以元啟發式演算法實現氫能燃料電池之參數最佳化

鄭瑞鈞 、劉恩睿*

摘要

質子交換膜燃料電池（Proton exchange membrane fuel cell, PEMFC）憑藉低噪音、無污染、壽命長等優勢，已成為我國推動淨零轉型的關鍵能源技術之一。然而，燃料電池的運作機制使其放電特性呈現高度非線性，這使得精確計算燃料電池的效能表現成為一大挑戰。為克服此困難點，本研究以基於美洲獅獵食行為的元啟發式演算法，針對兩種類型的質子交換膜燃料電池系統進行參數最佳化，並將其結果與五種新型元啟發式演算法進行比較。結果顯示，美洲獅演算法在兩種類型燃料電池系統的參數最佳化過程中，在最佳適應值以及平均適應值皆為最佳值。這一結果表明美洲獅演算法在處理多維度非線性問題方面具備極高的適用性，能夠精確地估算系統參數，從而準確預測燃料電池的最佳功率點，進一步提升系統的效能與穩定性，為推動國家氫能技術的落地實現提供重要助力。

關鍵字：氫能、質子交換膜燃料電池、元啟發式演算法、美洲獅演算法、參數最佳化

鄭瑞鈞，國立臺東大學綠能與資訊科技碩士學位學程學，Email: 11202205@gm.nttu.edu.tw
劉恩睿(通訊作者)，國立臺東大學綠能與資訊科技學系助理教授，Email: ejliu@nttu.edu.tw

Parameter Optimization of Hydrogen Fuel Cells Using Metaheuristic Algorithm

Jui-Chun Cheng , En-Jui Liu*

Abstract

Proton exchange membrane fuel cell (PEMFC), known for its low noise, zero emissions, and long lifespan, has become a key energy technology in the nation's push towards net-zero emissions. However, the nonlinear discharge characteristics of fuel cells present significant challenges in accurately predicting their performance. To address this, the study utilizes a metaheuristic algorithm inspired by the hunting strategy of pumas to optimize the parameters of two distinct PEMFC systems. The performance of the puma optimizer was compared with five other advanced metaheuristic algorithms. The results show that the puma optimizer outperformed the others, delivering superior results for both optimal and average fitness in parameter optimization across both fuel cell systems. These findings demonstrate the puma optimizer's strong ability to solve complex, multidimensional nonlinear problems, enabling precise estimation of system parameters and accurate prediction of the optimal power output. This method not only improves the performance and stability of PEMFC systems but also offers substantial support for advancing national hydrogen energy technologies.

Keywords: Hydrogen energy, Proton exchange membrane fuel cell, Metaheuristic algorithm, Puma optimizer, Parameter optimization

Jui-Chun Cheng, student, Master Program of Green Energy and Information Technology, National Taitung University. Email: 11202205@gm.nttu.edu.tw

En-Jui Liu (Corresponding Author), Assistant Professor, Master Program of Interdisciplinary Program of Green and Information Technology, National Taitung University. Email: ejliu@nttu.edu.tw

壹、前言

現如今全球環保課題趨勢朝向淨零排放以及淨零轉型，將溫室氣體排放量盡力降至零，且其餘排放量能夠讓自然環境重新吸收，若能在 2050 年前達成淨零，全球升溫幅度就不會超過 1.5°C，淨零轉型以「能源轉型」、「產業轉型」、「生活轉型」、「社會轉型」為主要方向來達成目標(方俊德，2023)，因此氫能與燃料電池技術發展與產業化也應該備受重視、積極發展。燃料電池是一種將化學能轉化為電能的可再生能源，質子交換膜燃料電池(Proton exchange membrane fuel cell, PEMFC)是其中一種常見的燃料電池類型，它更加堅固、成本更低(Turgut et al.,2016)。PEMFC 早已應用在許多地方，如運輸、攜帶型發電以及便攜式電子產品的電池，燃料電池 (FC)可將電化學能轉化為電能，它被公認為一種替代的乾淨能源 (Yan et al.,2019)，因此，它也可以更好得應用於家庭利用、工業發電站和運輸應用和快速發展中。質子交換膜是一種高效能的薄膜，一開始用於燃料電池和其他能源技術中，用「綠氫」產生乾淨的能源，在燃料電池中，質子交換膜有著關鍵的作用，它將氫氣中的質子與氧氣中的電子進行反應後，產生電能、水和熱能，能夠在高溫和高壓下長時間穩定運行 (Dannenberg et al.,2000)，質子交換膜還有著高效的催化反應、能量轉換效率高、環保、能在相對低溫的環境下操作等優點，只要使用氫氣搭配空氣中的氧氣，就可以運用在各種地方，並且只會排出無害的高純度水(Gamburzev et al.,2002)，這讓 PEM 成為燃料電池技術中非常關鍵的部分，也被廣泛應用於其他能源技術之中。

近年來，燃料電池的參數估計是人們競爭研究的方向，質子交換膜燃料電池被認為是能將化學能轉化為電能最有前途的裝置之一(Zhao et al.,2020)。而用於 PEMFC 參數估計的模型是 Mann 模型，這是一個具有七個未知參數的半經驗模型 (Mann et al.,2000)，因 PEMFC 為非線性的函數，很難進行簡單、有效又準確的預測，所以提取 PEMFC 的參數變得非常必要，目前有許多解析電池特徵的方式，如元啟發式優化演算法或傳統啟發式演算法等，然而啟發式演算法有許多缺點，例如它們依賴於問題的初始條件，而被困在局部解的可能等。許多研究人員開始使用元啟發式演算法，因為它們較無上述問題限制，且可以用於各種實際工程問題(Selem et al.,2020)，利用多種元啟發式演算法來獲取不確定參數的最優值，如 grey wolf optimization (GWO)、Manta rays foraging optimizer (MRFO)等演算法(Riad et al.,2023)，目的在計算數據表中 PEMFC 模型不確定參數的最佳近似解，實驗之電化學半經驗參數會直接影響 PEMFC 的數學建模，可使用演算法來計算最佳適應度函數值(Best Fitness)來決定參數值。本研究使用美洲獅最佳化演算法(Puma Optimizer)來估算 PEMFC 參數的最佳近似值，燃料電池模型選用 NedStack PS6 和 BCS 500W 兩種電池模組，並將 PO 演算法應用於計算兩個電池模組的實驗電壓和估計電壓之間差值誤差平方和(Sum-Square-Error, SSE)和標準差 (Standard Deviation, Std.)以及其他參數，並與上述提到的演算法和其他先進演算法進行誤差比對，來驗證 PO 演算法與不同演算法的計算結果。

貳、文獻探討

PEMFC 可應用於許多地方，例如微熱電聯產(μCHP)應用，Budak 提出在實驗中使用 μCHP 後，HT-PEMFC 以及 LT-PEMFC 兩種系統的效率都有所提高 (Budak & Devrim,2018)，HT-PEMFC 效率提高了 48-92%，LT-PEMFC 效率提高了 48-78%，HT-PEMFC 具有較低的功率密度，但當與 μCHP 系統結合時，可以提高效率。微電網應

用，Gong et al 等人為質子交換膜燃料電池開發了一個模型，以檢索運行時產生的不需要的熱能，利用燃料電池的空置容量來生產、儲存氫氣備用且提高其效率，並結合他們所提出的一種用於智慧混合 AC-DC 微電網最佳運行的架構，使用驗算法來解決混合微電網中的最佳功率調度 (Gong et al.,2020)。和家庭應用方面，Bizon 等人使用 real-time optimization (RTO)演算法找到用於住宅建築的燃料電池混合動力系統(FCHPS)燃料流量的最佳值 (Bizon et al.,2018)。

