

Mutation-enhanced Puma Optimizer for Lithium-ion Battery Parameter Optimization

En-Jui Liu,* Yan-Hao Huang, Qing-An Wang, Wan-Ling Lu, and Rou-Wen Chen

Department of Green Energy and Information Technology, National Taitung University,
No. 369, Sec. 2, University Road, Taitung City, Taitung County 950309, Taiwan

(Received March 27, 2026; accepted September 1, 2026)

Keywords: lithium-ion battery, uniform puma optimizer, mutation mechanism, parameter optimization, hybrid pulse power characterization

Lithium-ion batteries offer high energy density and low self-discharge, making them a key energy storage system for transportation electrification. However, the coupled electrochemical dynamics of lithium-ion batteries result in nonlinear characteristics and a multimodal optimization landscape, making accurate parameter identification challenging. To address the limitation, we propose a novel uniform puma optimizer (UPO) for battery model parameter optimization. UPO incorporates a uniform mutation strategy into the puma optimizer (PO) to maintain population diversity and enhance global search capability. In the identification stage, the hybrid pulse power characterization test was conducted to establish an accurate relationship between open-circuit voltage and the state of charge, while UPO was employed to identify battery model parameters. In the validation stage, capacity tests at 0.5 and 1 C are performed to evaluate the model accuracy. The results show that the UPO-based model achieves mean absolute error values of 26.93 and 44.01 mV at 0.5 and 1 C, respectively, representing reductions of 13.96 and 15.04% compared with the PO-based model. These results confirm that UPO provides an effective and robust parameter identification method, supporting the development of accurate battery models for advanced battery management systems.

1. Introduction

Lithium-ion batteries play a central role in global 2050 net-zero strategies because their high energy density, low self-discharge and zero operational emissions enable large-scale electrification. Their deployment has expanded across electric vehicles, renewable energy storage systems, and portable devices. Recent advances in IoT, embedded systems, and lightweight intelligent technologies have facilitated real-time sensing, intelligent monitoring, and energy-efficient operation in various applications.⁽¹⁻³⁾ These technologies facilitate the integration of battery systems with distributed sensors, communication networks, and intelligent monitoring platforms for real-time data acquisition, condition assessment, and energy management. However, electrochemical reactions exhibit strong sensitivity to temperature, load

*Corresponding author: e-mail: ejliu@nttu.edu.tw
<https://doi.org/10.18494/SAM6354>

variation, and environmental changes, causing impedance, polarization, and open-circuit voltage to vary across usage scenarios. These dynamics introduce nonlinear and multimodal system characteristics that complicate the development of an accurate battery model.⁽⁴⁾ Battery modeling can be grouped into three major approaches, including electrochemical models, equivalent circuit models (ECMs), and data-driven models.⁽⁵⁾ Electrochemical models describe ion transport, charge transfer, and concentration dynamics inside the electrodes and electrolyte. Common structures include the pseudo two-dimension (P2D) model⁽⁶⁾ and the single-particle model (SPM).⁽⁷⁾ These models provide high fidelity but require substantial computational resources and detailed material parameters and the complexity limits real-time use. ECMs describe internal processes with resistors and capacitors that represent ohmic resistance, polarization and diffusion effects.⁽⁸⁾ Typical structures include the Rint model, the partnership for a new generation of vehicles (PNGV) model,⁽⁹⁾ the Thevenin model, and higher order resistance-capacitor (RC) networks.⁽¹⁰⁾ ECMs offer clear physical meaning and low computational cost, but parameter values shift with the operating conditions and aging, and the accuracy drops under dynamic load. Data-driven models (DDMs) learn battery behavior directly from measured data through machine learning⁽¹¹⁾ or neural networks,⁽¹²⁾ and these methods capture complex nonlinear behavior but require large datasets and show limited generalization outside the training range.

The need for reliable real-time state estimation in electric vehicles has driven the substantial development of ECMs,⁽¹³⁾ while the high-dimensional optimization characteristics of lithium-ion batteries align well with metaheuristic search methods.⁽¹⁴⁾ Metaheuristic approaches provide flexible search strategies for high-dimensional optimization and avoid the constraints of gradient-based methods.⁽¹⁵⁾ These approaches are aimed at exploring the search space through randomized movement and evaluating candidates using objective functions. Swarm intelligence forms a major branch of metaheuristics and builds the search process on simple interactions among agents. The collective behavior guides the population toward promising regions and the structure supports strong global search under nonlinear and multimodal conditions.⁽¹⁶⁾

According to the no free lunch (NFL) theorem, no algorithm can maintain superior performance across all optimization problems.⁽¹⁷⁾ In recent studies, therefore, swarm intelligence algorithms have been adopted to address these challenges. Techniques such as particle swarm optimization,⁽¹⁸⁾ grey wolf optimizer,⁽¹⁹⁾ and the artificial hummingbird algorithm⁽²⁰⁾ have been widely applied to improve parameter estimation accuracy and robustness in lithium-ion battery parameter optimization. Various strategies have also been introduced to enhance the search capability of optimization algorithms, including opposition-based learning,⁽²¹⁾ fuzzy logic,⁽²²⁾ and chaotic maps.⁽²³⁾

The puma optimizer (PO) is a metaheuristic algorithm that exhibits rapid convergence and effective search performance through its phase-transition mechanism. However, insufficient population diversity may cause PO to converge prematurely to local optima. To address this limitation, in this study, we introduce a coin-toss-guided uniform mutation mechanism that generates multiple candidate solutions, maintains population diversity, and enhances the global exploration capability of PO.

For battery parameter identification, a suitable battery model and representative experimental data are required. Therefore, a hybrid pulse power characterization (HPPC) test is conducted in the first phase to capture the battery dynamics under controlled load conditions. The measured HPPC data are then used by the uniform puma optimizer (UPO) to identify the parameters of an RC ECM. The identified parameters are incorporated into the battery model, and the simulated terminal voltage is compared with constant-current discharge data to evaluate the accuracy and reliability of the proposed method. The overall research framework is illustrated in Fig. 1.

The main contributions of this study are summarized as follows.

- A uniform mutation strategy is incorporated into PO to enhance global exploration and reduce the risk of premature convergence.
- A parameter optimization and validation framework are established using HPPC and capacity tests to evaluate parameter accuracy under representative operating conditions.
- UPO is applied to optimize the parameters of a lithium-ion battery model, providing more stable convergence and a better balance between exploration and exploitation.
- Model validation results demonstrate that the parameters identified by UPO enable more accurate battery performance prediction across the entire state-of-charge (SOC) range under different C-rates.

