

國立台灣大學工學院工業工程學研究所



碩士論文

Institute of Industrial Engineering

College of Engineering

National Taiwan University

Master Thesis

以類神經網路及萬用啟發式演算法離線最佳化模擬系

統參數-海底渦輪機應用

Simulation parameters optimization based on Neural network and
metaheuristics - An application of marine turbine

徐子堯

Tzu-Yao Hsu

指導教授：洪一薰 博士

Advisor: I-Hsuan Hong Ph.D.

中華民國 108 年 7 月

July 2019



摘要

為順應綠色能源興起的潮流，海流能發電也是其中的選項之一，但研究該發電模式需要大量的實驗數據來進行模擬測試，在電腦模擬效率不夠應付研究需求的情況下，本研究以對海底渦流機的電腦模擬系統，以類神經網路架設後設模型，希望能取代電腦模擬系統，並且進行離線最佳化的方式，以求得的最佳解之參數來近似現實世界的物理實驗。此一架構主要分為兩大部分，首先以近似電腦模擬系統為目標，以類神經網路來當作此研究的後設模型。接著利用粒子團與基因演算法，以近似物理實驗結果為目標找出最佳解。本研究最終訓練出的類神經網路模型和電腦模擬系統間的誤差，以平均絕對百分比誤差的衡量，取得了百分之五以下的誤差，且以粒子團演算法所最佳化出來的結果，與物理實驗達到誤差低於十四個百分點以下。此一離線最佳化流程在不損失精準度的情況下，達到了降低運算時間以及提升效率的目的。

關鍵字:電腦模擬系統、類神經網路、後設模型、粒子團演算法、基因演算法、離線最佳化。

Abstract



Simulations are used to estimate array efficiencies and environmental impacts in the marine turbine design. However, simulations need a considerable computation effort. This paper utilizes the neural network to develop our surrogate model describing the response surface between the design parameters of the simulation and experimental quantities of interest. The metaheuristics has been applied to searching for an optimal parameter setting of the simulation so that the error between the simulation and field experiment is minimized. Our numerical study demonstrates a set of parameter setting, which returns 14% error between the simulation and field experiment.

Keyword: Simulation 、 Neural network 、 Surrogate model 、 Metaheuristics 、
Offline optimization.

目錄



摘要.....	i
Abstract	ii
目錄.....	iii
圖目錄.....	iv
表目錄.....	v
第一章 緒論.....	1
第二章 離線最佳化.....	5
2.1 交叉驗證.....	6
2.2 後設模型.....	7
2.2.1 模型選擇.....	7
2.2.2 類神經網路.....	8
2.3 最佳化演算法.....	12
2.3.1 定義問題.....	12
2.3.2 粒子團演算法.....	13
2.3.3 基因演算法.....	16
第三章 數值分析.....	18
3.1 模型參數設置與樣本分布.....	18
3.2 後設模型擬合誤差.....	22
3.3 預測誤差.....	23
3.4 最佳解擬合誤差.....	24
3.5 最終誤差.....	25
第四章 結論.....	29
參考文獻.....	30



圖目錄

圖 2.1:離線最佳化流程圖(改編自 Sterling et al., 2019)	5
圖 2.2:K-Fold 示意圖(改編自洪唯凱, 民 107)	6
圖 2.3:後設模型挑選(Simpson, Poplinski, Koch & Allen 2001)	7
圖 2.4:人工類神經網路架構圖	8
圖 2.5:類神經網路流程圖	11
圖 2.6:粒子團演算法示意圖	13
圖 2.7:粒子團演算法示意圖	15
圖 2.8:基因演算法示意圖	17
圖 3.1:基因演算法模擬系統樣本箱型圖	20
圖 3.2:粒子團演算法模擬系統樣本箱型圖	20
圖 3.3: 粒子團演算法模擬系統樣本去除 4、5、6 距離點箱型圖	21
圖 3.4:粒子團演算法模擬系統與物理實驗誤差	27
圖 3.5:基因演算法模擬系統與物理實驗誤差	27
圖 3.6:粒子團演算法模擬系統與物理實驗在去除過遠距離下誤差	28

表目錄




表 2.1:常用的損失函數.....	9
表 2.2:常用的激活函數.....	10
表 3.1:模型比較表.....	19
表 3.2:類神經網路架構.....	19
表 3.3:運算時間.....	21
表 3.4:後設模型擬合誤差在無去除距離點下.....	22
表 3.5:後設模型擬合誤差在去除距離點(4、5、6)下.....	23
表 3.6:各模型預期誤差.....	24
表 3.7:各模型最佳解擬合誤差.....	25
表 3.8:各模型最終誤差.....	26



第一章 緒論

隨著環保意識的抬頭，再生能源逐漸地成為人們思考的選項。在再生能源興起的這個時代，台灣的再生能源佔總發電的比例，根據經濟部統計處的報告，在 107 年僅佔總發電量不到百分之五(<https://www.moega.gov.tw/>)，其原因不外乎建置成本過高，且發電量不穩定。而在再生能源的眾多選項當中，海洋作為發電能源便是一個相對穩定的選擇。世界第一座潮汐發電場朗斯潮汐電站(Rance Tidal Power Station)便在 1966 年投入使用，而台灣的成功大學也在 2015 年，成功的在台灣屏東小琉球周邊海域(陳陽益等人，民 104)，進行了幾次的海流發電測試，並且取得良好的實驗結果，相較於其他的綠色能源，海流能提供了更加穩定且對環境衝擊力小的發電方式。但若要推廣此一發電方式，便要更加深入研究其物理原理以及相關資訊，遺憾的是每一次的實驗皆須將渦輪發電機置入深海中，或是藉由拖船進行拖拉測試，皆需耗費巨額的成本，因此不能經常性的將此實驗重複大量的製作，不但是一組渦流機組的造價不菲，且機組在還底必須具有高度抗侵蝕的能力。

面對實驗成本過高等問題時，通常會使用超級電腦來進行模擬系統的建立，像是 Cui et al. (2010)利用電腦模擬系統建立了更有效率的地震預警系統，並且擴大了可偵測的範圍。Djurfeldt et al. (2008)則是利用 IBM 所生產的超級電腦 Blue Gene/L™結合類神經網路模型，對人體大腦進行電腦模擬。Barry (1990)則是早期利用電腦模擬系統，來進行運輸業路徑規劃系統的研究。上述的研究皆藉由電腦模擬系統來解決實驗成本過高，或是提高了研究結果的效率與品質。至於在潮汐發電上，已經有相關論文在探討如何使用超級電腦來建立模擬系統模擬潮汐發電機(James, Johnson, Barco, & Roberts, 2017)，利用此一方法所建的模擬系統，也能延伸至許多相關的發電系統(Larose et al., 2009)。



但有時模擬系統計算一次的時間很長，而研究又需要大量的實驗數據時，模擬系統生成的實驗數據就不足以應付研究需求，於是開始有後設模型(Surrogate-model)的概念產生，也有相關的研究是利用此一概念來進行，Eisenhower, O'Neill, Narayanan, Fonoberov & Mezić (2012)利用了後設模型，大幅減少模擬時所需的參數，並且得到了更好的預估表現。而利用後設模型進行離線最佳化(offline optimization)時，便可以解決模擬系統不存在明確的模型或如黑盒子(Black box)般的複雜，以至於計算耗時且難以進行最佳化計算的問題。另外線上最佳化與離線最佳化的優缺點，也有學者對此問題進行相關的探討(Barton & Meckesheimer, 2006)，並且應用在許多領域上面，Palopoli, Passerone & Rizano (2011) 這篇研究，利用離線最佳化來對有無線感測網路模型進行最佳化，最後找出了更有效率和可擴展性的實驗程序。也有利用離線最佳化來大幅降低油電混合公車的能源消耗和營運成本的相關研究 (Ribau, Viegas, Angelino, Moutinho & Silva, 2014)，離線最佳化運用在使用者混合推薦系統(Dooms, De Pessemier, & Martens, 2015)時，也顯著的提升了品質。上述提到的多項應用皆有一個共通點，便是難以進行實驗或模擬系統耗時等特性。

在後設模型中又有許多種的選擇，像是使用克利金模型(Kriging model)去描述黑盒子系統(Mehdad & Kleijnen, 2018)，在早期也有使用迴歸來建立後設模型者(Madu, 1990)。對於如何模擬超級電腦背後未知的模型，也可以使用深度學習(deep learning)的方式來訓練出最近似模擬系統背後的黑盒子，先前便有相關的研究如(Fonseca, Navarrese & Moynihan, 2003)，其中最主要的架構之一便是類神經網路(artificial neural network)，利用神經元(neural)以及倒傳遞(back propagation)的計算，來調節權重(weight)以建立模型。而類神經網路也有分許多的種類，諸如最原始的人工類神經網路(artificial neural network)，善於處理影像的卷積類神經網路(convolutional neural network)，還有擅長處理序列性資訊的遞歸神經網絡(recurrent neural network)，以及深層類神經網路(deep neural networks)。



類神經網路除了用來模擬黑盒子外，對於有著大量樣本資料，或是對研究課題背後的性質有太多未知或難以掌控的變因時，則很適合使用類神經網路模型，例如物件偵測(Szegedy, Toshev & Erhan, 2013)與 AI 醫生(Choi, Bahadori, Schuetz, Stewart & Sun, 2016)這兩篇論文，對於研究課題的樣本未知的變因過多，因此採用類神經網路模型進行模擬。在機器學習的領域裡，類神經網路已經有被廣泛運用甚至是成為主流的趨勢。至於如何對類神經網路進行最佳化，也有相關研究分別使用了粒子團演算法(Chau, 2007) (Mirjalili, Hashim & Sardroudi, 2012)與基因演算法(Vlahogianni, Karlaftis, & Golias, 2005)和布穀鳥演算法(Valian, Mohanna & Tavakoli, 2011)，而本研究使用的是基因演算法與粒子團演算法。