Mann 等人提出的廣義穩態電化學模型 GSSEM 比早期的 SSEM 模型更具廣泛的適用性，此模型能夠以更理想的方式處理 Nafion 膜的質子交換膜 (PEM)燃料電池，並且可在相對較高的電流密度下應用，GSSEM 的實用性在於能夠靈活應用於各種操作條件，通過調整係數 λ 的公式來計算膜電阻，應足以模擬所有使用 Nafion 膜的電池(Mann et al.,2000)。

有關演算法的準確程度與效率，Kriegel 在研究中內文提到關於演算法效率評估的建議 (Kriegel et al.,2017)，經過他們使用不同演算法運算一些相同的工程問題，或是運算一些相同的數據集的實驗後，他們發現演算法的準確程度不能單憑計算時間以及運算過程來直接判斷，且每個實驗都有其局限性，無法適配於所有問題，而研究人員的一些無意識的假設可能已經對結果產生了相當大的影響，為進一步解釋演算法的探索機制，本研究將探索階段與開發階段的執行次數分別計算出來，觀察美洲獅演算法中探索與開發階段之間的比重。

在 Hasanien 等人的實驗中，對 TSO 演算法進行了增強，以有效地利用新開發的 ETSO 演算法提取 PEMFC 的不確定參數，實驗結果證明 ETSO 擁有優於 TSO、HHO、SSO 等的性能 (Hasanien et al.,2022)。本實驗使用 PO 演算法應用於計算 NedStack 與 BCS 500 W 兩種電池的參數，與其他演算法進行比較，證明 PO 的性能與用於燃料電池的可行性。

參、研究方法

一、質子交換膜燃料電池建模

質子交換膜燃料電池 (PEMFC)中含有陰極和陽極兩個電極，由帶有聚合物電解質的膜隔離開來，如**圖 1** 所示，陽極注入氫氣，陰極注入氧氣，薄膜形成電解質，電解質傳導離子並阻止電子通過，當離子流過電解質時，會產生輸出電壓，而外部電路允許離子通過。

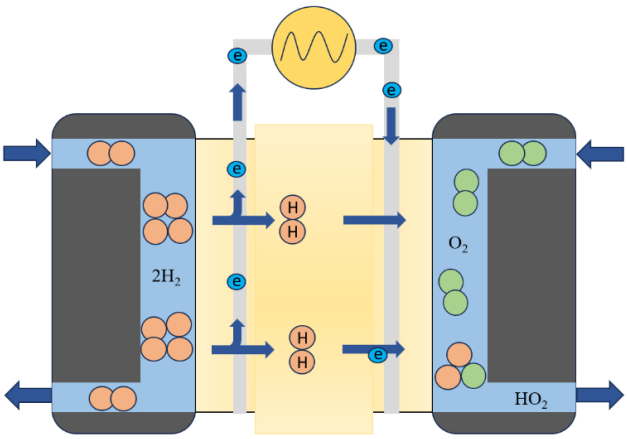

圖 1. PEMFC 示意圖

陽極 (Anode)：

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

陰極 (Cathode)：

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

化學反應式：

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

質子交換膜燃料電池 V_{cell} 的輸出電壓是由可逆開路電壓 E_{Nernst} 和轉換過程中的三種損耗所產生 (Duan et al.,2023)：

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

在實際系統中，通常將多個電池堆疊在一起，產生的電堆壓 V_{stack} 可以計算為:

$$V_{stack} = n.V_{cell} = n.\ (E_{Nernst} - V_{act} - V_{ohm} - V_{con}) \tag{5}$$

其中 n 為電池的數量，可逆開路電壓 E_{Nernst} 取自方程式 (6)，它取決於氫氣 (P_{H_2}) 和氧氣 (P_{O_2})的分壓以及溫度 (T in Kelvin):

$$E_{Nernest} = 1.229 - 8.5 \times 10^{-4}\ (T - 298.15) + 4.3085 \times 10^{-5}\ T\ ln[P_{H_2} + 0.5(P_{O_2})] \tag{6}$$

氫氣 (P_{H_2})和氧氣 (P_{O_2})的分壓計算方法如下方程式(7 和 8) (Ayyarao et al.,2024)：

$$P_{H_2} = \frac{R_{Ha}P_{H_2O}}{2}\left[\left(exp\left(\frac{1.635(\frac{I}{A})}{T^{1.334}}\right)\frac{R_{Ha}P_{H_2O}}{P_a}\right)^{-1} - 1\right] \tag{7}$$

$$P_{O_2} = R_{Hc}P_{H_2O}\left[\left(exp\left(\frac{4.192(\frac{I}{A})}{T^{1.334}}\right)\frac{R_{Hc}P_{H_2O}}{P_c}\right)^{-1} - 1\right] \tag{8}$$

$$log_{10}(P_{H_2O}) = 2.95 \times 10^{-2}(T - 273.15) - 9.18 \times 10^{-5}(T - 273.15)^2 + 1.44 \times 10^{-7}(T - 273.15)^3 - 2.18 \tag{9}$$

活化損耗 (V_{act})的計算方法如下：

$$V_{act} = -\ [\xi 1 + \xi 2T + \xi 3Tln\ (C_{O_2}) + \xi 4Tln\ (I_{cell})] \tag{10}$$

其中 I_{cell} 是燃料電池當前的電流值，$\xi 1$、$\xi 2$、$\xi 3$、$\xi 4$ 表示不確定參數值，C_{O_2}表示在陰極溶解氧的濃度 (mol/cm3)定義如下:

$$C_{O_2} = P_{O_2}/\ (5.08 \times 10^6 \times e^{-\left(\frac{498}{T}\right)}) \tag{11}$$

歐姆損耗 (V_{ohm})是由電阻 Rm 和 Rc 兩端的電壓 V_{ohm} 使用公式（12）計算：

$$V_{ohm} = I_{cell} \times (R_m + R_c) \tag{12}$$

其中 R_c 表示連接電阻，R_m 表示膜電阻，定義如下:

$$R_m = (\rho_m . l)/A \tag{13}$$

$$\rho_m = \frac{181.6\left[1 + 0.03\left(\frac{I_{cell}}{A}\right) + 0.062\left(\frac{T}{303}\right)^2\left(\frac{I_{cell}}{A}\right)^{2.5}\right]}{\left[\lambda - 0.634 - 3\left(\frac{I_{cell}}{A}\right)\right]exp\left[4.18\left(T - \frac{303}{T}\right)\right]} \tag{14}$$

其中ρ_m表示膜的電阻 (Ω.cm)，l 表示膜的厚度 (cm)，A 為膜的表面積 (cm2)，λ 為其中一個不確定參數。

濃度損耗 V_{con} 值使用方程式 (15) 計算 (Rezk et al.,2022)：

$$V_{con} = -\, b\ln\left(1 - (I_{cell}/A)/I_{max}\right) \tag{15}$$

其中 b 為不確定參數，I_{max} 表示當前最大電流密度 (A/cm^2)。

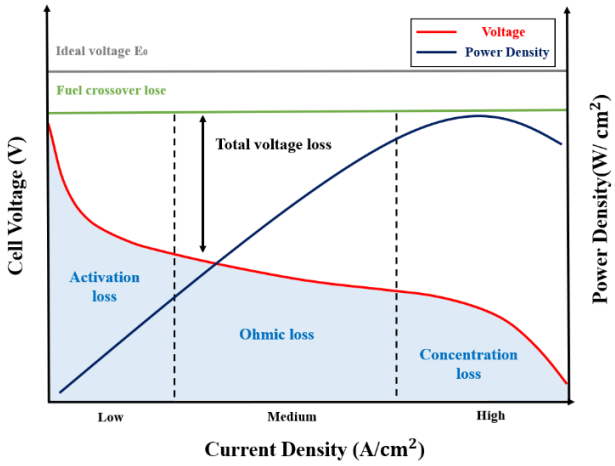

圖 2. 燃料電池內部極化作用。

二、美洲獅演算法

美洲獅演算法(Puma Optimizer, PO)是利用自然界中美洲獅的狩獵行為概念來創建一個數學模型 (Abdollahzadeh et al.,2024)，PO 演算法使用了一種創新的機制來改變探索和開發階段，這種階段變化機制為首次提出。PO 主要分成四個階段：