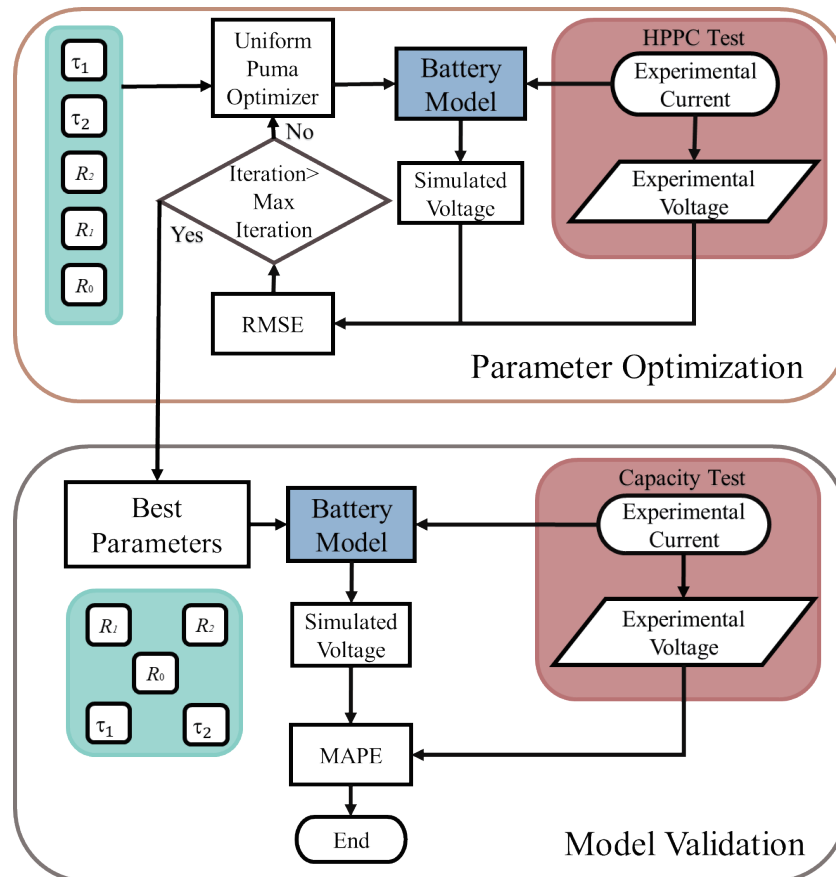


Fig. 1. (Color online) Overall research framework.

2. Methodology

2.1 Battery modeling and parameter optimization framework

In this study, we employ a second-order RC ECM to characterize the charge and discharge behavior of the lithium-ion battery. The two-RC model is widely used because it provides a suitable balance between modeling accuracy and computational cost.⁽²⁴⁾ The formulation of the model is given in Eqs. (1)–(4). The equivalent circuit structure is illustrated in Fig. 2.

$$V = V_{ocv} - IR_0 - V_1 - V_2 \quad (1)$$

$$V_1 = \frac{I}{C_1} - \frac{V_1}{R_1 C_1} \quad (2)$$

$$V_2 = \frac{I}{C_2} - \frac{V_2}{R_2 C_2} \quad (3)$$

$$\tau = RC \quad (4)$$

The term V_{OCV} denotes the open-circuit voltage. The resistor R_0 represents the internal resistance and reflects the resistive effect of the electrolyte and conductors. The resistors R_1 and R_2 represent polarization resistances and correspond to the voltage drop produced by charge-transfer reactions and mass-transport limitations. The capacitors C_1 and C_2 represent polarization capacitances, while τ denotes the polarization time constant. The term V denotes the terminal voltage of the circuit and represents the battery output.

The hybrid pulse power characterization (HPPC) test was conducted to generate the experimental dataset for battery model parameter optimization. HPPC is a standardized procedure used to evaluate the performance of high-power lithium-ion batteries at different

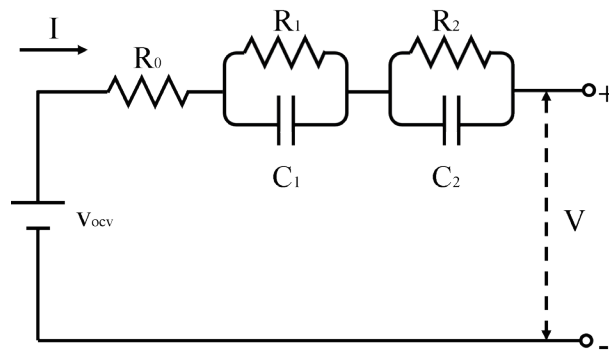


Fig. 2. Structure of the battery model.

states of charge. The experiments were performed using a prismatic Delta P 140-185 cell, as shown in Fig. 3(a).

In the test, the cell response at the controlled SOC was measured, providing voltage responses under short charge and discharge pulses. The HPPC test procedure enables the open-circuit voltage to be determined at different SOC levels. Accurate OCV characterization is essential for determining the initial battery state because an inaccurate OCV may introduce polarization-related bias and lead to an incorrect assessment of the battery operating condition. The relaxation period following each pulse represents the voltage recovery associated with activation and concentration polarization after the instantaneous ohmic voltage drop. It also reflects changes in relaxation behavior at different SOC levels. These measurements were used to establish the OCV–SOC relationship and provide the dynamic information required for parameter identification. The HPPC test profile is shown in Fig. 3(b).

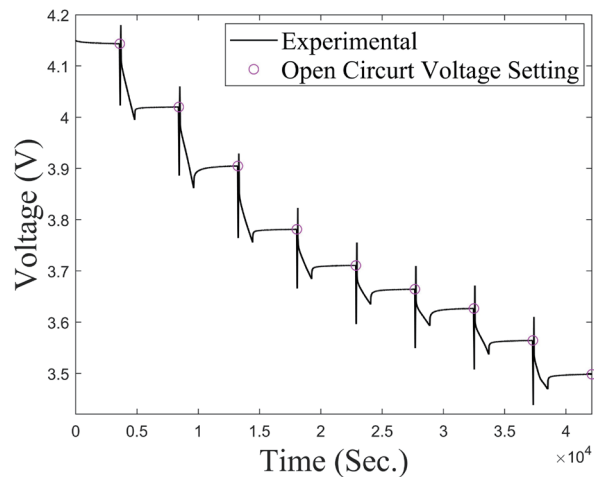
In the validation stage, constant-current capacity tests were conducted to evaluate the identified battery model. An accurate model should reproduce the battery characteristics under different operating conditions. The battery was discharged at constant rates of 0.5 C and 1 C, which are commonly used in energy storage applications.

The root mean square error (*RMSE*) is used as the objective function for parameter optimization. As defined in Eq. (5), *RMSE* quantifies the difference between the measured and simulated terminal voltages. Through iterative optimization, the algorithm identifies the parameter set that minimizes *RMSE*.

$$RMSE = \sqrt{\frac{1}{N} \sum_{t=1}^N (V_{cal} - V_{exp})^2} \quad (5)$$



(a)



(b)

Fig. 3. (Color online) (a) Experimental battery and (b) HPPC test procedure and OCV setting points.

N denotes the number of experimental data points, where V_{cal} and V_{exp} denote the simulated and measured voltages, respectively. Table 1 shows the settings of the optimization, and all tests are performed in MATLAB R2024b on a system equipped with an Intel i9-13950HX CPU and 48 GB of RAM.