本研究結合上述的後設模型概念與萬用啟發式演算法，建立了一套離線最佳化的方式，有效的解決了原先的模擬系統利用線上最佳化的方式，造成要耗費大量時間卻只能生成少數的模擬實驗結果的問題。利用後設模型對模擬系統的歷史數據進行建模後，以萬用啟發式演算法對此一模型進行最佳化，便可在不失精準度的情況下節省大量的運算時間。

此一研究除了可套用在本次研究中所引用的海流能發電的案例，如上述可減少實驗成本等應用外，也可用於最佳化機台調參。本次研究是以物理實驗作為調參目標，在未來可利用相似的概念，調參目標改為利潤極大或成本最小為目的進行調參。利用此一離線最佳化的架構，便可在不使整個產線進行停止生產來進行實驗或最佳化的情況下，達到找出最佳化參數的目的。

本研究在第二章時將會以離線最佳化的流程，按數據處理、後設模型選擇與建立、萬用啟發式演算法的計算、以海流能渦流機組數據為例等步驟進行逐步介紹。在第三章時會以本次實際從模擬系統中取得的數據，按第二章的流程，調整模型與演算法參數後，分別按後設模型擬合誤差(Surrogate-model Fitting Error)、預測誤差(Predicted Error)、最佳解擬合誤差(Optimal Solution Fitting Error)與最終

誤差(Final Error)的順序，進行數值分析與比較。後設模型擬合誤差為計算後設模型在既有的歷史資訊下，對模擬系統的擬合程度，預測誤差為萬用啟發式演算法在最佳化的過程中，在後設模型裡找出的最佳解輸出與物理實驗的輸出間誤差，最佳解擬合誤差為衡量在非歷史資訊中，當模擬系統產生出一筆新的輸出時，後設模型是否預測相近，而最終誤差就是模擬系統利用後設模型找出的最佳解輸出後，與真實物理實驗相比的誤差。第四章時將會以各個誤差的表現進行整體研究的評斷，並對未來能夠持續改進研究的方向提出建議。



第二章 離線最佳化

為了進行離線最佳化，本次研究建立了以下流程，並且以驗證此一流程能否順利取得最佳化後的模擬系統參數為目標，其中主要是由電腦模擬系統、後設模型、萬能啟發式演算法這三大部分所組成，如圖 2.1 所示。此外此篇論文所使用的後設模型和最佳化的演算法，分別為類神經網路模型和基因演算法與粒子團演算法。最後將會以海底渦輪機的模擬系統參數為例，蒐集模擬系統的歷史數據，以此進行模擬系統的建模，並且在後設模型建立後從中找出最近似於物理實驗的最佳化參數解，以離線最佳化替代線上最佳化。

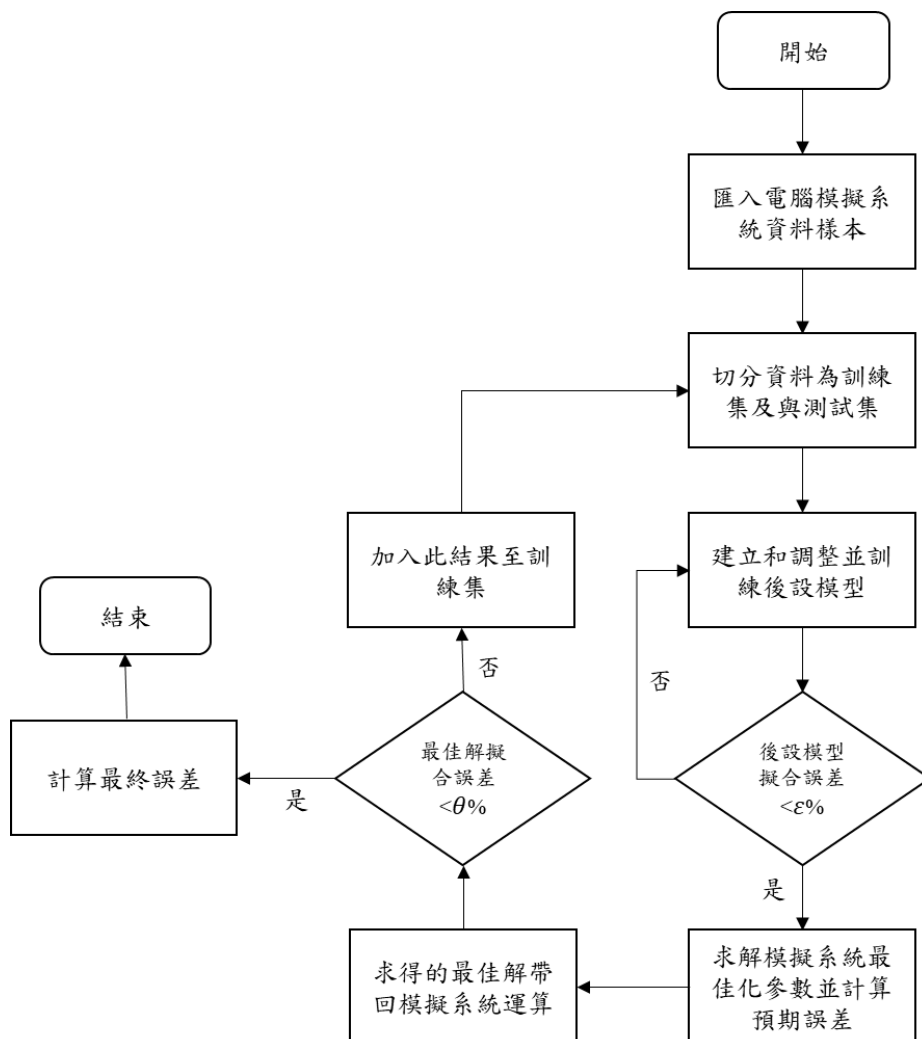


圖 2.1:離線最佳化流程圖(改編自 Sterling et al., 2019)



2.1 交叉驗證

在 Bengio & Grandvalet (2004) 與 Rodriguez, Perez & Lozano (2009) 這兩篇論文中，皆有提到當對樣本母體的分配未知的情況時，採用交叉驗證法(K-Fold Cross Validation)可以得到較為準確且合理的誤差。K-Fold 下樣本資料集的分法，是將樣本切分為訓練集(train set)、驗證集(validation set)、測試集(test set)。在進行 K 次的訓練如圖 2.2 所示，將樣本切為 K 等分，利用訓練集來建立模型，在利用驗證集來防止過度擬合(overfitting)，最後用測試集來選擇誤差最低者當作最終使用的模型。在本次的實驗當中，是將訓練集切分為五等分，其中四份為訓練集，另外一份為驗證集，測試集為原先獨立出來的另外一份資料，並不在訓練集當中，因此不會參與交叉驗證法的流程。

1.將訓練資料集切分為K等份						
1	2	3		K-2	K-1	K
2.選擇其中一份當作驗證集，其餘做為訓練集進行建模						
1	2	3		K-2	K-1	K
3.重複上述步驟K次						
1	2	3		K-2	K-1	K
4.計算平均誤差或挑選誤差最低者為最終模型						

圖 2.2:K-Fold 示意圖(改編自洪唯凱，民 107)

由於後設模型在訓練時需要有一個誤差函數來做為模型改善的依據，最後挑選誤差最低的模型來使用時，也需要一個衡量標準，在本次研究中所選用的誤差函數為平均絕對百分比誤差(Mean Absolute Percentage Error)，簡稱 MAPE。其算法為 $MAPE = \frac{1}{n} \sum_{i=1}^n \frac{|y_i^t - \hat{y}_i|}{y_i^t}$ ，式中的 y_i^t 為實際值而 \hat{y}_i 為預期值，此一誤差也是本次研究整個離線最佳化流程中和建立最佳化演算法時，所使用的誤差衡量方式。



2.2 後設模型

2.2.1 模型選擇

後設模型的存在是為了模擬一個高度複雜甚至是黑盒子的模型，在模擬系統進行模擬時，輸入的參數為 $X^s = [x_1^s, x_2^s, \dots, x_n^s]$ ，這當中的參數為模擬系統當中的參數，輸入參數後經過模擬系統這個黑盒子後，得到 $Y^s = [y_1^s, y_2^s, \dots, y_m^s]$ 。在建立後設模型時，目標為擬合模擬系統，輸入的參數是 $X^m = [x_1^m, x_2^m, \dots, x_j^m]$ ，其參數的選擇可以少於甚至是與模擬系統不同，但所輸出的結果 $Y^m = [y_1^m, y_2^m, \dots, y_m^m]$ ，則是希望盡量能近似模擬系統的輸出。後設模型的選擇上，Simpson, Poplinski, Koch & Allen (2001) 有建議的挑選方法，在對模擬系統背後的資料型態沒有基本的認知的情況下，可以使用類神經網路來進行對模擬系統的建模。

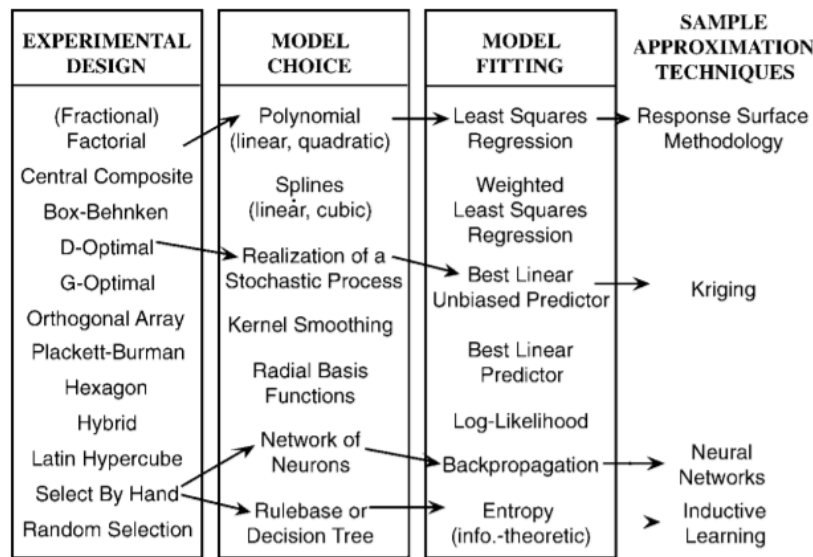


圖 2.3:後設模型挑選(Simpson, Poplinski, Koch & Allen 2001)