(一) 無經驗階段：

PO 演算法在無經驗階段的時候會先迭代三次來取得在前三次反覆運算中，探索和開發同時進行，直到完成初始化，由於每次反覆運算中都有探索和開發階段，因此使用了兩個函數 (*f1* 和 *f2*)，計算方式如下：

$$f1_{Explor} = PF_1 \times \left(\frac{Seq^1_{CostExplor}}{Seq_{Time}}\right) \tag{16}$$

$$f1_{Exploit} = PF_1 \times \left(\frac{Seq^1_{CostExploit}}{Seq_{Time}}\right) \tag{17}$$

$$f2_{Explor} = PF_2 \times \left(\frac{Seq^1_{CostExplor} + Seq^2_{CostExplor} + Seq^3_{CostExplor}}{Seq^1_{Time} + Seq^2_{Time} + Seq^3_{Time}}\right) \tag{18}$$

$$f2_{Exploit} = PF_2 \times \left(\frac{Seq^1_{CostExploit} + Seq^2_{CostExploit} + Seq^3_{CostExploit}}{Seq^1_{Time} + Seq^2_{Time} + Seq^3_{Time}}\right) \tag{19}$$

Seq_{Cost} 的變數值與探索和開發階段有關，由公式(16–19)計算，而 Seq_{Time} 是一個具有常數值的變數，將其視為 1。PF_1 和 PF_2 都是具有固定值的參數，且在整個過程之前設置，用來確定 *f1* 和 *f2* 兩個函數的優先順序。

$$Seq^1_{CostExplor} = \left|Cost^{Initial}_{Best} - Cost^1_{Explor}\right| \tag{20}$$

$$Seq^{2}_{CostExplor} = \left|Cost^{2}_{Explor} - Cost^{1}_{Explor}\right| \tag{21}$$
$$Seq^{3}_{CostExplor} = \left|Cost^{3}_{Explor} - Cost^{2}_{Explor}\right| \tag{22}$$
$$Seq^{1}_{CostExploit} = \left|Cost^{Initial}_{Best} - Cost^{1}_{Exploit}\right| \tag{23}$$
$$Seq^{2}_{CostExploit} = \left|Cost^{2}_{Exploit} - Cost^{1}_{Exploit}\right| \tag{24}$$
$$Seq^{3}_{CostExploit} = \left|Cost^{3}_{Exploit} - Cost^{2}_{Exploit}\right| \tag{25}$$

在方程式(20 和 23)中，初始化階段產生七個變數$Cost^{Initial}_{Best}$、$Cost^{1}_{Explor}$、$Cost^{2}_{Explor}$、$Cost^{3}_{Explor}$、$Cost^{1}_{Exploit}$、$Cost^{2}_{Exploit}$、$Cost^{3}_{Exploit}$，而其中的$Cost^{Initial}_{Best}$為最好的解，每一個 Cost 的最佳解來自探索和開發階段重複 1、2 和 3 在第三次反覆運算結束時計算函數 *f1* 和 *f2* 後，從這裡開始只選擇探索或開發階段其一，開發階段和探索階段的分數是使用方程 (26 和 27)計算：

$$Score_{Explor} = \left(PF_1 \times f1_{Explor}\right) + \left(PF_2 \times f2_{Explor}\right) \tag{26}$$
$$Score_{Exploit} = \left(PF_1 \times f1_{Exploit}\right) + \left(PF_2 \times f2_{Exploit}\right) \tag{27}$$

在計算 $Score_{Explor}$ 和 $Score_{Exploit}$ 之後，根據分數大小來決定進入探索還是開發階段，如果 $Score_{Exploit} \geq Score_{Explor}$ 進入開發階段，反之則進入探索階段，但是在第 3 次反覆運算結束時，每個階段都會有一個獨立的解決方案，而這會比總人口數還要多，為了解決這個問題，在這兩個階段所生成的解決方案的成本是在第三次反覆運算的最後階段才計算的，並且只有新的最佳解小於舊的解時才可替換當前的解決方案。

(二) 有經驗階段：

在經過三代之後，美洲獅有了足夠的經驗來決定改變階段，並且在反覆運算中，他們只選擇一個階段進行最佳化操作。在此階段，使用 *f1*、*f2* 和 *f3* 三個不同的函數進行評分，第一個函數偏向於探索階段，主要用來決定探索和開發的兩個階段的優先順序，第一個函數使用方程式 (28 和 29)計算。

$$f1^{exploit}_{t} = PF_1 \times \left|\frac{Cost^{exploit}_{old} - Cost^{exploit}_{new}}{T^{exploit}_{t}}\right| \tag{28}$$
$$f1^{explor}_{t} = PF_1 \times \left|\frac{Cost^{explor}_{old} - Cost^{explor}_{new}}{T^{explor}_{t}}\right| \tag{29}$$

在方程式(28 和 29)中，$f1^{exploit}_{t}$和$f1^{explor}_{t}$表示於開發或探索階段的第一函數的數量，其中 t 表示當前的反覆運算數。$Cost^{explor}_{old}$和$Cost^{exploit}_{old}$是最佳解改進之前的成本，而另一方面，$Cost^{explor}_{new}$和$Cost^{exploit}_{new}$是最佳解改進之後的成本。T^{explor}_{t}和 $T^{exploit}_{t}$是從前一個選定內容到當前選定內容的反覆運算次數。PF_1 是可調整的參數，在操作之前必須將其值設置為 0 到 1 之間，此參數確定第一個函數的重要性，並且隨著此函數的值的增加或減少而確定其優先順序。第二個函數導致階段執行比其他優先的階段更好，且可以幫助其選擇中的階段，使用方程式 (30 和 31)計算：

$$f2^{exploit}_{t} = PF_2 \times \left|\frac{\left(Cost^{exploit}_{old,1} - Cost^{exploit}_{new,1}\right) + \left(Cost^{exploit}_{old,2} - Cost^{exploit}_{new,2}\right) + \left(Cost^{exploit}_{old,3} - Cost^{exploit}_{new,3}\right)}{T^{exploit}_{t,1} + T^{exploit}_{t,2} + T^{exploit}_{t,3}}\right| \tag{30}$$

$$f2_t^{explor} = PF_2 \times \left| \frac{\left(Cost_{old,1}^{explor} - Cost_{new,1}^{explor}\right) + \left(Cost_{old,2}^{explor} - Cost_{new,2}^{explor}\right) + \left(Cost_{old,3}^{explor} - Cost_{new,3}^{explor}\right)}{T_{t,1}^{explor} + T_{t,2}^{explor} + T_{t,3}^{explor}} \right| \quad (31)$$

在方程式(30 和 31)中，$f2_t^{exploit}$和$f2_t^{explor}$代表與探索或開發階段相關的第二個函數，其中 t 代表當前的反覆運算數，$Cost_{old}$ 為探索和開發階段的最佳解決方案改進前的成本，$Cost_{new}$ 探索和開發階段改進後獲得的最佳解決方案的成本，也就是無經驗階段中的三次運算分別的成本。T 為探索和開發階段三次運算之間從前一個到當前的反覆運算次數。PF_2 是一個參數，在操作之前，須將其值設置為介於 0 和 1 之間，此參數用來確定第二個函數的權重，當此函數的值增加時，它將被優先考慮，而隨著其優先順序的降低而減少。