In model validation, the mean absolute error (MAE) and mean absolute percentage error ($MAPE$) are used to evaluate and compare voltage prediction accuracy. MAE quantifies the average absolute difference between the simulated and measured terminal voltages, whereas $MAPE$ expresses this difference as a percentage and reflects the relative model accuracy across different voltage levels.

$$MAE = \frac{1}{N} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (6)$$

$$MAPE = \left[\frac{1}{N} \sum_{i=1}^n |y_i - \hat{y}_i| \right] \times 100\% \quad (7)$$

In Eqs. (6) and (7), y_i denotes the measured value, $\hat{y}$ denotes the predicted value, and n denotes the number of samples.

2.1 PO

PO is a swarm intelligence algorithm proposed by Abdollahzadeh *et al.* and was inspired by the hunting behavior of pumas in natural environments.⁽²⁵⁾ The algorithm employs distinct exploration and exploitation mechanisms to improve search efficiency and solution accuracy. The exploitation phase simulates the running and ambush behaviors of pumas, whereas the exploration phase represents their movement toward new regions or other individuals in search of prey.

During the optimization process, PO switches between the exploration and exploitation phases on the basis of a score-based evaluation mechanism. The detailed transition procedure is provided in Appendix, and the corresponding search mechanisms are described below.

$$Z_{i,G} = \begin{cases} R_{Dim} \times (UB - LB) + LB, & \text{if } rand_1 > 0.5 \\ X_{a,G} + G \cdot (X_{a,G} - X_{b,G}) + G \cdot \left((X_{a,G} - X_{b,G}) - (X_{c,G} - X_{d,G}) + (X_{c,G} - X_{d,G}) - (X_{e,G} - X_{f,G}) \right) \end{cases} \quad (8)$$

Table 1
Algorithm settings.

Runtime	Agent	Iteration
30	30	1000

$$G = 2 \cdot rand_2 - 1 \quad (9)$$

$$X_{new} = \begin{cases} Z_{i,G}, & \text{if } j = f_{rand} \\ X_{a,G}, & \text{otherwise} \end{cases} \quad (10)$$

Equation (8) uses $rand_1$ and R_{Dim} as random numbers in the interval [0,1]. UB and LB denote the upper and lower bounds of the search range, respectively. The exploitation phase is inspired by the running and ambush behaviors of pumas. The algorithm generates new candidate solutions using two distinct schemes, as mathematically formulated in Eq. (11).

$$X_{new} = \begin{cases} \frac{\left(\frac{mean(Sol_{total})}{Npop} \right) \cdot X_1^r - (-1)^\beta \times X_i}{1 + (\alpha \cdot rand_5)}, & \text{if } rand_4 \geq 0.5 \\ Puma_{male} + (2 \cdot rand_7) \cdot \exp(randn_1) \cdot (X_2^r - X_i), & \text{otherwise, if } rand_6 \geq L \\ \frac{(F_1 \cdot R \cdot X(i) + F_2 \cdot (1 - R) \cdot Puma_{male})}{(2 \cdot rand_9 - 1 + randn_2)} - Puma_{male}, & \text{otherwise} \end{cases} \quad (11)$$

$$X_2^r = round\left(1 + (Npop - 1) \cdot rand_{10}\right) \quad (12)$$

$$R = 2 \cdot rand_{11} - 1 \quad (13)$$

$$F_1 = randn_3 \cdot \exp\left(2 - Iter \cdot \left(\frac{2}{MaxIter}\right)\right) \quad (14)$$

$$F_2 = w \times (randn_5)^2 \cdot \cos((2 \times rand_{12}) \cdot w) \quad (15)$$

$$w = randn_4 \quad (16)$$

$rand_4, rand_5, rand_6, rand_7, rand_8$, and $rand_9$ are uniform random numbers within [0,1]. $randn_1, randn_2, randn_3, randn_4$, and $randn_5$ are normally distributed random numbers. $mean$ represents the average fitness of the population and Sol_{total} denotes the sum of all solution values. N_{pop} is the size of the puma population. X_1^r is a randomly selected solution from the entire population. β is a randomly generated integer equal to either 0 or 1, and α and L are constants set to 2 and 0.67, respectively, in accordance with the default configuration of PO. $Puma_{male}$ denotes the best

solution in the population. exp represents the exponential function. X_2^r is a randomly selected solution based on Eq. (12) and X_i is the previous solution.

F_1 and F_2 are computed using Eqs. (14) and (15), respectively. $Iter$ is the number of current iterations and Max_{Iter} is the maximum iteration count. Equation (16) defines the computation of w in F_2 . cos denotes the cosine function.

2.2 Mutation mechanism for enhancing algorithm diversity

PO exhibits fast convergence and satisfactory accuracy in engineering optimization problems. However, its search process may stagnate in local optima when solving multimodal and strongly coupled nonlinear problems. This limitation is mainly attributed to insufficient exploration during the early search stage. To address this issue, we introduce a uniform mutation strategy into the original PO framework. The mutation mechanism perturbs selected individuals to generate new candidate solutions, thereby increasing population diversity, expanding the effective search space, and reducing the risk of premature convergence. The main characteristics of UPO and PO are presented in Table 2.

The proposed strategy generates additional promising candidate solutions within the search space, allowing the algorithm to evaluate a broader range of parameter configurations during the

```
Pseudocode of the uniform mutation strategy
Sort ascending population
Calculate p
for (each Puma ( $X_i$ )) do
    Select six solutions randomly
    Generate the exploration solution  $Y_i$  by Eq. (8)
    Check the boundary of  $Y_i$ 
    Generate a random Toss vector
    Generate  $N_{nx}$  by Eq. (17)
    Generate  $N_{ny}$  by Eq. (18)
    Calculate Cost ( $N_{nx}$ )
    Calculate Cost ( $N_{ny}$ )
    Calculate Cost ( $X_i$ )
end
Merge {  $X_i$ ,  $N_{nx}$ ,  $N_{ny}$  }
Sort all candidate solutions
Keep the best  $N$  solutions
```

Table 2
Main characteristics of UPO and PO.

Characteristic	UPO	PO
Search ability	Enhanced exploration through diversified candidate solutions	Higher risk of becoming trapped in local optima
Convergence stability	More stable and less sensitive to initialization	More sensitive to the initial population
Computational cost	Slightly higher due to the additional mutation operation	Baseline computational cost

exploration phase. This mechanism helps the algorithm escape local optima and improves its ability to approach the global optimum. The improved algorithmic structure is illustrated in Fig. 4, and the uniform mutation procedure is mathematically formulated in Eqs. (17)–(20).

$$X_{nx}(j) = \begin{cases} x(j), & \text{if } Toss(j) == 0 \\ y(j), & \text{else} \end{cases} \quad (17)$$

$$X_{ny}(j) = \begin{cases} y(j), & \text{if } Toss(j) == 0 \\ x(j), & \text{else} \end{cases} \quad (18)$$

$$X_n = \begin{cases} X_{nx}, & \text{if } CostX_{nx} < CostX_{ny} \\ X_{ny}, & \text{else} \end{cases} \quad (19)$$

$$X_{best} = \begin{cases} X_n, & \text{if } CostX_n < CostX_i \\ X_i, & \text{Otherwise} \end{cases} \quad (20)$$

Fitness evaluation is used to identify high-quality candidate solutions. As shown in Eqs. (17) and (18), the algorithm first generates two candidate solutions, X_{nx} and X_{ny} represent their corresponding objective-function values. The solution with the lower cost is selected as the intermediate solution X_n .