後設模型的概念類似於模型的模型，意指原先所建出的模型太過複雜，或是不易於進行線上最佳化，因此藉由擬合原先的模擬模型的歷史資料，來建出一個易於最佳化或增加運算效率的模型，以省時的離線最佳化的方式來代替耗時且耗費效能的線上最佳化。

2.2.2 類神經網路

在本研究當中使用了類神經網路模型當中的人工類神經網路(Artificial Neural Network)，人工類神經網路模型主要是由三個部分所組成，分別為輸入層(Input layer)、隱藏層(Hidden layer)、輸出層(Output layer)，每一層又包含了神經元(Neuron)與權數(Weights)和誤差項(Bias)，每一類神經元中由加權加總函數與非線性的激活函數(Activation function)所構成，以圖 2.4 為例，當中所選用的激活函數為 Relu。從上一層的神經元中獲取經由激活函數轉化後的值，作為加權加總函數的輸入值，再經由激活函數轉換後往下一層傳送，以此類推直到最後的輸出層，再以輸出層所輸出的預測結果與實際結果進行誤差計算。若誤差過大則可藉由驗證集誤差來判斷為低度擬合(underfitting)或過度擬合(overfitting)，當為低度擬合時可藉由增加樣本、加深隱藏層層數、增加神經元數量等方式將模型複雜化，但當為過度擬合的情況時，除了同樣的可藉由增加樣本外，也可藉由減少隱藏層層數、減少神經元數量等方式將模型簡化。人工類神經網路的架構圖如圖 2.4。

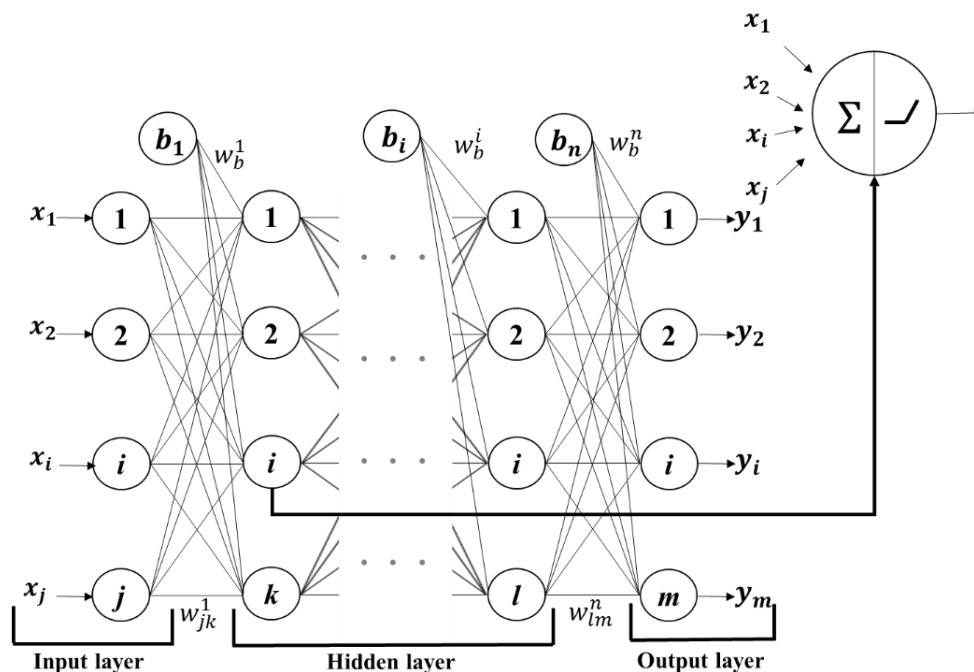


圖 2.4:人工類神經網路架構圖

在進行權數更新時，要利用倒傳遞(Backward Propagation)的方式，逐層推算回來並且進行每一層權數的更新，其基本的更新方式是藉由對損失函數(Loss function)中的權數進行偏為分和連鎖律的概念，如式(2.1):

$$\frac{\partial L}{\partial w} = \frac{\partial L}{\partial g} \frac{\partial g}{\partial h} \frac{\partial h}{\partial w} \quad (2.1)$$

上式中的 L 即為損失函數，常見的損失函數如表 2.1，在本次研究中所選用的損失函數為 MAPE。而上式中的 g 為激活函數，常見的激活函數如表 2.2，在本次研究中所選用的激活函數為 ReLu，最後 h 則是加權加總函數。由上式可知倒傳遞是將損失函數中的權數 w 進行偏微分，並藉由式(2.2)進行更新，式中的 α 為學習率。

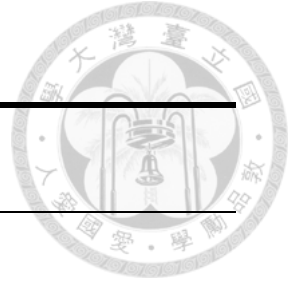
$$w_{new} = w_{old} - \alpha \frac{\partial L}{\partial w} \quad (2.2)$$

表 2.1:常用的損失函數

函數名稱	方程式
Mean Squared Error	$L(x) = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2$
Mean Absolute Percentage Error	$L(x) = \frac{1}{n} \sum_{i=1}^n \frac{ \hat{y}_i - y_i }{y_i}$
Binary Crossentropy	$L(x) = - \sum_{i=1}^n (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i))$
Categorical Crossentropy	$L(x) = - \sum_{i=1}^n \sum_{k=1}^k (y_{i,k} \log(\hat{y}_{i,k}))$

表 2.2:常用的激活函數

函數名稱	方程式
Sigmoid	$g(x) = \frac{1}{1 + e^{-x}}$
TanH	$g(x) = \tanh(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})}$
ArcTan	$g(x) = \tan^{-1}(x)$
ReLu	$g(x) = \begin{cases} 0 & \text{for } x \leq 0 \\ x & \text{for } x > 0 \end{cases}$
SoftPlus	$g(x) = \ln(1 + e^x)$



類神經網路層數與神經元數，常常是由研究者自己的經驗或是由試誤法來建設出。層數與神經元並非越多越好，隨著層數增加也許能降低訓練誤差，但也會造成過度擬合的情況，過少則會出現低度擬合，神經元的數量亦然。輸入層的神經元數量即為輸入的參數，輸出層的神經元數則為此模型的預測產出。本研究的後設模型為擬合模擬系統，因此所放入的參數也是模擬系統所需的四個參數，分別為 $C_{\epsilon 4}$ 、 $C_{\epsilon 5}$ 、 β_p 和 β_d ，此外還放入了距離點 d 和初始湍流強度，目標是要預測水流強度 I 與速度 U

人工類神經網路的計算流程如圖 2.5 所示，挑選完輸入的參數後，便給予更層權重亂數的起始值，並且按照前向傳播法(forwar propagation)進行計算各層的加權加總與激活函數值，逐層推算至輸出層。本研究將預期水流強度與速度與模擬系統的預期水流強度與速度計算誤差後，利用倒傳遞的方式進行更新權重，完全更新完後便完成一次的代次(epoch)，如此反覆計算直到代次達到設定上限次數便停止。

但由於本次選用的損失函數為 MAPE，因此在倒傳遞的過程中，可能會出現不易微分的問題。在本研究當中所採用的方式為分段式的微分，其方式如式(2.3)。



$$\frac{\partial \text{MAPE}}{\partial \hat{y}_i} = \begin{cases} -\frac{1}{N}, & \text{if } \hat{y}_i < y_i \\ \text{undefined}, & \text{if } \hat{y}_i = y_i \\ \frac{1}{N}, & \text{if } \hat{y}_i > y_i \end{cases} \quad (2.3)$$