第三個函數確保在多次重複選擇中未被選中的階段也有機會被選中，因為只選擇一個階段會導致陷入到局部最優的陷阱中，該函數用方程式 (32 和 33)計算：

$$f3_t^{exploit} = \begin{cases} if\ \ selected, f3_t^{exploit} = 0 \\ otherwise, f3_t^{exploit} + PF_3 \end{cases} \quad (32)$$

$$f3_t^{explor} = \begin{cases} if\ \ selected, f3_t^{explor} = 0 \\ otherwise, f3_t^{explor} + PF_3 \end{cases} \quad (33)$$

在方程式(32 和 33)中，$f3_t^{exploit}$和 $f3_t^{explor}$代表與開發和探索階段的第三個函數，其中 t 代表當前反覆運算數。如果未選擇其中一個階段，則每個探索和開發階段的第三個函數的值將在每次反覆運算中按參數 PF_3 增加，否則將設置為零。PF_3 參數是使用者可調整的參數，在操作之前須設置其值介於 0 和 1 之間，PF_3 參數的值越接近 1，分數低的階段被選中的機率越高，而選擇的機率隨著值的減小而降低，使用方程式 (34 和 35)計算階段函數改變的成本：

$$F_t^{exploit} = \left(\alpha_t^{exploit} \times \left(f1_t^{exploit}\right)\right) + \left(\alpha_t^{exploit} \times \left(f2_t^{exploit}\right)\right) + \left(\delta_t^{exploit} \times \left(lc \times f3_t^{exploit}\right)\right) \quad (34)$$

$$F_t^{explor} = \left(\alpha_t^{explor} \times \left(f1_t^{explor}\right)\right) + \left(\alpha_t^{explor} \times \left(f2_t^{explor}\right)\right) + \left(\delta_t^{explor} \times \left(lc \times f3_t^{explor}\right)\right) \quad (35)$$

$$lc = \left\{\{|Cost_{old} - Cost_{new}|\}^{exploit}, \{|Cost_{old} - Cost_{new}|\}^{explor}\right\}, 0 \notin lc \quad (36)$$

$$\delta_t^{exploit} = 1 - \alpha_t^{exploit} \quad (37)$$

$$\delta_t^{explor} = 1 - \alpha_t^{explor} \quad (38)$$

使用方程式(34 和 35)，計算每個探索和開發階段的最終成本，參數α 和δ 都是探索和開發階段的變數，優先考慮多樣性 f_3，否則，f_3 的優先權會減小，如果參數α的值接近 1，則優先處理 f_2。如果探索階段函數的成本大於開發階段函數的成本，則開發階段函數的參數α的值將降至值為 0.01，而另一方面，開發階段的參數α的最大值將趨近於 1，但是如果開發階段函數的成本高於探索階段函數的成本，則上述過程將相反。lc 是從開發和探索階段改進過程中獲得的一組計算成本差異，其中包括一組非零的值 ($0 \notin lc$)。

(三) 探索階段：

在探索階段，受到美洲獅尋找食物行為的啟發，可能去舊的領域之中尋找，也可能到全新的領域。在這個階段，美洲獅在他們的領土上隨機搜索以尋找食物，或者隨機靠近其他美洲獅並使用它們的獵物，因此，美洲獅會隨機跳入搜索空間或在美洲獅之間的空間中尋找食物，首先將整個種群按升冪排序，然後在探索階段使用方程式(39)改進其解決方案：

$$If\ \ rand_1 \ > \ 0.5, Z_i, G \ = \ R_{dim} \times (ub \ - \ lb) + lb$$
$$Otherwise, Z_{i,G} = X_{a,G} + G \times (X_{a,G} - X_{b,G}) + G \times \Big(\big((X_{a,G} - X_{b,G}) - (X_{c,G} - X_{d,G})\big) + \big((X_{c,G} - X_{d,G}) - (X_{e,G} - X_{f,G})\big)\Big) \tag{39}$$
$$G = 2 \times rand_2 - 1 \tag{40}$$

在方程式(39)中，*ub* 和 *lb* 是整個矩陣的上下限，R_{dim} 是隨機生成範圍在 0 和 1 之間的數，$rand_1$ 和 $rand_2$ 兩者都是隨機生成介於 0 和 1 之間的數。$X_{a,G}$、$X_{b,G}$、$X_{c,G}$、$X_{d,G}$、$X_{e,G}$、$X_{f,G}$是整個總體中隨機選擇的其他個體，*G* 使用方程式(40)計算。根據式(39)，根據現有條件選擇兩個方程式中的一個方程式來產生不同的解，然後應用新的解來改進當前解。

$$X_{new} = \begin{cases} Z_{i,G}, if j = j_{rand} or rand_3 \le U \\ X_{a,G}, otherwise \end{cases} \tag{41}$$
$$NC = 1 - U \tag{42}$$
$$p = \frac{NC}{Npop} \tag{43}$$
$$if Cost X_{new} < Cost X_i, U = U + p \tag{44}$$

在方程式(41)中，$Z_{i,G}$來自方程式(39)，j_{rand} 是矩陣維度範圍內隨機的整數，$rand_3$ 也是一個介於 0 和 1 之間的隨機數，*NC* 來自方程式(42)，*U* 是在整個過程之前設置的參數，其值介於 0 到 1 之間。在方程式(43)中，*Npop* 是美洲獅的總數，根據方程式(44)中的條件改進，如果滿足此條件，則僅更新解的維度，這個動作可以避免陷入局部最優，且具有好的多樣性，另外，考慮到在探索階段每次反覆運算開始時，搜索的代理個體根據其成本按升冪排序，高品質的解決方案優先，然後根據方程式(42–44)，一開始，由於 *U* 參數的值很小，因此解決方案不會發生太多變化，後來隨著該參數的增加，具有較高成本的解會發生變化，這種方法導致探索出更差解，目的是在空間中找最佳解，另外，如果找到的美洲獅不比現在的美洲獅好，那麼就不會做方程式(44)，因為如果已經得到改進，就沒有必要增加多餘的發現，但是高品質的解決方案幾乎沒有發生變化和只是避免陷入局部最優。最後，使用方程式(45)將新生成的解替換為當前解：

$$X_{a,G} = X_{new}, if X_{i,new} < X_{a,G} \tag{45}$$

根據方程式(45)，如果新的解比當前解具有更好的成本，則該解決方案將取代當前解決方案。

(四) 開發階段：

在開發階段，PO 演算法利用兩個不同的算子來改進解決方案，這兩種機制是基於美洲獅的伏擊和衝刺進行狩獵，在自然界中，美洲獅會試圖在各種環境中伏擊獵物，在某些情況下它會追趕獵物，這些行為是使用方程式(46)來判斷的：

$$\mathrm{X_{new}} = \begin{cases} \text{if } rand_4 \ge 0.5, \mathrm{X_{new}} = \dfrac{\left(\frac{\mathrm{mean(Sol_{total})}}{\mathrm{Npop}}\right) \times \mathrm{X_1^r} - (-1)^{\beta} \times \mathrm{X_i}}{1 + (\alpha \times rand_5)} \\ \text{otherwise, if } rand_6 \ge \mathrm{L}, \mathrm{X_{new}} = \mathrm{Puma_{male}} + 2rand_7 \times \exp(randn_1) \times (\mathrm{X_2^r} - \mathrm{X_i}) \\ \text{otherwise}, \mathrm{X_{new}} = 2rand_8 \times \dfrac{\mathrm{F_1 \times R \times X(i) + F_2 \times (1-R) \times Puma_{male}}}{2rand_9 - 1 + randn_2} - \mathrm{Puma_{male}} \end{cases} \tag{46}$$