In accordance with Eq. (19), X_n is then compared with the current solution X_i . Their corresponding costs, $CostX_{nx}$ and $CostX_{ny}$, are evaluated, and the solution with the lower cost is retained as X_{best} , with the corresponding cost denoted by $CostX_{best}$. The flowchart of UPO is shown in Fig. 5.

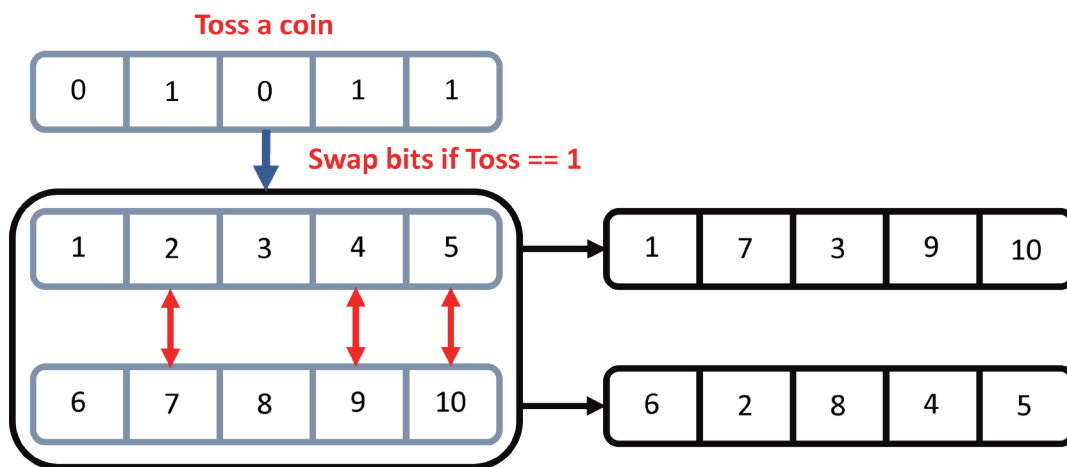


Fig. 4. (Color online) Uniform mutation mechanism based on the toss chain.

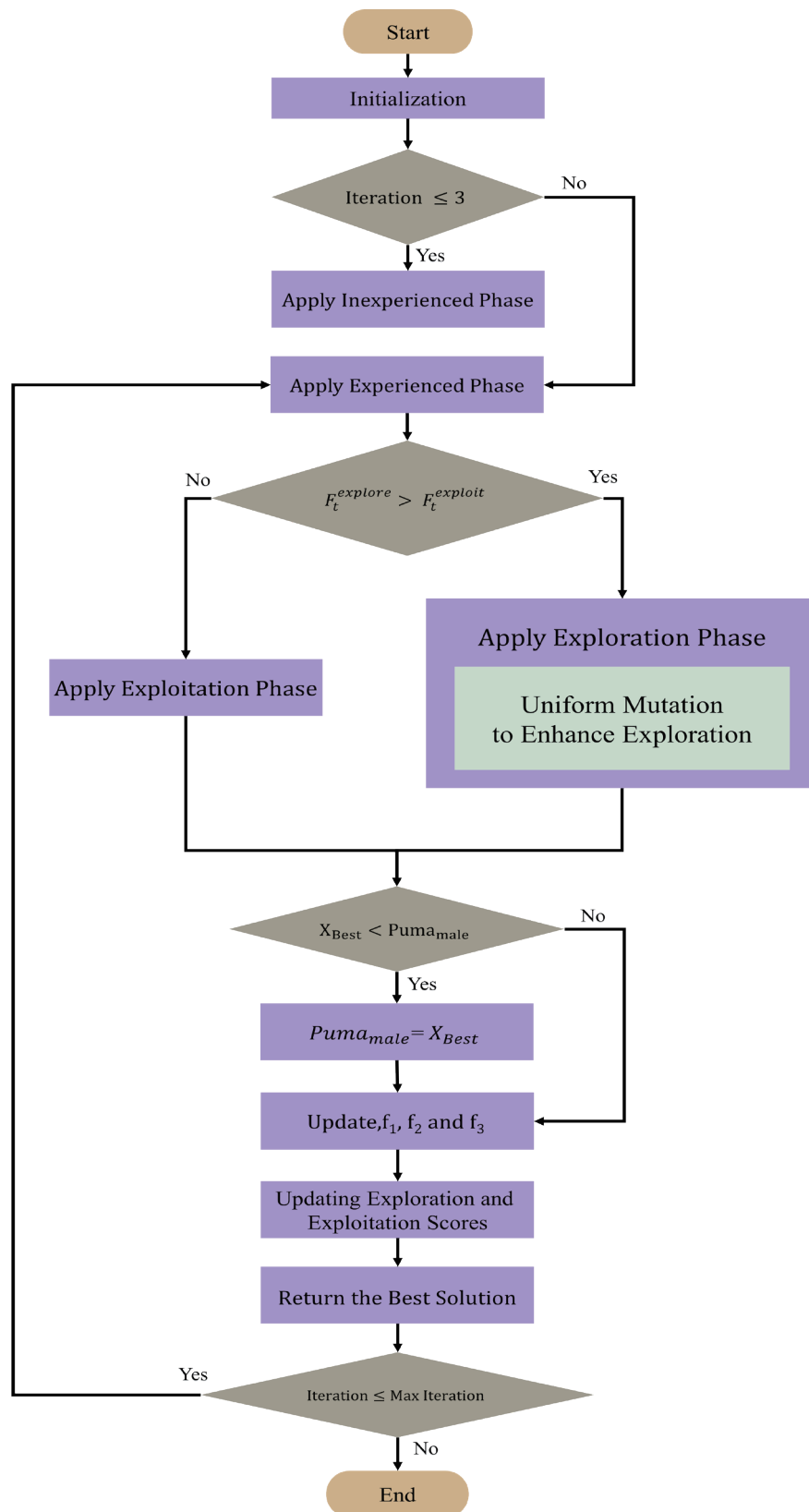


Fig. 5. (Color online) Flowchart of UPO.

3. Results and Discussion

3.1 Parameter optimization

The robustness of UPO is evaluated by comparing its best, mean, and standard deviation of the fitness values with those obtained by five benchmark algorithms (honey badger algorithm (HBA), modified honey badger algorithm (MHBA), moth-flame optimization (MFO), whale optimization algorithm (WOA), and gazelle optimization algorithm (GOA)).⁽²⁶⁾ The robustness results are reported in Table 3. The optimized parameter sets are presented in Fig. 6, while the convergence curves in Fig. 7 show the rapid and stable convergence of UPO.

The optimization process evaluates each parameter set using *RMSE*. UPO converges more rapidly toward low-error regions of the search space. Its robustness is evaluated over multiple

Table 3
Parameter optimization results for UPO and PO.