由於 \hat{y}_i 幾乎不會等於 y_i ，因此在本次實驗當中的損失函數便可順利的進行倒傳遞。

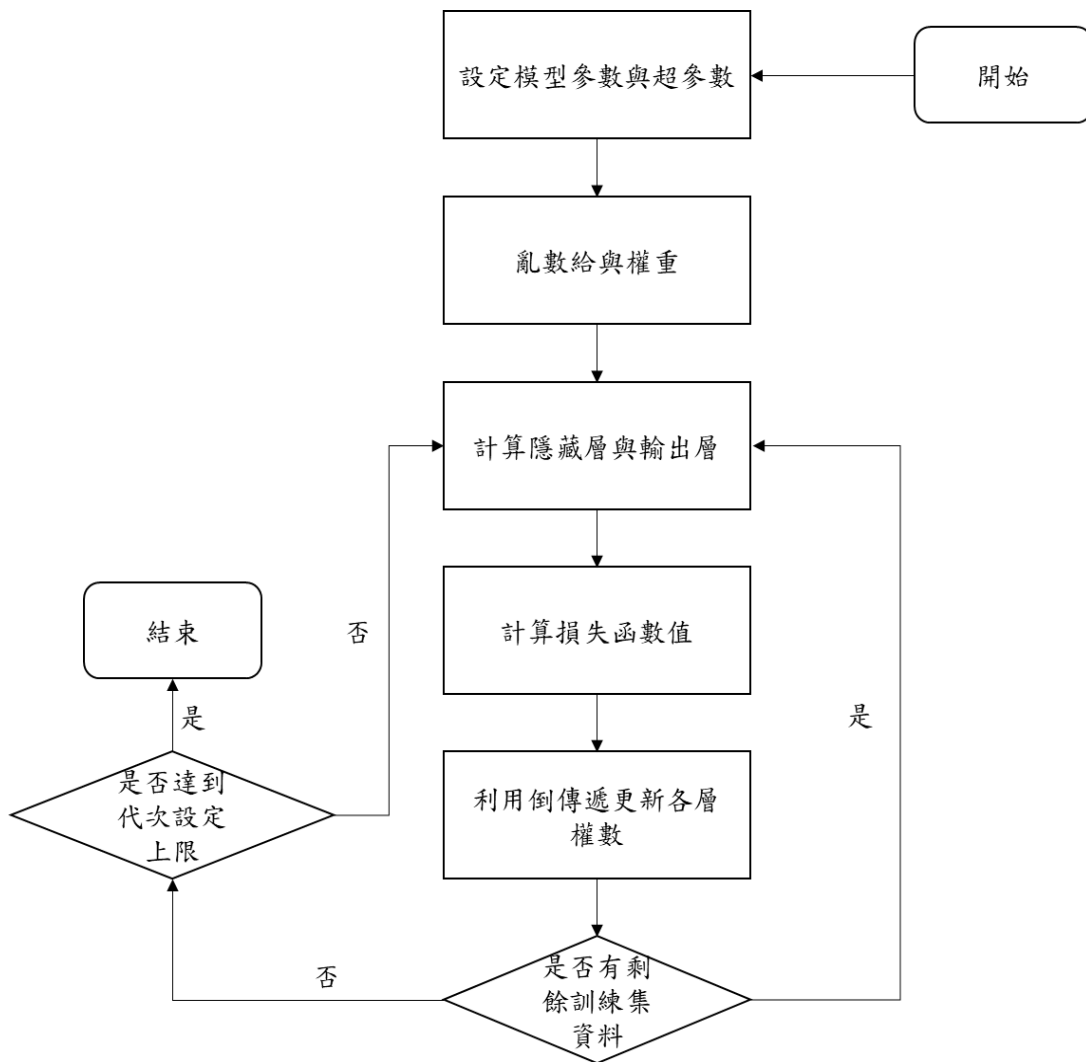


圖 2.5:類神經網路流程圖



2.3 最佳化演算法

2.3.1 定義問題

此研究的最終目的為從後設模型中找尋一組解，代回模擬系統後能夠近似物理實驗，因此必須藉由將後設模型做最佳化來進行求解。若直接將電腦模擬系統的模型進行線上最佳化求解，將會曠日廢時且耗費巨大的效能，因此轉而對模擬系統模型進行建模的後設模型進行離線最佳化求解。而最佳化求解中的目標函數 (objective function) 在本研究中設定為 MAPE，限制式 $\underline{x}_i < x_i < \bar{x}_i$ 是研究人員所設定的參數上下界，共有 m 個參數， $g(x)$ 則為後設模型，所產生的輸出為 \hat{y}_t ， y_t 為物理實驗數據，本次研究的整體架構如下式。

$$\min_{x_1, x_2, \dots, x_j} \frac{1}{n} \sum_{i=1}^n \frac{|y_i - \hat{y}_i|}{y_t} \quad (2.4)$$

$$\text{s.t. } \underline{x}_i < x_i < \bar{x}_i, i=1, 2, \dots, m \quad (2.5)$$

$$\text{where } \hat{y}_i = g(x)$$

從目標函數中可看出，本研究的最終目標即為找出一組從後設模型當中最佳化的參數解，代回模擬系統後能最近似於物理實驗結果的解，因此最小化後設模型輸出與物理實驗輸出兩者間的誤差即為本次研究離線最佳化的目標。就演算法方面，本篇研究所採用的分別為基因演算法與粒子團演算法，此二種演算法皆時常用於對類神經網路進行最佳化，其相關詳細內容將於第三章進行詳述。



2.3.2 粒子團演算法

粒子團演算法是從人類觀察鳥群時，所發現鳥群覓食的特殊現象，並且將其轉化為啟發式演算法 (Kennedy & Eberhart, 1995)。其原理為將每一個可能的解皆視為一顆粒子，粒子的個數可以依研究者自行決定，並且每顆粒子皆有記憶性且會參考其他粒子的位置，來修正自己的移動方向，藉此來趨向最佳解，如圖 2.6 所示：

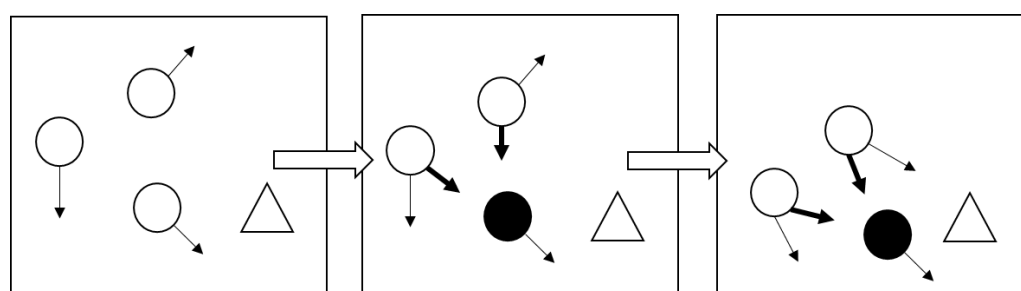


圖 2.6:粒子團演算法示意圖

圖中的方框即為可行解區，而每個圓則為一個可行解，黑色圓則為當前距離最佳解的粒子，三角形是最佳解的位置，意即在後設模型當中，一組輸出最近似於物理實驗輸出的一組解。而每一個圓圈的好壞則如式(2.6)的 MAPE 來衡量：

$$MAPE_{srgt/exp} = \frac{1}{N} \sum_{i=1}^N \left| \frac{Q_{srgt} - Q_{exp}}{Q_{exp}} \right|_i \quad (2.6)$$

上式當中 Q_{srgt} 為後設模型的輸出， Q_{exp} 為物理實驗的輸出，分別各有水流強度與水流速度，除此之外，每筆水流強度與速度又有兩種初始湍流度的差別。因此 N 若在未刪除模擬系統與物理實驗相差過大的距離點的情況下 N 為初始川流度乘上 d 乘上兩種輸出，但若給定初始湍流值與限定輸出為水流強度或速度的話，則 $N = d$ 。在刪除模擬系統與物理實驗相差過大的距離點的情況下，在固定初始湍流度為特定的距離點將被剔除因此 N 將會減少。

隨著代次的演進，每個粒子皆會藉由參考自身上一代次的移動方向(細箭頭)與當前最佳粒子的位置方向(粗箭頭)，來逐漸朝最佳解收斂，其更新的方式如式(2.7)。

$$V_i^t = V_i^{t-1} + C_1 * r * (L_i - x_i^{t-1}) + C_2 * r * (g - x_i^{t-1}) \quad (2.7)$$
$$x_i^t = x_i^{t-1} + V_i^t$$

式(2.7)中的 V_i^t 為第 t 期第 i 個粒子的移動方向， C_1 為以自身經驗做為參考的權重， C_2 為以最佳位置做為參考的權重， r 為一個隨機值， L_i 為第 i 顆粒子的自身曾經最佳位置， g 為所有粒子中的最佳位置， x_i^t 則為 t 期第 i 顆粒的位置。在本次實驗當中，每一個粒子即為一組 $C_{\epsilon 4}$ 、 $C_{\epsilon 5}$ 、 β_p 和 β_d ， g 則為最接近物理實驗下的水流強度 I 與速度 U 之解，意即 Q_{srgt} 與 Q_{exp} 間差距最小的一組解，隨著圖 2.7 中的流程圖，即可向最佳解收斂。