方程式(46)顯示了 PO 開發階段中使用的三種策略，如果 $rand_5$ (介於 0 和 1 之間的隨機數)大於 0.5，執行開發階段中方程式(46)中的案例 1 快速奔跑策略，否則選擇伏擊策略，該策略由兩種不同的操作組成，短跳以及長跳，方程式(46)中的案例 2 用於模擬美洲獅進行短跳的狩獵，方程式(46)中的案例 3 用於模擬美洲獅進行長跳的狩獵。*mean* 表示函數的均值，Sol_{total} 表示所有解的總和，*Npop* 是執行的總體總數。X_1^r是整個總體中隨機選擇的解，β 是隨機產生的 0 或 1。X_i是當前反覆運算中的當前解，α 和 L 是在整個過程之前必須調整的靜態參數，$Puma_{male}$ 是整個總體的最佳解，$rand_4$、$rand_5$、$rand_6$、$rand_7$、$rand_8$ 和 $rand_9$ 是介於 0 到 1 之間的隨機數，*exp* 表示指數函數，$randn_1$ 和 $randn_2$ 是常態分佈和問題維度的隨機數，X_2^r是於方程式 (47)隨機選擇的解：

$$X_2^r = round\ (1 + (N_{pop} - 1) \times rand_{10}) \tag{47}$$

在方程式(47)中，將 X 的每個元素四捨五入到最接近的整數，$rand_{10}$ 是介於 0 到 1 之間的隨機數，*Npop* 是美洲獅的總數，最後，R、$F1$ 和 $F2$ 由方程式 (48-50)計算：

$$R = 2 \times rand_{11} - 1 \tag{48}$$

$$F_1 = randn_3 \times \exp\left(2 - Iter \times \left(\frac{2}{maxIter}\right)\right) \tag{49}$$

在方程式(49)中，$randn_2$ 是常態分佈和問題維度的隨機數，*Iter* 表示當前的反覆運算次數，*maxIter* 表示最大的反覆運算次數，*exp* 表示指數函數。

$$F_2 = randn_4 \times (randn_5)^2 \times \cos\ (\ (2 \times rand_{12}) \times randn_4) \tag{50}$$

在方程式(50)中，$randn_4$ 和 $randn_5$ 都是常態分佈和問題維度的隨機數，*Cos* 代表餘弦函數，$rand_{12}$ 是介於 0 到 1 之間的隨機數。最後，在此階段結束時，如果新生產的解決方案的成本低於當前解決方案，則將其替換，取低成本解決方案。圖 3 為 PO 演算法運作流程之示意圖。

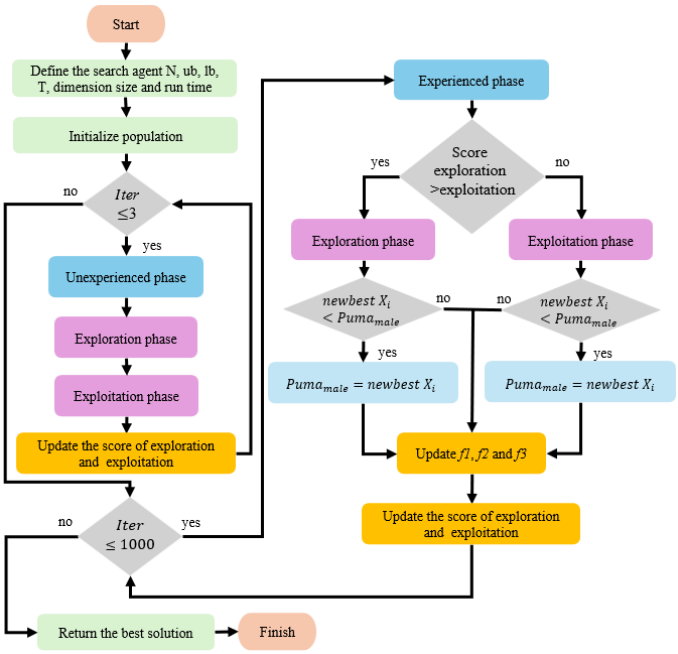

圖 3. PO 運作流程圖。

三、PEM 燃料電池參數估計

使用 PO 演算法調整 PEMFC 的最佳參數值，其中含有七個參數值 ($\xi 1$、$\xi 2$、$\xi 3$、$\xi 4$、λ、R_c 和 b)。在決定待定參數的解的優劣，會以誤差平方和 (Sum Square Error, SSE) 評估的演算法之適應度 (Fitness)，方程式(51)的 V_{exp} 和 V_{est} 表示實驗電壓和估計電壓，k 表示指數，而 N 是可用數據集的數量 (Singla et al.,2023)。

$$SSE = \sum_{k=1}^{N} \left(V_{exp} - V_{est}\right)^2 \tag{51}$$

根據滿足條件初始化初始隨機解，並輸入 PO 演算法的模組，由 PO 演算法輸出最佳解決方案 (Best Solution)和相應的最佳的 SSE。

四、演算法計算階段的多樣性平衡

元啟發式演算法需要在探索和開發之間取得適當的平衡才能取得較好的搜尋策略，其中一種就是通過觀察當前的種群多樣性來量化演算法中的探索和開發階段，參考 Mirjalili 研究中提出的維度多樣性測量 (Mirjalili et al.,2014)。

對於每種元啟發式演算法，探索和開發階段各代表了演算法在計算過程中的工作方向，探索階段是指演算法搜索分佈在設定空間中不同區域的能力，屬於全域搜索。另一方面，開發階段則是加強搜索過程的解，以找到更好的解或改進現有的解，屬於區域探索。探索與開發能力與其收斂率之間存在很緊密的關係，雖然開發階段可以提高向最佳解的收斂速度，但同時也會增加陷入局部最優的可能性。相反，更傾向於探索而不是開發的演算法會增加在空間中找到全域最佳解更有可能所在的區域的可能性，但代價是演算法的收斂速度惡化。

元啟發式演算法使用一組候選解決方案以及代理人來探索空間，目的是為了找到最佳的解決方案，在過程中，擁有最佳解決方案的搜索代理會傾向於將搜索過程引導到他們身上，在這種情況下，搜索代理之間的距離減小，而開發效果增加。另一方面，當搜索代理之間的距離增加時，探索階段的效果則更加明顯。為了計算探索/開發百分比，本研究將探索階段與探索階段在最大迭代次數內的執行次數計算出來，並觀測 PO 演算法的兩種階段運行分布結果。

肆、結果與討論

本實驗首先使用 NedStack PS6 的數據集和 BCS500W 的數據集來評估 PO 演算法用於 PEMFC 的效率，**表 1** 給出了每個參數的邊界(即下限 lb 和上限 ub) (Kandidayeni et al.,2019)，**表 2** 為 NedStack PS6 電堆的規格 (Cao et al.,2019)，要輸入到燃料電池的數學模型中做計算，為了證明 PO 的能力，將燃料電池的數學模型套入其演算法計算所獲得的七個最佳的參數 ($\xi 1$、$\xi 2$、$\xi 3$、$\xi 4$、λ、R_c 和 b)、適應度 (*Fitness*)、以及標準差 (*Std.*)，並與其他演算法進行了比較，包括 GWO (Mirjalili et al.,2014)、MRFO (Selem et al.,2020)、HHO、LHHO、HBA (Haddad et al.,2024)每種演算法的參數值都是根據每種演算法初始化分配的，但個體總數和反覆運算次數等標準參數分別都為 30 和 1000，**表 3** 為 PO 與其他每種演算法計算 NedStack PS6 所獲得的最佳參數、計算之適應度以及標準差，從這些結果可以看出，在測試數據集中，PO 的 *Fitness* 數值以及 *Std* 值優於其他演算法，也就是說使用 PO 演算法可以獲得最佳的解，用來估算七個參數值較為準確。**表 4** 為 NedStack PS6 理論實驗電流 I_{cell}、電壓 V_{exp} 與用 PO 演算法所獲得的估計電壓 V_{est} (Abd Elaziz et al.,2023)，可以看到使用 PO 演算法計算出來的估計電壓