	Min (mV)	Mean (mV)	Std. (mV)
UPO	1.397	1.542	0.085
PO	1.532	1.798	0.124
HBA ⁽²⁶⁾	1.553	2.347	0.413
MHBA ⁽²⁶⁾	1.512	2.127	0.529
MFO ⁽²⁶⁾	2.473	3.058	0.476
WOA ⁽²⁶⁾	2.473	3.058	0.317
GOA ⁽²⁶⁾	2.851	3.380	0.333

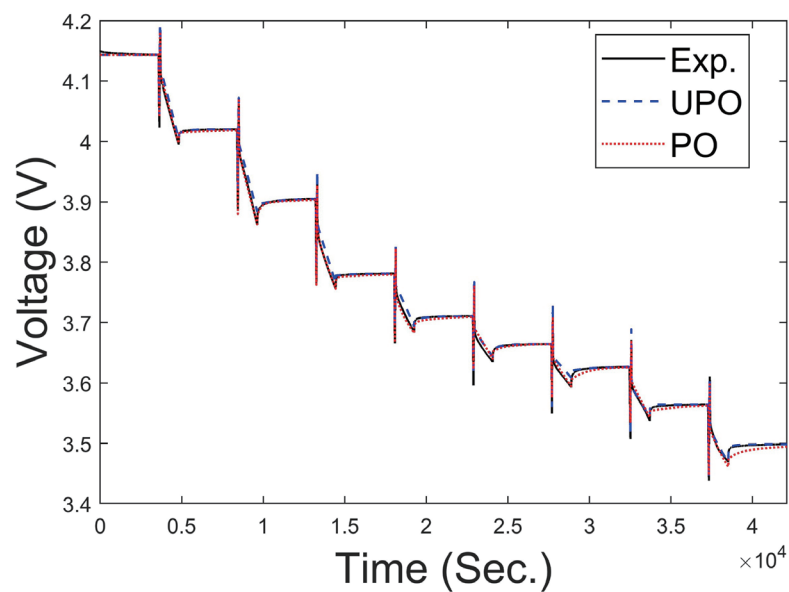


Fig. 6. (Color online) Experimental voltage and UPO and PO optimization results.

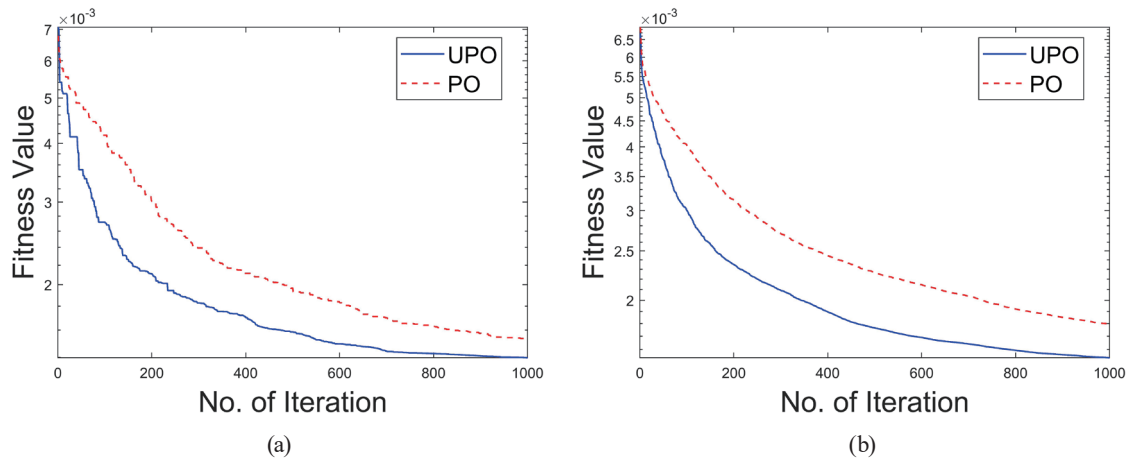


Fig. 7. (Color online) (a) Best and (b) mean convergence performances of parameter optimization.

independent runs and compared with PO and five benchmark algorithms, and achieves the lowest best and mean fitness values, together with the smallest standard deviation, demonstrating higher optimization accuracy and more stable performance across repeated runs. These results indicate that the uniform mutation strategy enhances global search capability and improves the reliability of battery parameter identification. Figure 8 shows the optimal parameter estimates obtained by UPO and PO at different SOC levels.

The mutation strategy effectively enhances the global search capability. Figure 9 illustrates the search modes selected by UPO and PO during each iteration, where Phase 1 represents exploration and Phase 2 represents exploitation. The corresponding exploration-to-exploitation ratios are summarized in Table 4. The higher exploration ratio of UPO demonstrates that the proposed strategy strengthens the global search capability of the algorithm.

3.2 Model validation

The constant-current discharge test is a standard method for evaluating the capacity and voltage response of lithium-ion batteries. To verify the accuracy of the identified model parameters, we used the discharge curve as the experimental reference for model validation. Discharge rates of 0.5 and 1 C were selected because these current levels provide stable voltage responses that reflect the combined effects of ohmic resistance and polarization. In addition, the complete discharge profiles capture the overall dynamic and polarization behavior of the battery, providing a more rigorous approach for validating the accuracy of the identified battery model. The corresponding results are presented in Figs. 10(a) and 10(b) and summarized in Table 5.

