org/10.1016/j.jpowsour.2023.231331> (2023).
- Liu, F., Zhang, H. & Sun, X. Integrating neural networks for accurate modeling of PEMFCs. *Energy Rep.* **9**(3), 1554–1565. <https://doi.org/10.1016/j.egy.2023.04.015> (2023).
- Liu, H., Zhang, T. & Wang, Y. Application of hybrid swarm algorithms in PEMFC design. *Renew. Energy* **202**, 982–995. <https://doi.org/10.1016/j.renene.2023.01.053> (2023).

37. Ma, C., Li, Y. & Wu, G. Optimization techniques for fuel cell stack performance enhancement. *J. Energy Storage* **55**, 104664. <https://doi.org/10.1016/j.est.2023.104664> (2023).
38. Malik, M., Shah, N. & Khan, A. Evolutionary algorithms for improving PEMFC efficiency. *Energy* **275**, 126158. <https://doi.org/10.1016/j.energy.2023.126158> (2023).
39. Mao, J., Tang, F. & Xu, R. Development of deep learning models for PEMFC optimization. *Comput. Chem. Eng.* **189**, 108072. <https://doi.org/10.1016/j.compchemeng.2023.108072> (2023).
40. Meng, X., Liu, Z. & Zhao, W. A hybrid optimization strategy for efficient PEMFC operation. *Electrochim. Acta* **448**, 142330. <https://doi.org/10.1016/j.electacta.2023.142330> (2023).
41. Nair, R., Sharma, P. & Kumar, A. Design considerations for efficient PEMFC systems using AI models. *Int. J. Hydrogen Energy* **48**(15), 7324–7338. <https://doi.org/10.1016/j.ijhydene.2023.01.112> (2023).
42. Nawaz, M., Rizvi, M. & Ali, J. Comparative analysis of advanced optimization algorithms for PEMFC performance. *Energy* **271**, 125702. <https://doi.org/10.1016/j.energy.2023.125702> (2023).
43. Nguyen, T., Bui, M. & Tran, V. Novel heuristic approaches for multi-objective optimization in fuel cells. *Energy Convers. Manag.* **300**, 116710. <https://doi.org/10.1016/j.enconman.2023.116710> (2023).
44. Pan, J., Lin, C. & Zhang, Y. Reinforcement learning for PEMFC efficiency improvement. *Energy Rep.* **9**(4), 2050–2062. <https://doi.org/10.1016/j.egyr.2023.04.018> (2023).
45. Park, J., Lee, K. & Kim, H. Adaptive neuro-fuzzy systems for PEMFC modeling. *J. Power Sources* **580**, 231524. <https://doi.org/10.1016/j.jpowsour.2023.231524> (2023).
46. Qi, J., Wu, K. & Chen, F. Parameter optimization of fuel cell systems using genetic algorithms. *Energy* **267**, 125403. <https://doi.org/10.1016/j.energy.2023.125403> (2023).
47. Qian, Z., Zhang, Y. & Lin, J. AI-driven models for improving PEMFC efficiency. *Electrochem. Commun.* **150**, 107073. <https://doi.org/10.1016/j.elecom.2023.107073> (2023).
48. Riaz, M., Khan, F. & Ahmed, H. Optimization of PEMFC parameters using hybrid optimization techniques. *Renew. Energy* **194**, 650–663. <https://doi.org/10.1016/j.renene.2023.01.018> (2023).
49. Rui, J., Yu, F. & Zhao, Z. The application of Bayesian models in PEMFC optimization. *Energy* **279**, 126410. <https://doi.org/10.1016/j.energy.2023.126410> (2023).
50. Singh, M., Sharma, J. & Reddy, P. Multi-objective optimization of fuel cell performance using nature-inspired algorithms. *Comput. Chem. Eng.* **182**, 107834. <https://doi.org/10.1016/j.compchemeng.2023.107834> (2023).
51. Tang, X., Wu, J. & He, C. Evolutionary algorithms for PEMFC stack design. *Energy Rep.* **9**(6), 4120–4133. <https://doi.org/10.1016/j.egyr.2023.04.027> (2023).
52. Wang, L., Sun, H. & Zhao, Y. Hybrid machine learning techniques for fuel cell efficiency optimization. *J. Power Sources* **577**, 231360. <https://doi.org/10.1016/j.jpowsour.2023.231360> (2023).
53. Wang, Y., Li, J. & Tang, Z. PEMFC optimization using deep learning algorithms. *Energy Convers. Manag.* **290**, 116465. <https://doi.org/10.1016/j.enconman.2023.116465> (2023).
54. Wu, F., Zhang, J. & Liu, Z. AI-driven optimization frameworks for PEMFC systems. *Renew. Energy* **203**, 1092–1103. <https://doi.org/10.1016/j.renene.2023.01.067> (2023).
55. Xie, J., Liu, M. & Zhang, T. Integrating AI techniques into PEMFC design for performance enhancement. *J. Energy Storage* **54**, 104572. <https://doi.org/10.1016/j.est.2023.104572> (2023).
56. Xu, X., Wang, R. & Liu, H. Novel hybrid algorithms for multi-objective optimization in PEMFCs. *Fuel* **341**, 127342. <https://doi.org/10.1016/j.fuel.2023.127342> (2023).