V_{est} 非常接近理論實驗電壓 V_{exp}，**圖 4 (a)**可以看出 PO 在搜尋解答的速度非常快，**圖 4 (b)**繪出 PO 演算法比對實際燃料電池數據之 I-V 數值。

表 1. PEMFC 參數的範圍 (Kandidayeni et al.,2019)。

Parameter	lb	ub
ξ1	-1.1997	-0.8532
ξ2 (10^{-3})	0.8	6
ξ3 (10^{-5})	3.6	9.8
ξ4 (10^{-4})	-2.6	-0.954
λ	10	23
R_c (10^{-4})	1	8
b	0.0136	0.5

表 2. NedStack PS6 電堆規格 (Cao et al.,2019)。

PEMFC stacks	NedStack PS6
n	65
l (μm)	178
A (cm^2)	240
I_{max} (A/cm^2)	1.4
T (K)	343
P_{H2} (atm)	1
P_{O2} (atm)	1

表 3. NedStack PS6 燃料電池以 PO 其他演算法的參數比較 (Riad et al.,2023; Haddad et al.,2024)

	PO	GWO	MRFO	HHO	LHHO	HBA
ξ1	-0.8532	-0.892956	-0.9381	-0.8532	-0.8532	-0.8647
ξ2 (10^{-3})	2.98616	3.28997	3.4861	2.9249	3.0449	2.5666
ξ3 (10^{-5})	7.8	9.14197	9.512	7.366	8.229	4.5764
ξ4 (10^{-4})	-0.954	-0.954	-0.9544	-0.954	-0.954	-0.954
λ	13.0947	13.6364	13.096	18.532	13.4056	10
R_c (10^{-4})	1	1.01276	1	3.4467	3.2791	3.111
b	0.0136	0.0234776	0.01451	0.15553	0.15847	0.16109
Fitness	**2.0792**	2.1052	2.136	2.304	2.29328	2.288128
Std.	**0.02265**	0.18417	0.0326	2.502033	1.655048	0.95129

表 4. NedStack PS6 實驗電流、電壓與 PO 計算之電壓。

NedStack PS6		
I_{cell} (A)	V_{exp} (V)	V_{est} (V)
2.25	61.64	62.3539
6.75	59.57	59.7798
9	58.94	59.0483
15.75	57.54	57.4961
20.25	56.8	56.7175
24.75	56.13	56.0442
31.5	55.23	55.157
36	54.66	54.6204
45	53.61	53.6329

NedStack PS6		
I_{cell} (A)	V_{exp} (V)	V_{est} (V)
51.75	52.86	52.9439
67.5	51.91	51.4396
72	51.22	51.0271
90	49.66	49.4188
99	49	48.6281
105.8	48.15	48.0323
110.3	47.52	47.638
117	47.1	47.0497
126	46.48	46.2551
135	45.66	45.453
141.8	44.85	44.8404
150.8	44.24	44.019
162	42.45	42.9766
171	41.66	42.1201
182.3	40.68	41.0171
189	40.09	40.347
195.8	39.51	39.6534
204.8	38.73	38.7128
211.5	38.15	37.9945

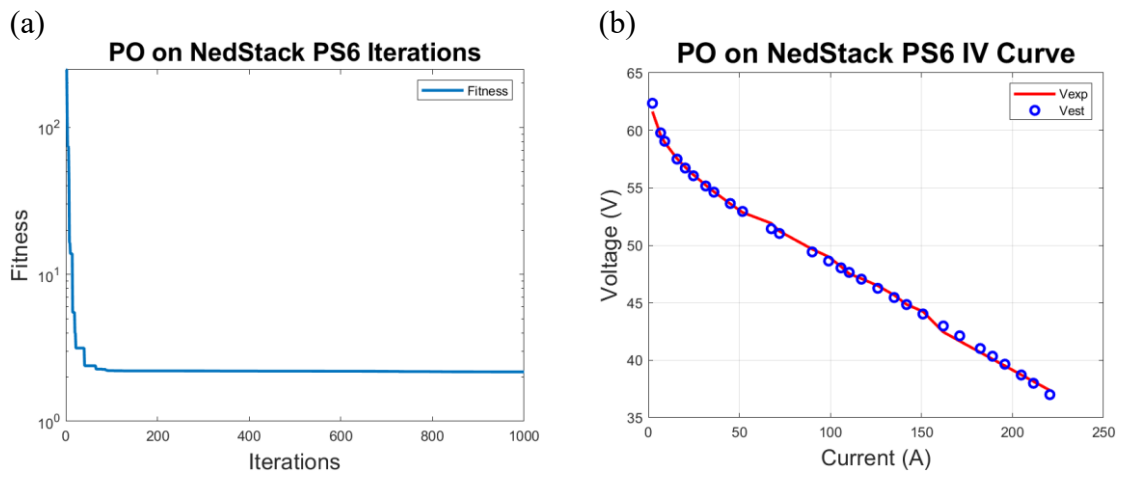

圖 4. NedStack PS6 系統的 **(a)**參數辨識迭代圖與 **(b)**估測電壓與實驗電壓對比圖。

當 PO 應用於 NedStack PS6 電池模組時，分別計算探索與開發的比例，最大迭代次數為 1000，無經驗階段分別各運行 3 次，有經驗階段為第 4 次到第 1000，**表 5** 中在有經驗階段，探索總共執行了 558 次，為 55.9%，開發執行了 439 次，為 44 %，由**圖 5** 可見其過程中的比例變化，Flag1 為探索階段，Flag2 為開發階段。

表 5. NedStack PS6 探索以及開發階段占比。

Phase	Exploration	Exploitation
無經驗	3	3
有經驗	558	439
百分比	55.9%	44%

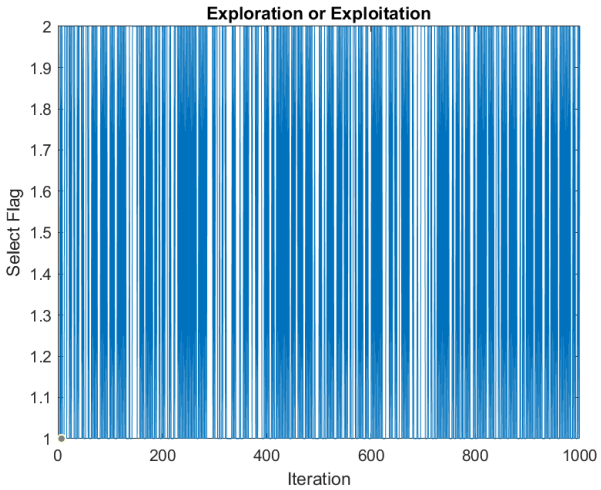

圖 5. PO-NedStack PS6 運算中切換探索或開發階段的狀態分布。

使用 BCS 500 W 的數據集與實驗結果比對來評估 PO 演算法用於 PEMFC 的效果，**表 6** 為 BCS 500 W 電堆規格 (Sultan et al.,2021)，要輸入到燃料電池的數學模型中做計算，**表 7** 為 PO 與其他演算法 GWO、WOA、COA、HHO、LHHO 計算 BCS 500 W 之所獲得的最佳參數及計算之適應度和標準差，從**表 7** 可以看出，在 BCS 500 W 的測試數據集中，PO 的 *Fitness* 以及 *Std* 值皆優於其他演算法， PO 演算法能為這個系統算出最佳的解，用來估算七個參數值較為準確。**表 8** 為 BCS 500 W 理論實驗電流 I_{cell}、電壓 V_{exp} 與用 PO 演算法所獲得的估計電壓 V_{est} (Abd Elaziz et al.,2023)，可以看到使用 PO 演算法計算出來的估計電壓 V_{est} 非常接近理論實驗電壓 V_{exp}，這表示使用 PO 演算法計算的準確度很高，**圖 6 (a)**可以看出 PO 在 BCS 500 W 系統搜尋最佳解的狀況，**圖 6 (b)**繪出 PO 演算法比對實際燃料電池數據之 I-V 數值。