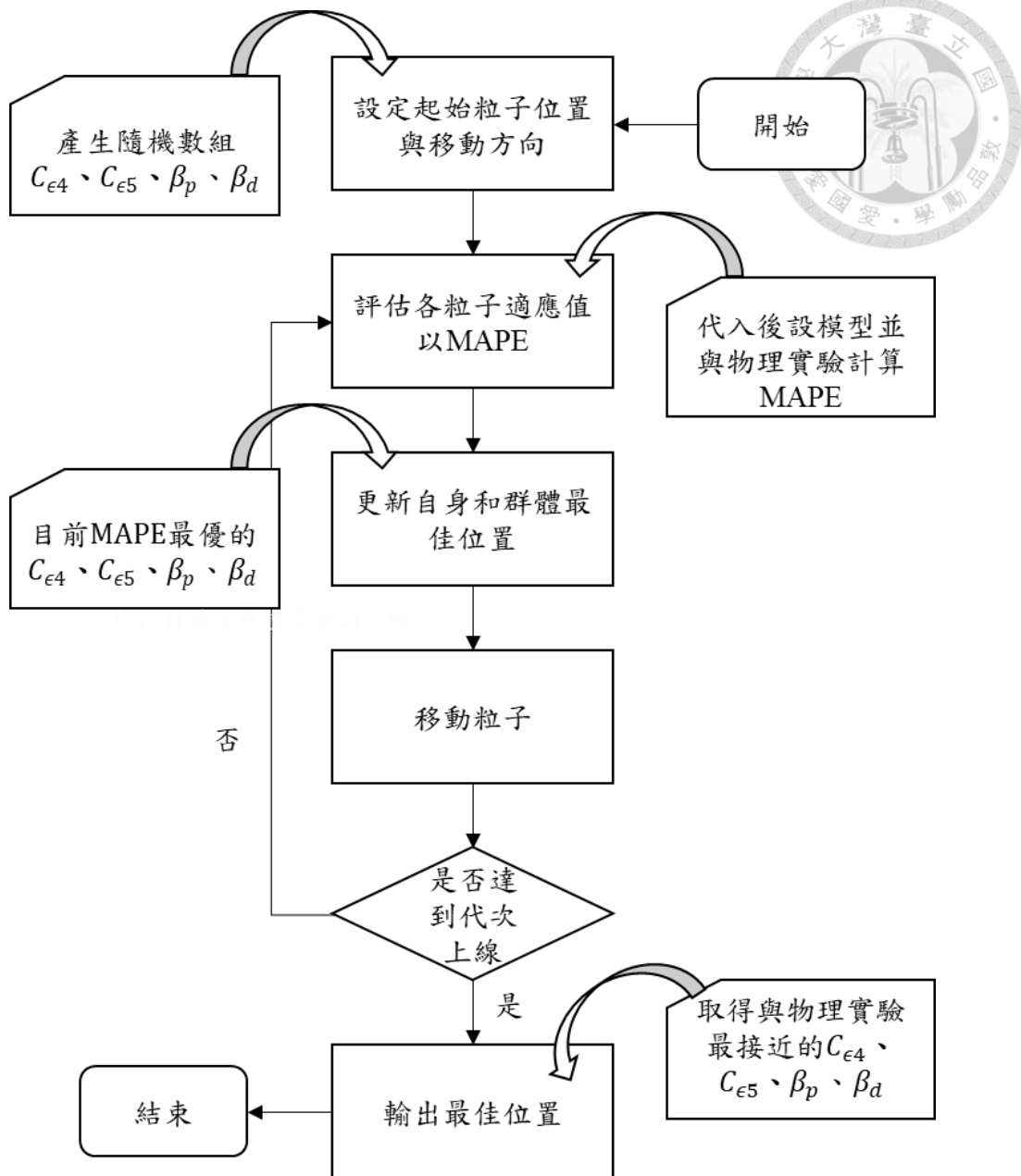


圖 2.7: 粒子團演算法示意圖



2.3.3 基因演算法

除了粒子團演算法，本次研究當中也使用了基因演算法。將每一組可行的 $C_{\epsilon 4}$ 、 $C_{\epsilon 5}$ 、 β_p 和 β_d 視為一條染色體，本研究所使用的基因演算法為二進制編碼，按本研究案例其編碼方式如圖 2.6 所示。決定二進位編碼長度(binary code length)為 n 後，將每個變數的上界 \bar{x} 及下界 \underline{x} 中切分為 2^n 段，依每個變數的標準化值進行分配，便完成了二進位轉換編碼，其解碼則反之亦然。

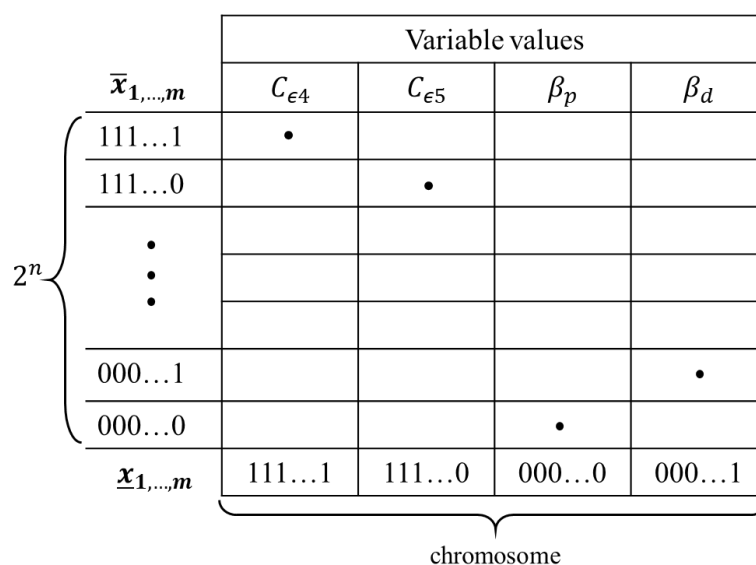


圖 2.6:基因演算法二元編碼

上圖中的變數即為本次研究的四個模擬系統參數值，一個參數值代表的二元編碼值代表一組基因，將所有變數排列便成為一條染色體，意即一組模擬系統的參數解。每一條染色體代入後設模型 $g(x)$ 後得到預期水流強度與速度，與實際物理實驗值計算 MAPE 後做為適應值如式(2.6)所示。由於本次研究期望為找出最能近似物理實驗的一組最佳化參數解，因此適應值 MAPE 為越小越好，其計算方式為適應值倒數後進行計算，適應值低則選為複製交配的染色體機率越高。此外還需決定突變機率來防止陷入區域最佳解，計算流程如圖 2.7 所述，最終目標為找出最能解釋物理實驗之後設模型中的一組解。

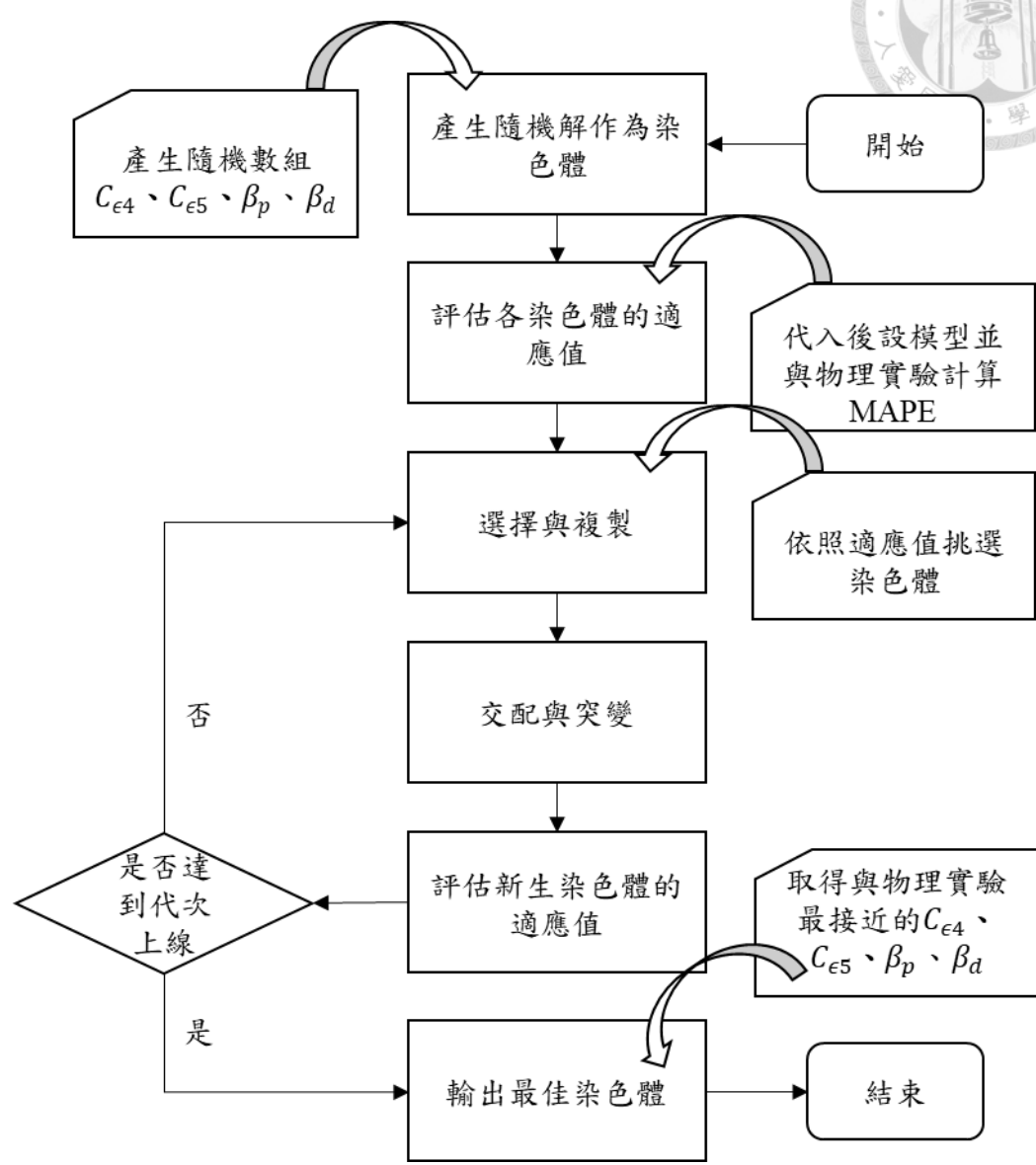


圖 2.8:基因演算法示意圖

第三章 數值分析



在此一章節當中，將會分別就模型參數設置，後設模型擬合誤差(Surrogate-model Fitting Error)、預測誤差(Predicted Error)、最佳解擬合誤差(Optimal Solution Fitting Error)與最終誤差(Final Error)等順序，將本次研究的案例代入離線最佳化流程當中，其中後設模型擬合誤差與最佳擬合誤差，在離線最佳化的流程當中，皆設為需小於 10%。期望在能縮短計算時間下，能同時不失卻準確度。

3.1 模型參數設置與樣本分布

本研究將訓練集共 81 組的樣本下，切分的方式使用了 K-fold 法將樣本切分為五等分，意即 $K = 5$ ，此外共有 16 組的樣本作為測試集。此外初始湍流強度共有 3%與 15%兩種，距離點 d 分為將所有距離點納入模型的案例跟刪除模擬系統與物理實驗輸出過遠距離點的案例， d 分別為 10 與刪除 4、5、6 的剩餘 7 點。因此在建立類神經網路模型作為後設模型時，所放入的參數共有六項，其架構如表 3.2 所示。此外本研究也有嘗試其他的結構，當神經元數過少時，在初始湍流強度 3%下的水流強度，其預測的 MAPE 表現皆不佳，本次嘗試的結構按層數排列神經元數分別有 6, 12, 12, 12, 6, 6 與 12, 24, 24, 24, 12, 12 與 24, 48, 48, 48, 24, 24 和最終本次實驗的架構，按標號標為模型 1、模型 2 等進行比較，代次皆為 250 次，如表 3.1 所示。