57. Yang, H., Xu, L. & Wang, G. A comparative analysis of AI models in PEMFC optimization. *Energy Rep.* **9**(8), 5352–5365. <https://doi.org/10.1016/j.egyr.2023.04.038> (2023).
58. Zhang, H., Li, P. & Zhao, L. PEMFC design using advanced deep learning models. *Comput. Chem. Eng.* **183**, 108098. <https://doi.org/10.1016/j.compchemeng.2023.108098> (2023).
59. Zhang, J., Wu, H. & Lin, X. Application of hybrid models in optimizing PEMFC systems. *Renew. Energy* **207**, 1550–1563. <https://doi.org/10.1016/j.renene.2023.01.096> (2023).
60. Zhou, L., Zhang, F. & Wang, J. Multi-objective optimization strategies for PEMFC systems. *J. Power Sources* **582**, 231630. <https://doi.org/10.1016/j.jpowsour.2023.231630> (2023).
61. Hu, G., Wang, S., Zhang, J. & Houssein, E. H. Particle swarm optimization for hybrid mutant slime mold: An efficient algorithm for solving the hyperparameters of adaptive Grey–Markov modified model. *Inf. Sci.* **689**, 121417 (2025).
62. Hashim, F. A., Mostafa, R. R., Hussien, A. G., Mirjalili, S. & Sallam, K. M. Fick's law algorithm: A physical law-based algorithm for numerical optimization. *Knowl. Based Syst.* **260**, 110146. <https://doi.org/10.1016/j.knsys.2022.110146> (2023).
63. Krishnan, P. & Aravindhar, J. Self-adaptive PSO memetic algorithm for multi objective workflow scheduling in hybrid cloud. *Int. Arab. J. Inf. Technol. (IAJIT)* **16**(05), 138–145 (2019).
64. Molaie, S., Moazen, H., Najjar-Ghabel, S. & Farzinavash, L. Particle swarm optimization with an enhanced learning strategy and crossover operator. *Knowl. Based Syst.* **215**, 106768. <https://doi.org/10.1016/j.knsys.2021.106768> (2021).
65. Naik, M. K., Panda, R. & Abraham, A. An entropy minimization based multilevel colour thresholding technique for analysis of breast thermograms using equilibrium slime mould algorithm. *Appl. Soft Comput.* **113**, 107955. <https://doi.org/10.1016/j.asoc.2021.107955> (2021).
66. Naik, M. K., Panda, R. & Abraham, A. Normalized square difference based multilevel thresholding technique for multispectral images using leader slime mould algorithm. *J. King Saud Univ. Sci.* **34**, 4524–4536. <https://doi.org/10.1016/j.jksuci.2020.10.030> (2022).
67. Hu, G., Zheng, Y., Abualigah, L. & Hussien, A. G. DETDO: An adaptive hybrid dandelion optimizer for engineering optimization. *Adv. Eng. Inf.* **57**, 102004. <https://doi.org/10.1016/j.aei.2023.102004> (2023).
68. Hu, G., Chen, L. & Wei, G. Enhanced golden jackal optimizer-based shape optimization of complex CSGC-Ball surfaces. *Artif. Intell. Rev.* **56**, 2407–2475 (2023). <https://doi.org/10.1007/s10462-023-10581-6>
69. Sultan, H. M., Menesy, A. S., Hassan, M., Jurado, F. & Kamel, S. 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> (2023).
70. Menesy, A. S., Sultan, H. M., Selim, A., Ashmawy, M. G. & Kamel, S. Developing and applying chaotic Harris Hawks optimization technique for extracting parameters of several proton exchange membrane fuel cell stacks. *IEEE Access* **8**, 1146–1159 (2020). <https://doi.org/10.1109/ACCESS.2019.2961811>
71. Yongguang, C. & Guanglei, Z. 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> (2022).
72. Hasanien, H. M., Shaheen, M. A., Turkey, R. A., Qais, M. H., Alghuwainem, S., Kamel, S., & Jurado, F. Precise modeling of PEM fuel cell using a novel enhanced transient search optimization algorithm. *Energy*, **247**, 123530 (2022).
73. Kanouni, B. & Laib, A. Extracting accurate parameters from a proton exchange membrane fuel cell model using the differential evolution ameliorated meta-heuristics algorithm. *Energies* **17**(10), 2333 (2024).
74. Saad, B., El-Sehiemy, R. A., Hasanien, H. M. & El-Dabah, M. A. Robust parameter estimation of proton exchange membrane fuel cell using Huber loss statistical function. *Energy Convers. Manag.* **323**, 119231 (2025).
75. Jangir, P. et al. Precision parameter estimation in Proton Exchange Membrane Fuel Cells using depth information enhanced Differential Evolution. *Sci. Rep.* **14**(1), 29591 (2024).