表 6. BCS 500 W 電堆規格 (Sultan et al.,2021)。

PEMFC stacks	BCS 500 W
n	32
l (μm)	178
A (cm^2)	64
I_{max} (A/cm^2)	0.469
T (K)	333
P_{H2} (atm)	1
P_{O2} (atm)	0.2095

表 7. 基於 BCS 500 W 與其他演算法的參數比較 (Riad et al.,2023; Haddad et al.,2024) 。

	PO	GWO	WOA	COA	HHO	LHHO
ξ1	-1.15768	-1.10556	-0.927045	-0.862256	-1.1747	-1.0812
ξ2 (10^{-3})	3.85806	3.08595	2.48206	2.96136	3.7964	3.726
ξ3 (10^{-5})	9.72608	5.31237	4.75249	9.66927	7.7614	9.1478
ξ4 (10^{-4})	-1.92863	-1.92121	-1. 93062	-1.88069	-1.9242	-1.9203
λ	22.1933	21.191	21.5094	23	12.8019	12.263
R_c (10^{-4})	2.17198	2.1986	1.56128	6.297	2.2064	1.911
b	0.0162455	0.0154543	0.0193433	0.0144618	0.013729	0.016813
Fitness	**0.011556**	0.011809	0.032762	0.011768	0.046078	0.0129385
Std.	**0.00011003**	0.0025672	0.97891	0.0040944	2.950069	0.0674357

表 8. BCS 500 W 實驗電流、電壓與用 PO 獲得的估計電壓 (Abd Elaziz et al.,2023)。

BCS 500 W		
I_{cell} (A)	V_{exp} (V)	V_{est} (V)
0.6	29	28.9964
2.1	26.31	26.3065
3.58	25.09	25.0946
5.08	24.25	24.2559
7.17	23.37	23.3768
9.55	22.57	22.5859
11.39	22.06	22.0618
12.54	21.75	21.7597
13.73	21.45	21.4624
15.73	21.09	20.9887
17.02	20.68	20.6953
19.11	20.22	20.2316
21.2	19.76	19.7713
23	19.36	19.3662
25.08	18.86	18.8665
27.17	18.27	18.2746
28.06	17.95	17.9531
29.26	17.3	17.2929

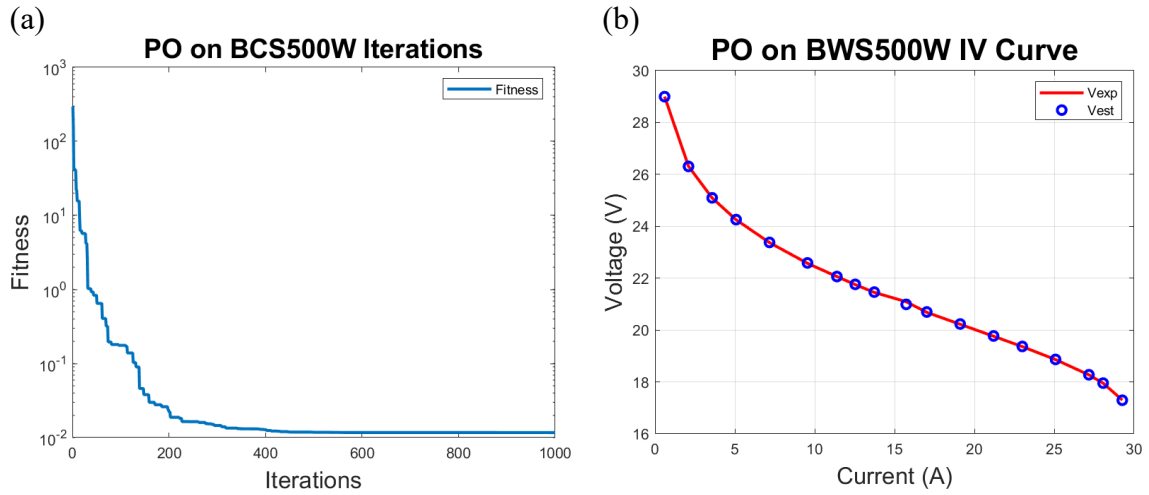

圖 6. BCS500W 系統的 **(a)** PO 參數辨識迭代圖與 **(b)** 估測電壓與實驗電壓對比圖。

當 PO 應用於 BCS500W 電池模組時，分別計算探索與開發的比例，最大迭代次數為 1000，無經驗階段分別各運行 3 次，有經驗階段為第 4 次到第 1000，**表 9** 中在有經驗階段，探索總共執行了 577 次，為 57.8%，開發執行了 420 次，為 44%，由**圖 7** 可見其過程中的比例變化，Flag1 為探索階段，Flag2 為開發階段。

表 9. BCS500W 探索以及開發階段占比。

Phase	Exploration	Exploitation
無經驗	3	3
有經驗	577	420
百分比	57.8%	42.1%

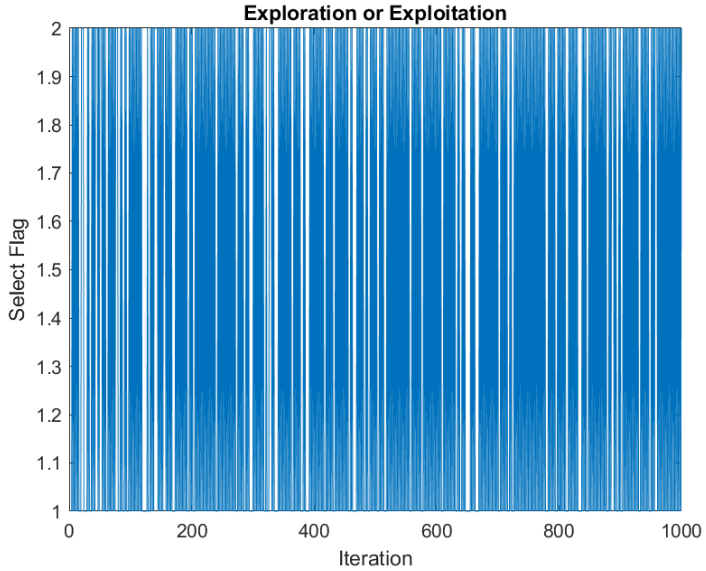

圖 7. PO-BCS500W 運算中切換探索或開發階段的狀態分布。

為了進一步研究 PO 作用於 PEMFC 模型的性能，計算了兩種統計指標，如**表 3** 所示的 *Fitness* 和標準差 *Std*，從這些數據中，可以注意到 PO 的適應度 (*Fitness*)值和標準差 *Std* 值比其他演算法還要好，證明 PO 運用在這兩種燃料電池的系統上比其他演算法更具有找到最佳參數的能力，PO 在運算中進行探索階段的次數皆大於開發階段，證明 PO 開發階段相對較弱，對於探索與開發的平衡還有探討的空間。

伍、結論

演算法應用於燃料電池的性能對於系統設計、模擬和預測效能極為重要，由於 PEMFC 的特性非常複雜多變，因此尋找到適合的優化方法對於建模技術而言至關重要。本研究發展以 Puma Optimizer (PO)鑑定 PEMFC 的模組參數方法，進一步推估出燃料電池系統的最大功率點。透過演算法快速求解 PEMFC 的未知參數，並與其他演算法進行比較，以驗證 PO 方法的適配性，燃料電池類型使用了 NedStack PS6 和 BCS 500 W 兩種不同案例。計算結果顯示，PO 在 *SSE* 以及標準差方面均優於其他演算法，證實 PO 方法在處理燃料電池的非線性特徵時具有高度適配的搜索能力。