粒子團演算法粒子個數為 2000，參考權重分別為 0.6 與 0.5，代次在約莫 30 左右變會停止，原因為改善程度已在最低改善步距 0.0001 以下。基因演算法染色體共 100 條，代次上限設為為 1000，交配率與突變率分別為 0.7 與 0.1，二元編碼長度為 10，兩種演算法其變數上界皆為 $[\overline{C_{\epsilon 4}}, \overline{C_{\epsilon 5}}, \overline{\beta_p}, \overline{\beta_d}] = [1, 4, 1, 4]$ ，下界為 $[\underline{C_{\epsilon 4}}, \underline{C_{\epsilon 5}}, \underline{\beta_p}, \underline{\beta_d}] = [0.1, 0.1, 0.1, 0.1]$ 。

表 3.1:模型比較表

結構	神經元數量	I_3 之 MAPE
模型 1	6, 12, 12, 12, 6, 6	6.76%
模型 2	12, 24, 24, 24, 12, 12	18.35%
模型 3	24, 48, 48, 48, 24, 24	3.97%
最終模型	128, 64, 64, 64, 32, 16	2.81%

表 3.2:類神經網路架構

Layer	Nodes	Transformation function
Input	6	
Hidden 1	128	ReLu
Hidden 2	64	ReLu
Hidden 3	64	ReLu
Hidden 4	64	ReLu
Hidden 5	32	ReLu
Hidden 6	16	ReLu
Output	2	ReLu

Loss function: Mean Absolute Percentage Error

樣本的分布如圖 3.1 到圖 3.3 所示，分別就兩種演算法與有無去除 4、5、6 距離點來進行展示，因此共有三張圖。由於本研究的離線最佳化流程，會先將先前最佳解擬合誤差未達 10% 以下的結果，放回到訓練集中來增加訓練樣本，藉此對最佳解周圍的反應曲面進行更好的描述，因此圖 3.1 到圖 3.3 中會標示出是第幾次最佳化所加入的樣本，以供對物理實驗點進行比較，觀察是否有無逐漸靠向物理實驗。

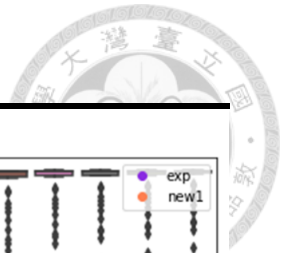


圖 3.1: 基因演算法模擬系統樣本箱型圖

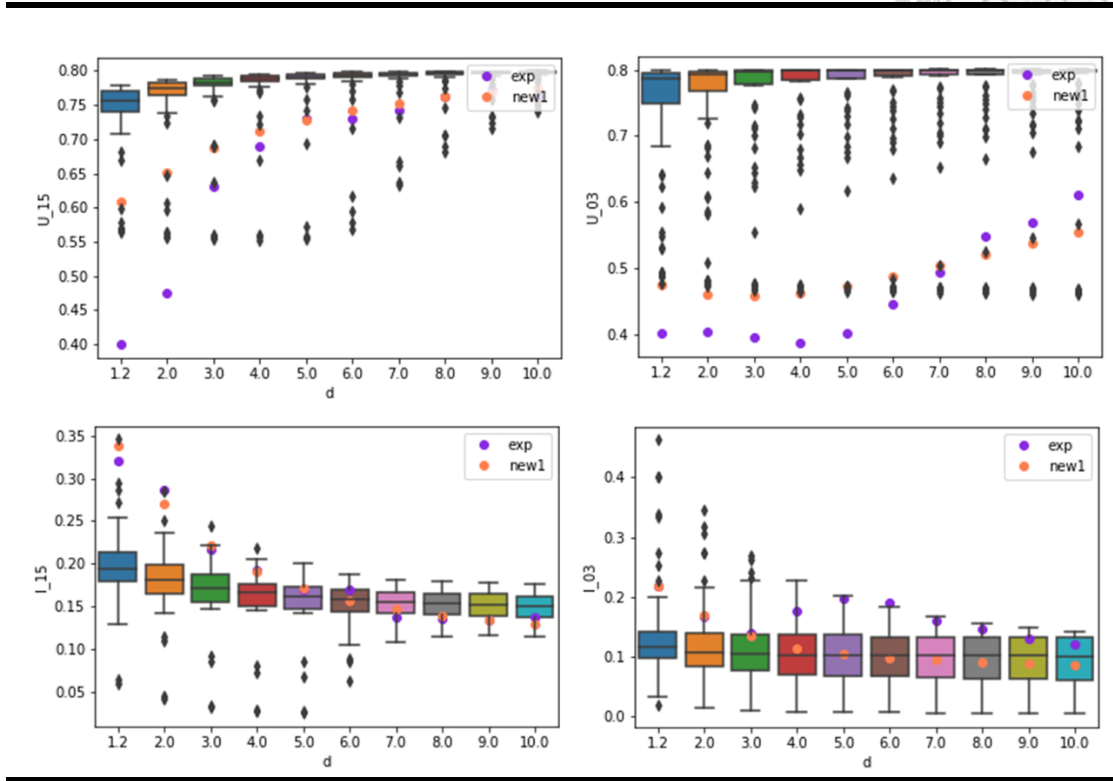


圖 3.2: 粒子團演算法模擬系統樣本箱型圖

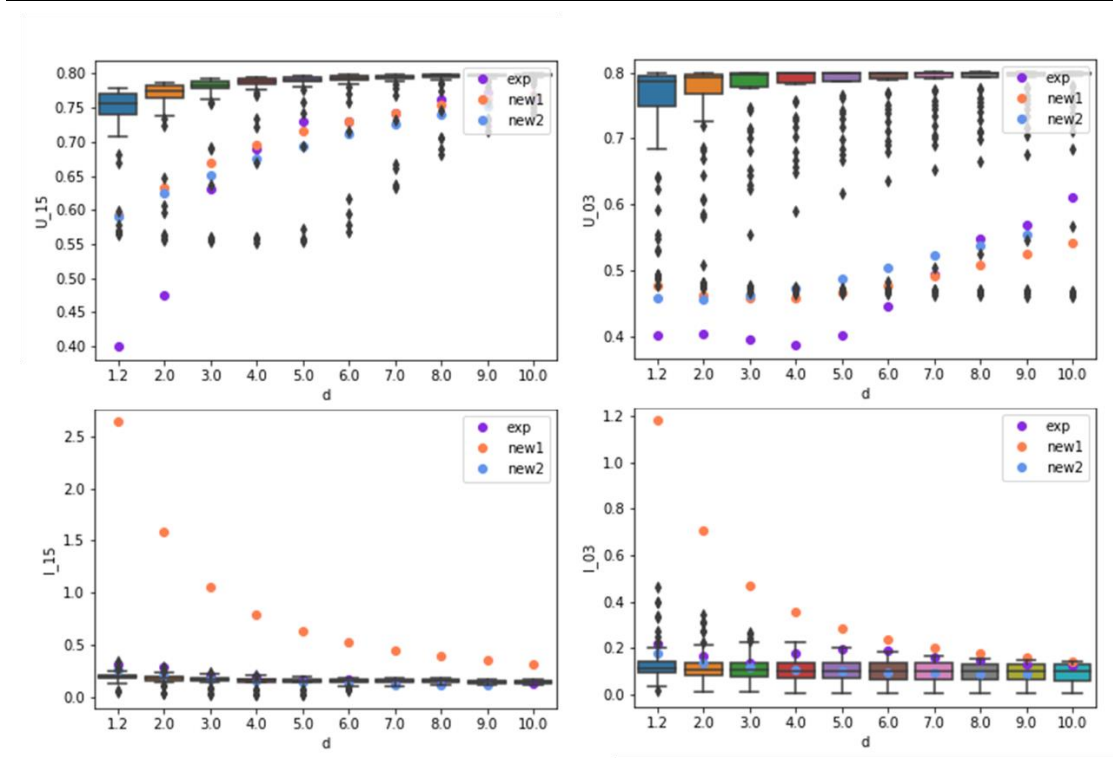
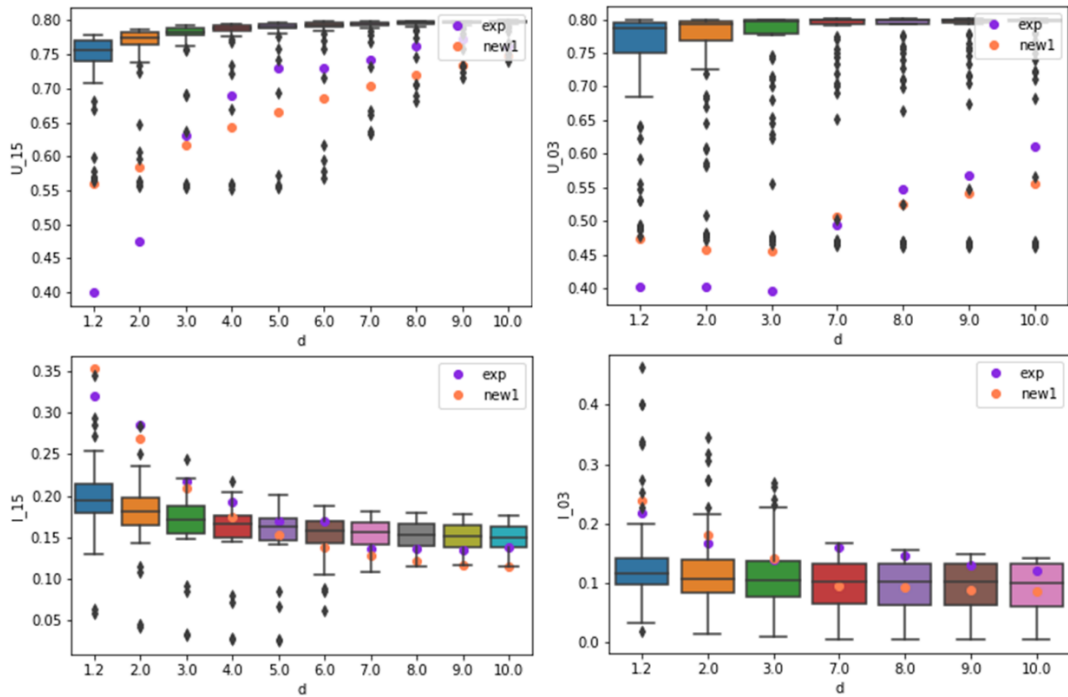