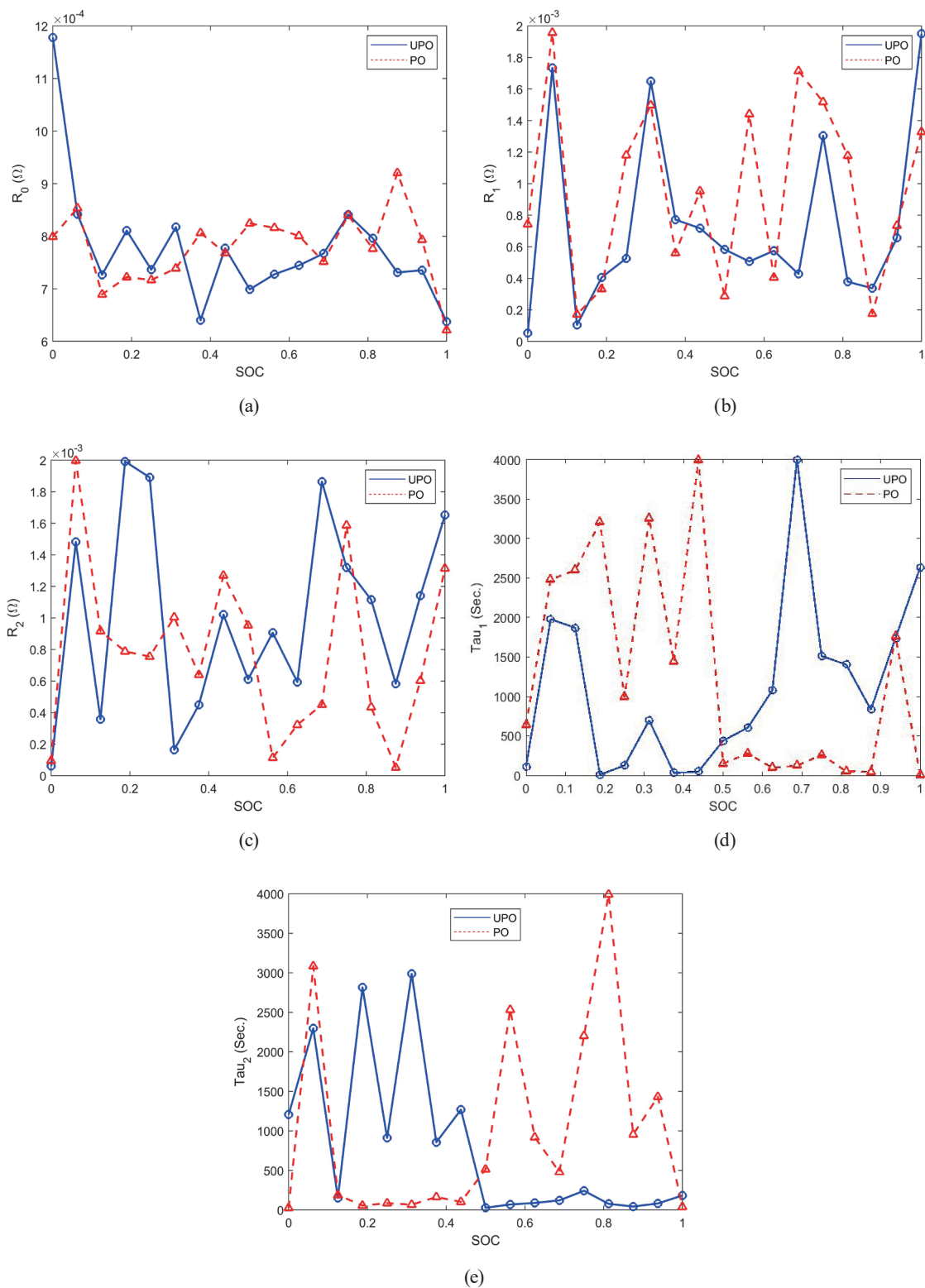


Fig. 8. (Color online) Best battery parameters at different SOC levels.

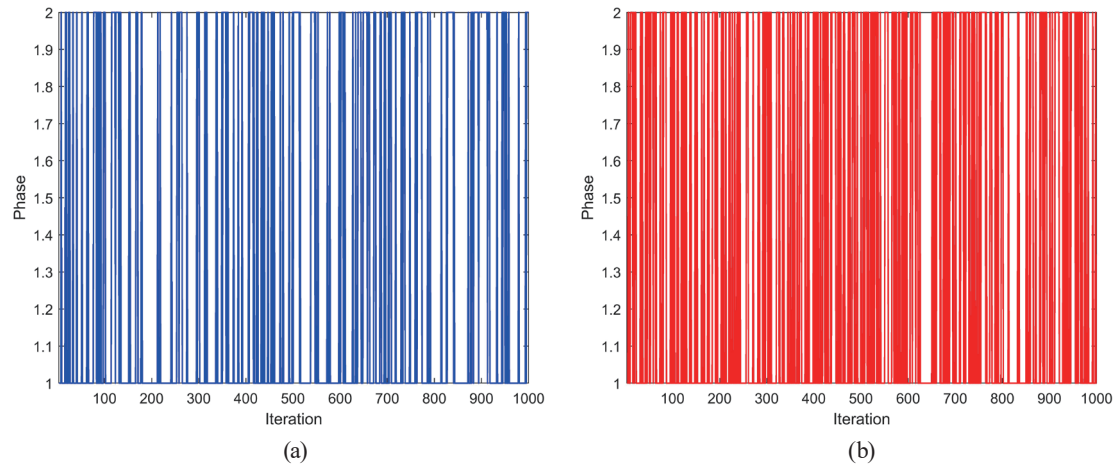


Fig. 9. (Color online) (a) UPO and (b) PO search phase transition processes.

Table 4
UPO and PO transition ratios.

	Exploration (%)	Exploitation (%)
UPO	79.3586	20.6414
PO	64.0923	35.9077

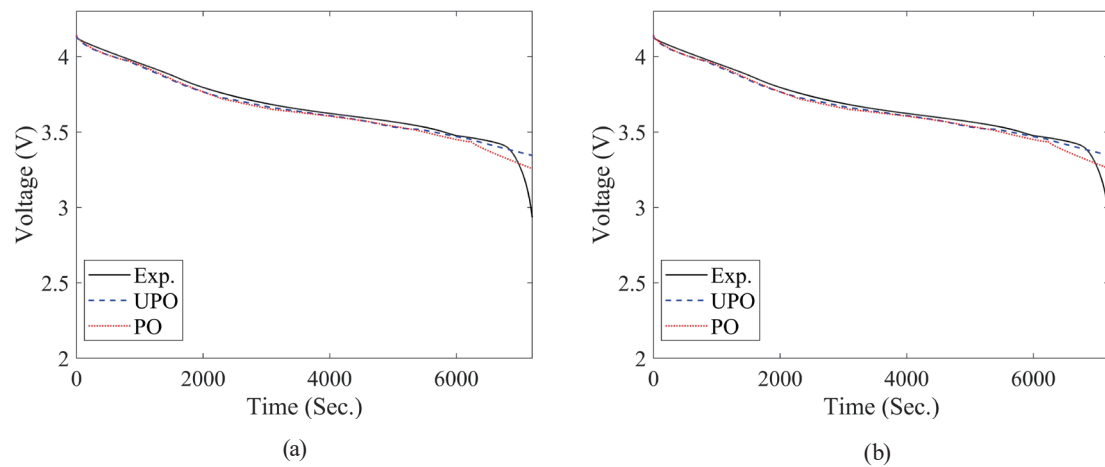


Fig. 10. (Color online) Results of validation tests at discharges of (a) 0.5 and (b) 1 C for UPO and PO with optimal parameters.

Table 5
Results for models under validation conditions.

C-rate	Method	MAE (mV)	MAPE (%)	Improvement (%)
0.5 C	UPO	26.93	0.7614	13.96
	PO	31.30	0.8810	–
1 C	UPO	44.01	1.2240	15.04
	PO	51.80	1.4444	–

4. Conclusions

In this study, we developed UPO for lithium-ion battery parameter identification by incorporating a coin-toss-guided uniform mutation strategy into the exploration phase of PO. In PO, new solutions during the exploitation phase are generated on the basis of the current population, the population mean, and the best-so-far solution, referred to as the Puma male. As the population gradually converges, these reference solutions become increasingly similar, thereby reducing the search distance and concentrating the candidate solutions around the current best region. In addition, the greedy replacement mechanism accepts a new solution only when it improves the current fitness, making it difficult for the algorithm to accept any temporary deterioration that may be necessary to escape from a local optimum.