PO 擁有與燃料電池高度適配的能力，在兩種不同電池系統的運算中，探索階段皆大於開發階段，推測是因為演算法的開發能力較弱所導致，演算法在開發階段中進行運算的方式若較不完全，若要提升 PO 的開發階段，可以直接進行開發階段算法的修改，或是與不同演算法混合應用，以提升演算法整體搜尋性能，為未來可關注的研究方向。

引用文獻

方俊德(2023)。台灣淨零碳排路徑初探。*臺灣經濟研究月刊*，*46* (1)，13-20。

Abdollahzadeh, B., Khodadadi, N., Barshandeh, S., Trojovský, P., Gharehchopogh, F. S., El-kenawy, E. S. M., ... & Mirjalili, S. (2024). Puma optimizer (PO): A novel metaheuristic optimization algorithm and its application in machine learning. *Cluster Computing*, 1-49.

Abd Elaziz, M., Abualigah, L., Issa, M., & Abd El-Latif, A. A. (2023). Optimal parameters extracting of fuel cell based on Gorilla Troops Optimizer. *Fuel*, *332*, 126162.

Agushaka, J. O., Ezugwu, A. E., & Abualigah, L. (2023). Gazelle optimization algorithm: a novel nature-inspired metaheuristic optimizer. *Neural Computing and Applications*, *35* (5), 4099-4131.

Ayyarao, T. S., Polumahanthi, N., & Khan, B. (2024). An accurate parameter estimation of PEM fuel cell using war strategy optimization. *Energy*, *290*, 130235.

Bizon, N., Mazare, A. G., Ionescu, L. M., & Enescu, F. M. (2018). Optimization of the proton exchange membrane fuel cell hybrid power system for residential buildings. *Energy Conversion and Management*, *163*, 22-37.

Budak, Y., & Devrim, Y. (2018). Investigation of micro-combined heat and power application of PEM fuel cell systems. *Energy Conversion and Management*, *160*, 486-494.

Cao, Y., Li, Y., Zhang, G., Jermsittiparsert, K., & Razmjooy, N. (2019). Experimental modeling of PEM fuel cells using a new improved seagull optimization algorithm. *Energy Reports*, *5*, 1616-1625.

Dannenberg, K., Ekdunge, P., & Lindbergh, G. (2000). Mathematical model of the PEMFC. *Journal of Applied Electrochemistry*, *30*, 1377-1387.

Duan, F., Chen, C., Song, F., Zhang, S., Li, X., & Eslami, M. (2023). An optimal parameters estimation for the proton exchange membrane fuel cells based on amended deer hunting optimization algorithm. *Sustainable Energy Technologies and Assessments*, *58*, 103364.

El-Fergany, A. A. (2018). Extracting optimal parameters of PEM fuel cells using Salp Swarm Optimizer. *Renewable Energy*, *119*, 641-648.

Fathy, A., Abd Elaziz, M., & Alharbi, A. G. (2020). A novel approach based on hybrid vortex search algorithm and differential evolution for identifying the optimal parameters of PEM fuel cell. *Renewable Energy*, *146*, 1833-1845.

Gamburzev, S., & Appleby, A. J. (2002). Recent progress in performance improvement of the proton exchange membrane fuel cell (PEMFC). *Journal of power sources*, *107* (1), 5-12.

Gouda, E. A., Kotb, M. F., & El-Fergany, A. A. (2021). Jellyfish search algorithm for extracting unknown parameters of PEM fuel cell models: Steady-state performance and analysis. *Energy*, *221*, 119836.

Gong, X., Dong, F., Mohamed, M. A., Abdalla, O. M., & Ali, Z. M. (2020). A secured energy management architecture for smart hybrid microgrids considering PEM-fuel cell and electric vehicles. *Ieee Access*, *8*, 47807-47823.

Hasanien, H. M., Shaheen, M. A., Turky, R. A., Qais, M. H., Alghuwainem, S., Kamel, S., ... & Jurado, F. (2022). Precise modeling of PEM fuel cell using a novel Enhanced Transient Search Optimization algorithm. *Energy*, *247*, 123530. Kandidayeni, M.,

Macias, A., Khalatbarisoltani, A., Boulon, L., & Kelouwani, S. (2019). Benchmark of proton exchange membrane fuel cell parameters extraction with metaheuristic optimization algorithms. *Energy*, *183*, 912-925.

Haddad, S., Benghanem, M., Hassan, B., Soukkou, A., Lekouaghet, B., & Soukkou, Y. (2024). Parameters optimization of PEMFC model based on gazelle optimization algorithm. *International Journal of Hydrogen Energy*, *87*, 214-226.

Kriegel, H. P., Schubert, E., & Zimek, A. (2017). The (black) art of runtime evaluation: Are we comparing algorithms or implementations?. *Knowledge and Information Systems*, *52*, 341-378.

Mann, R. F., Amphlett, J. C., Hooper, M. A., Jensen, H. M., Peppley, B. A., & Roberge, P. R. (2000). Development and application of a generalised steady-state electrochemical model for a PEM fuel cell. *Journal of power sources*, *86* (1-2), 173-180.

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in engineering software*, *69*, 46-61.

Qais, M. H., Hasanien, H. M., Turky, R. A., Alghuwainem, S., Loo, K. H., & Elgendy, M. (2022). Optimal PEM fuel cell model using a novel circle search algorithm. *Electronics*, *11* (12), 1808.

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

Riad, A. J., Hasanien, H. M., Turky, R. A., & Yakout, A. H. (2023). Identifying the PEM fuel cell parameters using artificial rabbits optimization algorithm. *Sustainability*, *15* (5), 4625.

Selem, S. I., Hasanien, H. M., & El-Fergany, A. A. (2020). Parameters extraction of PEMFC's model using manta rays foraging optimizer. *International Journal of Energy Research*, *44* (6), 4629-4640.

Singla, M. K., Hassan, M. H., Gupta, J., Jurado, F., Nijhawan, P., & Kamel, S. (2023). An enhanced efficient optimization algorithm (EINFO) for accurate extraction of proton exchange membrane fuel cell parameters. *Soft Computing*, *27* (14), 9619-9638.

Sultan, H. M., Menesy, A. S., Kamel, S., Turky, R. A., Hasanien, H. M., & Al-Durra, A. (2021). Optimal values of unknown parameters of polymer electrolyte membrane fuel cells using improved chaotic electromagnetic field optimization. *IEEE Transactions on Industry Applications*, *57* (6), 6669-6687.

Sun, S., Su, Y., Yin, C., & Jermsittiparsert, K. (2020). Optimal parameters estimation of PEMFCs model using converged moth search algorithm. *Energy Reports*, *6*, 1501-1509.

Turgut, O. E., & Coban, M. T. (2016). Optimal proton exchange membrane fuel cell modelling based on hybrid Teaching Learning Based Optimization–Differential Evolution algorithm. *Ain Shams Engineering Journal*, *7* (1), 347-360.

Yan, C., Chen, J., Liu, H., & Lu, H. (2019). Model-based fault tolerant control for the thermal management of PEMFC systems. *IEEE Transactions on Industrial Electronics*, *67* (4), 2875-2884.

Zhao, Y., Mao, Y., Zhang, W., Tang, Y., & Wang, P. (2020). Reviews on the effects of contaminations and research methodologies for PEMFC. *International journal of hydrogen energy*, *45* (43), 23174-23200.