圖 3.3: 粒子團演算法模擬系統樣本去除 4、5、6 距離點箱型圖



最後最佳化的所耗費的時間如表 3.3，可看出其實耗費的時間皆不長。其中基因演算法在無去除距離點下達到 2768 多秒是由於跑了 1000 代次。但事後發先其實跑 100 代次便可得到差不多品質的解，因此在往後的研究當中可考慮減少代次。

表 3.3: 運算時間

演算法	耗費秒數
粒子團演算法	172.55
基因演算法	2768.28
粒子團演算法無 4、5、6 距離點	112.95



3.2 後設模型擬合誤差

建設模型時需衡量一個模型是否準確，我們將會以測試集的資料來進行衡量。本次實驗中共有 97 組的資料，從當中抽取 16 組作為測試集，且這 16 組並不會進入模型進行訓練。在這一步驟的目標是要將後設模型擬合誤差低於 10%，計算方式為 MAPE 如式(3.1)。

$$MAPE_{srgt/sim} = \frac{1}{|D|} \frac{1}{|M|} \sum_{d=1.2}^{|D|} \sum_{m=1}^{|M|} \frac{|\hat{y}_{d,m} - y_{d,m}^S|}{y_{d,m}^S} \quad (3.1)$$

$$d \in D = \{1.2, 2, 3, \dots, 8, 9, 10\}$$

$$m \in M = \{1, 2, 3, \dots, 14, 15, 16\}$$

當中的 d 與 m 分別為距離點與組別， \hat{y} 與 y^S 分別為後設模型的輸出和模擬系統的輸出。此一誤差著重於建立一個能擬合模擬系統的模型，並且驗證所建立的模型在測試集下是否能有效模擬模擬系統。而我們將分為在粒子團演算法與基因演算法下，分別對有無去除距離點 4、5、6 進行探討，如表 3.4 與表 3.5 所示。

表 3.4:後設模型擬合誤差在無去除距離點下

	粒子團演算法	基因演算法
$I_{3\%}$	2.81%	2.19%
$I_{15\%}$	1.40%	1.17%
$U_{3\%}$	1.23%	0.64%
$U_{15\%}$	0.40%	0.38%
平均	1.46%	1.095%



表 3.5:後設模型擬合誤差在去除距離點(4、5、6)下

粒子團演算法	
$I_{3\%}$	2.77%
$I_{15\%}$	0.65%
$U_{3\%}$	1.07%
$U_{15\%}$	0.38%
平均	1.22%

在上述兩張表中，誤差平均皆遠低於標準所設置的 10%，且刪除在初始湍流值 3% 下的距離點 4、5、6 後，後設模型擬合誤差並沒有明顯的下降甚至是上升，代表後設模型在現有的歷史資料 81 組當中，不論是否刪減模擬系統與物理實驗相距較遠的距離點，皆能夠有效的被訓練出具有對這 81 組資料有良好的解釋能力的模型。可能的原因是雖然在此步驟下，由於是要擬合模擬系統，因此刪除模擬系統與物理實驗誤差大的資料，反而減少了訓練後設模型時所需的訓練集資料。

3.3 預測誤差

當有了一個具有解釋能力的模型後，就是要在其中找出一組最佳近似解能解釋物理實驗的結果，以便再代回模擬系統後進行對物理實驗的預測。因此在最佳化的過程當中，必須取與物理實驗結果間的誤差極小化，作為演算法過程中的目標函數，而本次研究所使用的目標函數為 MAPE 如式 2.5。其結果如所示表，分為有去掉模擬系統與離物理實驗差距較大的距離點與無去除距離點。

表 3.6:各模型預期誤差

	$I_{3\%}$	$I_{15\%}$	$U_{3\%}$	$U_{15\%}$	Average
粒子團演算法	26.86%	5.74%	11.89%	10.76%	13.81%
基因演算法	29.45%	7.71%	11.65%	14.00%	15.70%
粒子團去除過遠距離點(4、5、6)	21.74%	8.33%	10.50%	13.26%	12.99%

從表 3.6 可看出，去除掉模擬系統與物理實驗輸出相距過遠的距離點後，模型的預期誤差有下降，其原因為去除相距過遠的距離點後，便降低了最佳化的難度。此一誤差在本次研究中的最佳化流程中，並無訂定是否需低於 10% 的標準。在後設模型對模擬系統已經具有相當良好的解釋能力時，計算預期誤差才會有意義，預期誤差除了示為了找出最近似於物理實驗的後設模型參數解之為，也能作為對最終誤差的預測，因此越示擬合模擬系統的後設模型，其愈其誤差與最終誤差的值將會越示接近。此外表的結果皆已是額外加入最佳化後但未通過最佳解擬合誤差的標準的參數解，因此樣本已較原先的 81 組多，且對於最佳解附近的反應曲面有著更加詳盡的描述了，因此更能找出近似於物理實驗的最佳化參數解。

3.4 最佳解擬合誤差

最佳化參數代回模擬系統後，將會獲取模擬系統的輸出，與後設模型的輸出比較後，將會得到最佳解擬合誤差，誤差公式如式(3.2)。本研究設定最佳擬合誤差需低於 10%，才能進行最終誤差的計算。此一誤差為衡量後設模型是否具有替代模擬系統進行離線最佳化的可能，若誤差過高，代表後設模型的預計值與模擬系統的預計值差距過大，後設模型無法替代模擬系統進行離線最佳化，反之則較具有替代模擬系統進行離線最佳化的可能。

$$MAPE_{srgt/sim} = \frac{1}{N} \sum_{i=1}^N \left| \frac{Q_{srgt} - Q_{sim}}{Q_{sim}} \right|_i \quad (3.2)$$



式(3.2)與式(2.6)相似，僅將 Q_{exp} 改為 Q_{sim} ，其 N 的定義方式與式(2.6)相同。在本次研究當中最佳解擬合誤差也分為有去除模擬系統與物理實驗相距過遠的距離點與無去除模擬系統與物理實驗相距過遠的距離點，其結果如表所示。

表 3.7:各模型最佳解擬合誤差

$MAPE_{srgt/sim}$	$I_{3\%}$	$I_{15\%}$	$U_{3\%}$	$U_{15\%}$	Average
粒子團演算法	10.87%	1.00%	3.15%	2.35%	4.34%
基因演算法	9.29%	6.78%	4.44%	1.45%	5.50%
粒子團去除過遠 距離點(4、5、6)	3.88%	5.12%	1.25%	3.20%	3.50%

從表 3.7 可看出各模型最佳解擬合誤差皆遠低於 10%，代表後設模型對於模擬系統具有一定的解釋能力。利用此一具有解是能力的後設模型，進行最終誤差的計算，來替代對整個模擬系統進行線上最佳化。

3.5 最終誤差

除了將最佳化後的解代入模擬系統與後設模型做比較，最終依然要與物理實驗進行驗證離線最佳化的效果是否能取代線上最佳化。若利用後設模型最佳化後的參數代入模擬系統中所取得的解，與物理實驗相當接近，則代表利用人工類神經網路進行離線最佳化是可行的，衡量的誤差方式如式(3.3)。本次研究的最終誤差分為有無去除模擬系統離物理實驗點過遠的結果，如表 3.8 所示。

$$MAPE_{sim/exp} = \frac{1}{N} \sum_{i=1}^N \left| \frac{Q_{sim} - Q_{exp}}{Q_{exp}} \right|_i \quad (3.3)$$



表 3.8: 各模型最終誤差

	$I_{3\%}$	$I_{15\%}$	$U_{3\%}$	$U_{15\%}$	Average
粒子團演算法	28.62%	$I_{15\%}$	12.37%	9.49%	14.21%
基因演算法	27.08%	4.74%	11.72%	9.31%	13.21%
粒子團去除過遠 距離點(4、5、6)	22.66%	10.22%	9.52%	9.85%	12.53%

從表 3.8 可看出去除模擬系統離物理實驗過遠的結果，的確能降低最終誤差。其原因為模擬系統與物理實驗再初始湍流值 3% 時，模擬系統與物理實驗數值相差過大。但縱然不刪除與物理實驗過遠的點，也能獲得令人可接受的誤差結果，其最終比較圖表分別為如圖 3.4、圖 3.5、圖 3.6 所示。