76. Rowe, A. & Li, X. Mathematical modeling of proton exchange membrane fuel cells. *J. Power Sources* **102**(1–2), 82–96. [https://doi.org/10.1016/S0378-7753\(01\)00798-4](https://doi.org/10.1016/S0378-7753(01)00798-4) (2001).
77. Correa, J. M., Farret, F. A., Canha, L. N. & Simoes, M. G. An electrochemical-based fuel-cell model suitable for electrical engineering automation approach. *IEEE Trans. Ind. Electron.* **51**(5), 1103–1112. <https://doi.org/10.1109/TIE.2004.834972> (2004).
78. Aouali, F. Z. et al. Analytical modelling and experimental validation of proton exchange membrane electrolyser for hydrogen production. *Int. J. Hydrogen Energy* **42**(2), 1366–1374. <https://doi.org/10.1016/j.ijhydene.2016.03.101> (2017).
79. Amphlett, J. C., Baumert, R. M., Mann, R. F., Peppley, B. A., Roberge, P. R. & Harris, T. J. Performance modeling of the Ballard Mark IV solid polymer electrolyte fuel cell: I. Mechanistic model development. *J. Electrochem. Soc.* **142**(1), 1–8 (1995). <https://doi.org/10.1149/1.2043866>
80. Mann, R. F. et al. Development and application of a generalised steady state electrochemical model for a PEM fuel cell. *J. Power Sources* **86**(1–2), 173–180. [https://doi.org/10.1016/S0378-7753\(99\)00484-X](https://doi.org/10.1016/S0378-7753(99)00484-X) (2000).
81. Amphlett, J. C., Baumert, R., Mann, R. F., Peppley, B. A., Roberge, P. R. & Harris, T. J. Performance modeling of the Ballard Mark IV solid polymer electrolyte fuel cell: II. Empirical model development. *J. Electrochem. Soc.* **142**(1), 9–15 (1995). <https://doi.org/10.1149/1.2043959>
82. Mo, Z.-J., Zhu, X.-J., Wei, L.-Y. & Cao, G.-Y. Parameter optimization for a PEMFC model with a hybrid genetic algorithm. *Int. J. Energy Res.* **30**, 585–597. <https://doi.org/10.1002/er.1170> (2006).
83. Alatas, B. Chaotic harmony search algorithms. *Appl. Math. Comput.* **216**(3), 2687–2699. <https://doi.org/10.1016/j.amc.2010.03.114> (2010).
84. Nguyen, T. V. & White, R. E. A water and heat management model for proton-exchange-membrane fuel cells. *J. Electrochem. Soc.* **140**(8), 2178–2186. <https://doi.org/10.1149/1.2220792> (1993).
85. El-Fergany, A. A. 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> (2017).
86. Ali, M., El-Hameed, M. A. & Farahat, M. A. Effective parameters identification for polymer electrolyte membrane fuel cell models using grey wolf optimizer. *Renew. Energy* **111**, 455–462. <https://doi.org/10.1016/j.renene.2017.04.036> (2017).
87. Kennedy, J. & Eberhart, R. Particle swarm optimization. *Proceedings of ICNN'95—International Conference on Neural Networks* 1995, p. 1942–8 vol.4. <https://doi.org/10.1109/ICNN.1995.488968>
88. Hu, G., Du, B., Li, H. & Wang, X. Quadratic interpolation boosted black widow spider-inspired optimization algorithm with wavelet mutation. *Math. Comput. Simul.* **200**, 428–467. <https://doi.org/10.1016/j.matcom.2022.04.031> (2022).
89. M. Dehghani, Z. Montazeri, E. Trojovska, P. Trojovský, Coati Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems, *Knowl Based Syst.* 259 (2023) 110011. <https://doi.org/10.1016/j.knsys.2022.110011>
90. Li, S., Chen, H., Wang, M., Heidari, A. A. & Mirjalili, S. Slime mould algorithm: A new method for stochastic optimization. *Future Gener. Comput. Syst.* **111**, 300–323. <https://doi.org/10.1016/j.future.2020.03.055> (2020).
91. Solis, F. J. & Roger, J. B. W. Minimization by random search techniques. *Math. Oper. Res.* **6**, 19–30. <https://doi.org/10.1287/moor.6.1.19> (1981).

Acknowledgements

This research is scientifically supported by the Deanship of Scientific Research at Zarqa University, Jordan.

Author contributions

M.A., S.P.A., and A.P. conceptualized the study and designed the research framework. P.J., A., and B.I.T. contributed to data collection and preprocessing. G.G. and A.F.A. developed the methodology and conducted the experiments. R.J. and M.K. performed the data analysis and validation. M.A. and G.G. wrote the main manuscript text, while A.F.A. prepared figures and visual representations. All authors reviewed, revised, and approved the final manuscript.

Funding

The Researchers would like to thank the Deanship of Graduate Studies and Scientific Research at Qassim University for financial support (QU-APC-2025).

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1038/s41598-025-92528-1>.

Correspondence and requests for materials should be addressed to M.A., G.G., A.F.A. or M.K.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025