The proposed mechanism generated additional candidate solutions, maintained population diversity, and reduced the risk of premature convergence. Statistical comparisons with PO and five benchmark algorithms demonstrated that UPO achieved better best and mean fitness values as well as a lower standard deviation, indicating improved optimization accuracy and stability. The exploration ratio increased from 64.0923% for PO to 79.3586% for UPO. This result confirms that the uniform mutation strategy strengthened global exploration while maintaining effective convergence.

The parameters identified from the HPPC data were incorporated into a two-RC ECM and validated through constant-current discharge tests at 0.5 and 1 C. Compared with the PO-based model, the UPO-based model achieved lower errors under both discharge conditions. The results demonstrate that UPO improves the reliability of battery parameter identification and provides an accurate modeling basis for SOC and SOH estimation and energy management. Future work will include extending the method to battery state estimation and energy system integration.

Acknowledgments

This work was supported by the National Science and Technology Council under Grant No. 114-2221-E-143-005. The authors sincerely thank the editor and reviewers for their constructive comments and careful review.

References

- 1 C.-H. Hsia, S.-C. Yen, and J.-H. Jang: *Sens. Mater.* **31** (2019) 1803. <https://doi.org/10.18494/SAM.2019.2351>
- 2 L.-Y. Ke and C.-H. Hsia: *Sens. Mater.* **36** (2024) 945. <https://doi.org/10.18494/SAM4858>
- 3 C.-H. Hsia and C.-F. Lai: *Sens. Mater.* **32** (2020) 3221. <https://doi.org/10.18494/SAM.2020.2861>
- 4 L. Zhou, X. Lai, B. Li, Y. Yao, M. Yuan, J. Weng, and Y. Zheng: *Batteries* **9** (2023) 131. <https://doi.org/10.3390/batteries9020131>
- 5 W. Liu, J. Teh, and J. Shi: *Battery-Integrated Residential Energy Systems*, A. Berrada, A. Q. H. Badar, M. Sanjari, Eds. (CRC Press, Boca Raton, 2024) 1st ed., Chap. 3. <https://doi.org/10.1201/9781003441236>
- 6 A. Jokar, B. Rajabloo, M. Désilets, and M. Lacroix: *J. Power Sources* **327** (2016) 44. <https://doi.org/10.1016/j.jpowsour.2016.07.036>
- 7 S. J. Moura, F. B. Argomodo, R. Klein, A. Mirtabatabaei, and M. Krstic: *IEEE Trans. Control Systems Technology* **25** (2016) 453. <https://doi.org/10.1109/TCST.2016.2571663>

- 8 M.-K. Tran, A. DaCosta, A. Mevawalla, S. Panchal, and M. Fowler: Batteries **7** (2021) 51. <https://doi.org/10.3390/batteries7030051>
- 9 P. Lin, P. Jin, A. Zou, and Z. Wang: Int. J. Energy Res. **45** (2021) 9351. <https://doi.org/10.1002/er.6465>
- 10 A. S. Pande, B. P. Soni, and K. V. Bhadane: J. Inst. Eng. India Ser. B **104** (2023) 533. <https://doi.org/10.1007/s40031-022-00852-1>
- 11 H. Tu, S. Moura, Y. Wang, and H. Fang: Appl. Energy **329** (2023) 120289. <https://doi.org/10.1016/j.apenergy.2022.120289>
- 12 H. Benhammou, K. Anoune, and A. Tajmouati: Discover Electron. **2** (2025) 70. <https://doi.org/10.1007/s44291-025-00106-6>
- 13 C.-H. Wu and W.-Z. Gao: Energy Convers. Manage.: X **27** (2025) 101132. <https://doi.org/10.1016/j.ecmx.2025.101132>
- 14 L. Zhang, H. Peng, Z. Ning, Z. Mu, and C. Sun: Appl. Sci. **7** (2017) 1002. <https://doi.org/10.3390/app7101002>
- 15 K. Rajwar, K. Deep, and S. Das: Artif. Intell. Rev. **56** (2023) 13187. <https://doi.org/10.1007/s10462-023-10470-y>
- 16 A. Nayyar and N. G. Nguyen: Introduction to Swarm Intelligence: In Advances in Swarm Intelligence for Optimizing Problems in Computer Science (Chapman and Hall/CRC, 2018) p. 53. <https://doi.org/10.1201/9780429445927>
- 17 D. H. Wolpert and W. G. Macready: IEEE Trans. Evol. Comput. **1** (2002) 67. <http://doi.org/10.1109/4235.585893>
- 18 Z. Yu, L. Xiao, H. Li, X. Zhu, and R. Huai: IEEE Trans. Ind. Electron. **64** (2017) 5690. <http://doi.org/10.1109/TIE.2017.2677319>
- 19 G.-J. Chen, Y.-H. Liu, S.-C. Wang, Y.-F. Luo, and Z.-Z. Yang: J. Energy Storage **33** (2021) 101933. <https://doi.org/10.1016/j.est.2020.101933>
- 20 M. A. Hamida, R. A. El-Sehiemy, A. R. Ginidi, E. Elattar, and A. M. Shaheen: J. Energy Storage **51** (2022) 104535. <https://doi.org/10.1016/j.est.2022.104535>
- 21 N. Rojas-Morales, M.-C. R. Rojas, and E. M. Ureta: Comput. Ind. Eng. **110** (2017) 424. <https://doi.org/10.1016/j.cie.2017.06.028>
- 22 F. Valdez, P. Melin, and O. Castillo: Expert Syst. Appl. **41** (2014) 6459. <http://doi.org/10.1016/j.eswa.2014.04.015>
- 23 A. Limane, F. Zitouni, S. Harous, R. Lakbichi, A. Ferhat, A. S. Almazyad, P. Jangir, and A. W. Mohamed: Complex Intell. Syst. **11** (2025) 177. <https://doi.org/10.1007/s40747-025-01791-2>
- 24 M. A. Mohamed, T. F. Yu, G. Ramsden, J. Marco, and T. R. Grandjean: J. Energy Storage **113** (2025) 115581. <https://doi.org/10.1016/j.est.2025.115581>
- 25 B. Abdollahzadeh, N. Khodadadi, S. Barshandeh, P. Trojovský, F. S. Gharehchopogh, E.-S. M. El-kenawy, L. Abualigah, and S. Mirjalili: Cluster Comput. **27** (2024) 5235. <https://doi.org/10.1007/s10586-023-04221-5>
- 26 W.-L. Yu, E.-J. Liu, and J.-Y. Chang: Int. J. Electr. Power Energy Syst. **174** (2026) 111566. <http://doi.org/10.1016/j.ijepes.2026.111566>

Appendix

This Appendix presents the mathematical formulation of PO and describes its distinctive phase-transition mechanism in detail.

$$A = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1D} \\ x_{21} & x_{22} & \cdots & x_{2D} \\ & \cdots & \cdots & \\ x_{n1} & x_{n2} & \cdots & x_{nD} \end{bmatrix} \quad (\text{A1})$$

Equation (A1) defines the puma population A . X_i denotes the position of the i -th puma in the search space. n denotes the number of pumas. x denotes a position vector in the D -dimensional search space. D denotes the problem dimension.

$$x_i = LB_i + r_1 \times (UB_i - LB_i) \quad (\text{A2})$$

Equation (A2) uses r_1 as a random number in the interval $[0,1]$. The term generates values within the upper bound (UB) and lower bound (LB) of the search range. The initial phase of the algorithm reflects the stage in which pumas have limited knowledge of the environment and no information about prey locations. The process executes exploration and exploitation at the same time to identify suitable hunting regions. The algorithm performs both tasks during the first three iterations.

$$f1_{mode} = PF_1 \times \left(\frac{Seq_{mode}^1}{Seq_{Time}} \right) \quad (\text{A3})$$

$$f2_{mode} = PF_2 \times \left(\frac{Seq_{mode}^1 + Seq_{mode}^2 + Seq_{mode}^3}{Seq_{Time}^1 + Seq_{Time}^2 + Seq_{Time}^3} \right) \quad (\text{A4})$$

Equations (A3) and (A4) define the computation of $f1_{mode}$ and $f2_{mode}$, which denote the values obtained in the exploration and exploitation modes, respectively. The variable Seq_{mode}^1 denotes the sequence value produced by each mode in iteration t , and the mathematical expressions appear in Eqs. (A5)–(A7). The value of Seq_{Time} gives the number of selections for each mode.

$$Seq_{mode}^1 = \left| Cost_{mode}^1 - Cost_{Best}^{Initial} \right| \quad (\text{A5})$$

$$Seq_{mode}^2 = \left| Cost_{mode}^2 - Cost_{mode}^1 \right| \quad (\text{A6})$$

$$Seq_{mode}^3 = \left| Cost_{mode}^3 - Cost_{mode}^2 \right| \quad (A7)$$

The term $Cost_{mode}$ denotes the best solution in the population during the current iteration.

$$Score_{mode} = (PF_1 \times f1_{mode}) + (PF_2 \times f2_{mode}) \quad (A8)$$

Equation (A8) computes the final scores of exploration and exploitation in the inexperienced phase. The constants P_{F1} and P_{F2} are set to 0.5. The values of $f1_{mode}$ and $f2_{mode}$ are obtained from Eqs. (A3) and (A4) and represent the numerical results of the exploration and exploitation modes, respectively. If $Score_{Explore}$ is greater than $Score_{Exploit}$, the algorithm selects exploration in the experienced phase. Otherwise, the algorithm switches to exploitation.

After the first three iterations, the search mode is selected in accordance with the performance accumulated during the inexperience phase. Each subsequent iteration executes either exploration or exploitation and the mode switches at every step. This stage is defined as the experienced phase.

During the experienced phase, the algorithm selects the operating mode through the scoring functions f_{1t}^{mode} , f_{2t}^{mode} and f_{3t}^{mode} . The computation of f_{1t}^{mode} and f_{2t}^{mode} differs from the definitions used in the inexperienced phase. The function f_{3t}^{mode} acts as an escape mechanism that prevents the algorithm from remaining in a single mode for too many iterations. The three functions combine the historical performance of exploration and exploitation and support the selection of the most effective mode for the next iteration.

$$f_{1t}^{mode} = PF_1 \times \left| \frac{Cost_{new}^{mode} - Cost_{old}^{mode}}{T_t^{mode}} \right| \quad (A9)$$

The term f_{1t}^{mode} denotes the computation used for the two operating modes. The constant P_{F1} is set to 0.5. $Cost$ denotes the best fitness value obtained in the past or in the current execution of the mode. The value of T_t^{mode} is the number of iterations since the mode was last selected. The computation of f_{2t}^{mode} is as given in Eq. (A10).

$$f_{2t}^{mode} = PF_2 \times \left| \frac{S_{t,1}^{mode} + S_{t,2}^{mode} + S_{t,3}^{mode}}{T_{t,1}^{mode} + T_{t,2}^{mode} + T_{t,3}^{mode}} \right| \quad (A10)$$

The term f_{2t}^{mode} follows the same definition as $f1$ and the constant P_{F2} follows the same definition as P_{F1} . T_t represents the number of iterations in which the mode has not been selected since its last use. S is defined in Eqs. (A11)–(A13).

$$S_{t,1}^{mode} = Cost_{New,1}^{mode} - Cost_{Old,1}^{mode} \quad (A11)$$

$$S_{t,2}^{mode} = Cost_{New,2}^{mode} - Cost_{Old,2}^{mode} \quad (A12)$$

$$S_{t,3}^{mode} = Cost_{New,3}^{mode} - Cost_{Old,3}^{mode} \quad (A13)$$

The term $Cost^{mode}$ is defined as the best fitness value obtained by the puma in the selected mode. The computation of f_{3t}^{mode} is as given in Eq. (A14).

$$f_{3t}^{mode} = \begin{cases} \text{if selected, } f_{3t}^{mode} = 0 \\ \text{otherwise, } f_{3t}^{mode} + PF_3 \end{cases} \quad (A14)$$

Equation (A14) defines the computation of f_3 . The constant P_{F3} is set to the default value of 0.3. The term P_{F3} controls the probability of selecting the mode with a lower score. A nonselected mode increments P_{F3} and the selected mode resets P_{F3} to zero. The function prevents the algorithm from remaining in a single mode and falling into a local optimum. Equation (A15) defines the score computation for the exploration and exploitation modes.

$$F_t^{mode} = (\alpha_t^{mode} \cdot f_{1t}^{mode}) + (\alpha_t^{mode} \cdot f_{2t}^{mode}) + (\delta_t^{mode} \cdot lc \cdot f_{3t}^{mode}) \quad (A15)$$

F_t^{mode} denotes the score of the exploration and exploitation modes in iteration t . The definitions of lc , α , and δ are as in Eqs. (A16)–(A19).

$$lc = \begin{cases} \{ |Cost_{New} - Cost_{New}| \}^{exploitation} \\ \{ |Cost_{New} - Cost_{New}| \}^{exploitation} \end{cases} \quad (A16)$$

$$\alpha_t^{explore, exploit} = \begin{cases} \alpha^{exploit} = 0.99 \\ \alpha^{explore} = [\alpha^{explore} - 0.01, 0.01] \\ \text{if } F^{exploit} > F^{explore} \\ \alpha^{explore} = 0.99 \\ \alpha^{exploit} = [\alpha^{exploit} - 0.01, 0.01] \end{cases} \quad (A17)$$

$$\delta_t^{exploit} = 1 - \alpha_t^{exploit} \quad (A18)$$

$$\delta_t^{explore} = 1 - \alpha_t^{explore} \quad (A19)$$

The algorithm adjusts the weights of f_{1t}^{mode} and f_{2t}^{mode} in accordance with the scores obtained in the exploration and exploitation modes. When the score of exploitation exceeds the score of exploration, the algorithm increases the effect of exploitation by reducing the weight of exploration through a linear update of α . When the score of exploration exceeds that of exploitation, the adjustment is made in the opposite direction. The value δ varies inversely with α . The algorithm maintains the balance between exploration and exploitation, and the structure prevents excessive global search that produces inaccurate solutions and reduces the risk of early convergence to local optima.