圖 3.4: 粒子團演算法模擬系統與物理實驗誤差

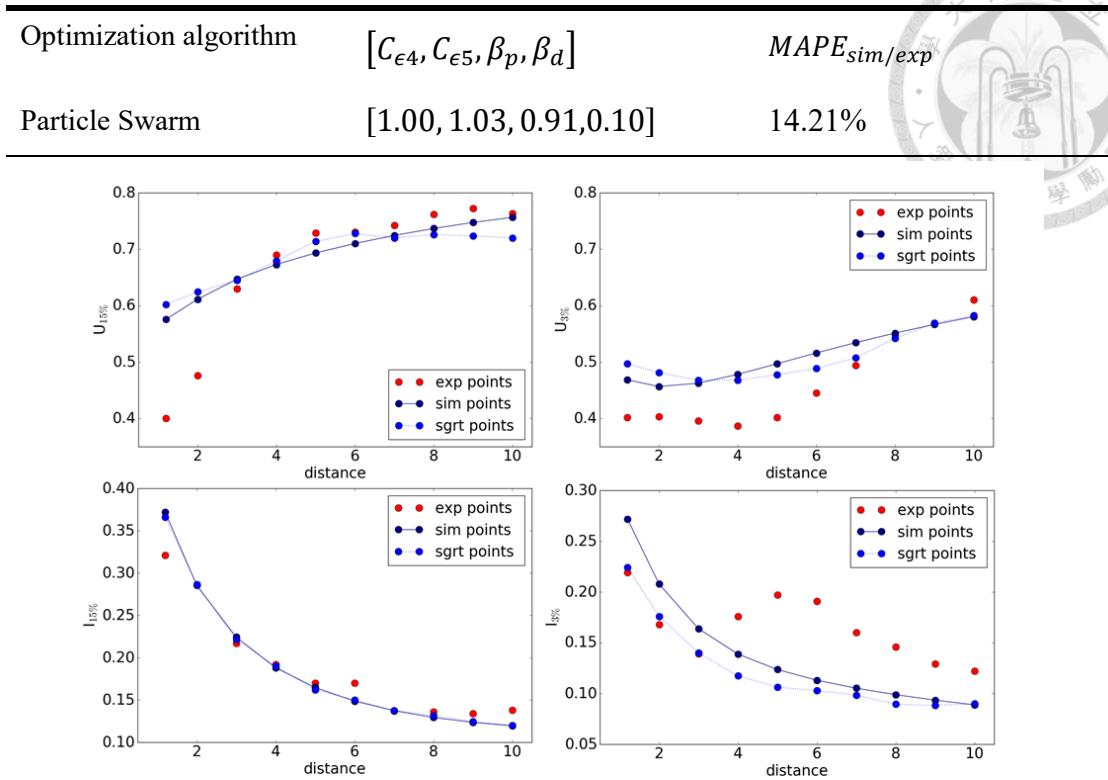


圖 3.5: 基因演算法模擬系統與物理實驗誤差

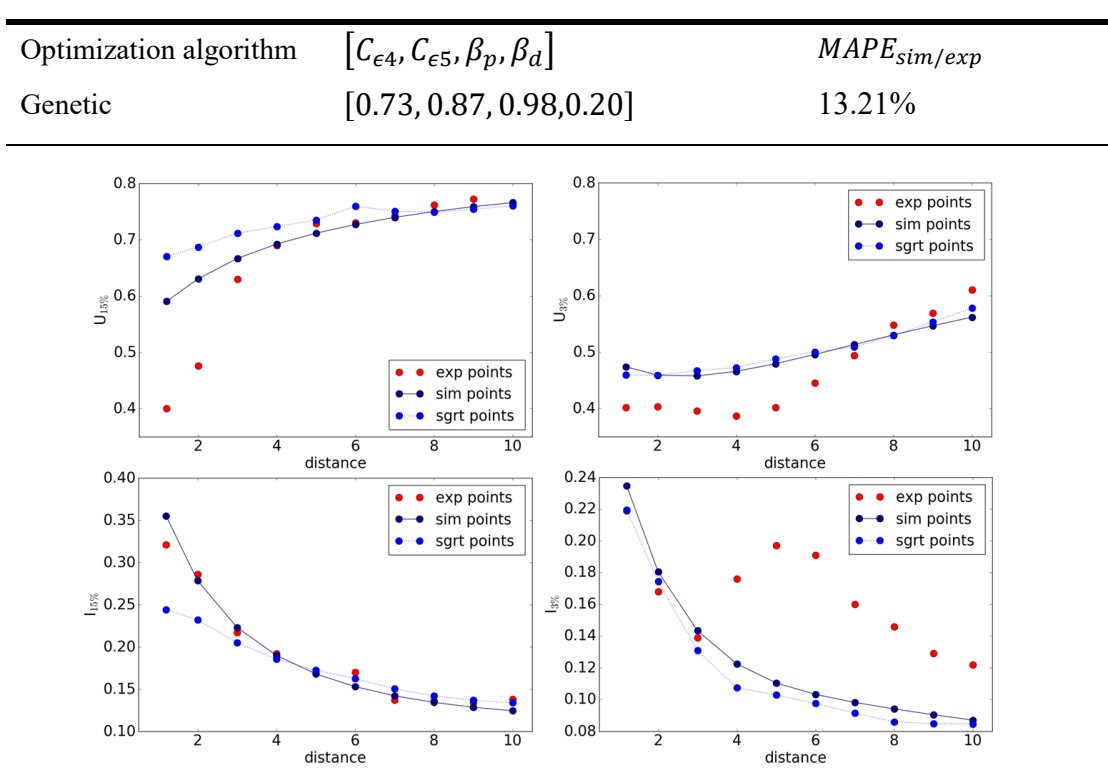
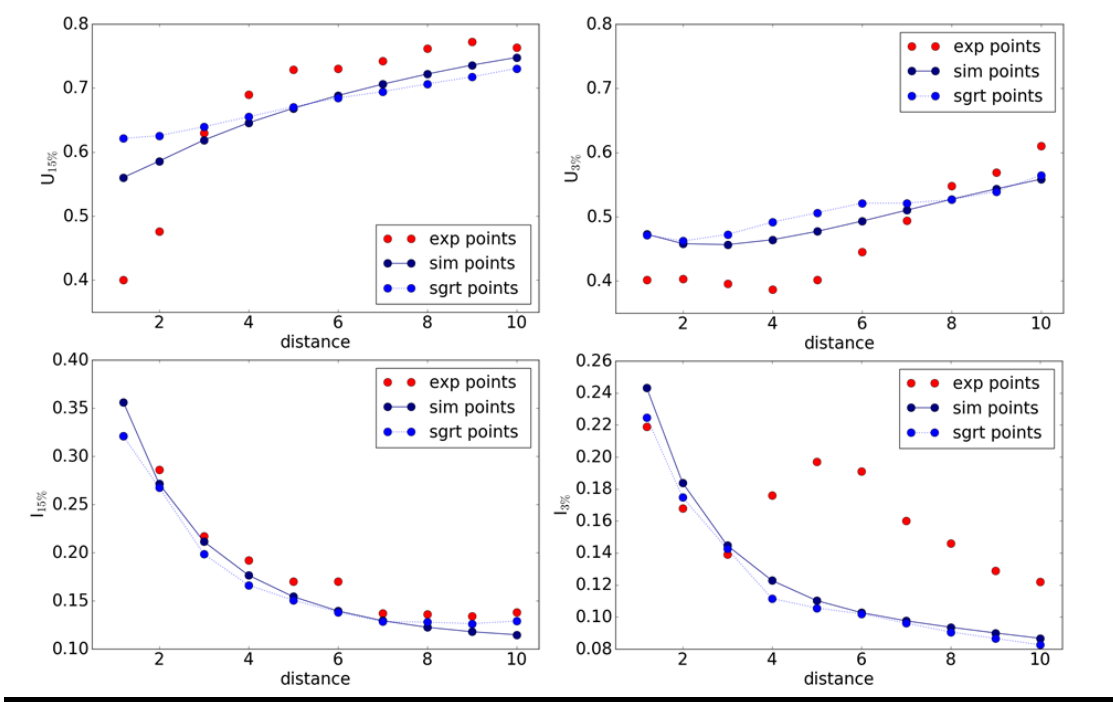




圖 3.6: 粒子團演算法模擬系統與物理實驗在去除過遠距離下誤差

Optimization algorithm	$[C_{\epsilon 4}, C_{\epsilon 5}, \beta_p, \beta_d]$	$MAPE_{sim/exp}$
Particle Swarm	$[1.00, 0.80, 1.00, 0.10]$	14.90%



由於最佳解擬合誤差不管在何種情境皆已低到近似 5%，因此在許多情況下後設模型與模擬系統會處於近乎重疊的情況，意即後設模型已經充分的擬合了模擬系統。而後設模型與模擬系統在初始湍流值 3% 的情況下，對水流強度依然預測十分不準確，因此線和點依然存在著明顯的差距，最終誤差也因為這樣而被拉高，雖然去除距離點 4、5、6 後在初始湍流值 3% 的水流強度最終誤差皆有降 5% 以上的誤差，但依然高於 20% 以上，其問題明顯在於模擬系統在這種情況下無法準測預測物理實驗結果，因此後設模型即便再擬合模擬系統，也無法預測這情況下的物理實驗值。

第四章 結論



在本次的研究中，從原本將模擬系統線上最佳化時，所需耗費的大量時間，藉由離線最佳化大幅縮短，原先線上最佳化跑一個代次所需花費的時間，便可完成離線最佳化所有的流程。在減少了運算的耗能後，也並無損失最終誤差的精確度，最後達到了在後設模型與模擬系統擬合誤差皆低於 5%的情況下，最終誤差僅有 13%至 14%。美中不足的是模擬系統原先在初始湍流值 3%的情況下，對距離點 4、5、6 本身預測便不準確，因此縱然後設模型高度擬合模擬系統，最佳化後的參數代入模擬系統後，仍然無法與物理實驗近似

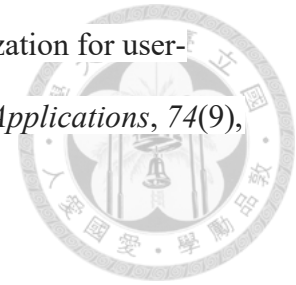
未來若能藉由改善模擬系統，提供在初始湍流值 3%時的準確預測數據，或許就能解決後設模型最佳化後的參數解，代入模擬系統後依然無法近似物理實驗的問題。另一方面若要在降低最終誤差，往後可調降最佳解擬合誤差的門檻，使離線最佳化的流程進行更多的代次，以此增加更多最佳解附近的樣本點，使準確度提升。在後設模型最佳化方面，也可選擇其他最佳化演算法進行比較。

參考文獻



- Barry, D. A. (1990). Supercomputers and their use in modeling subsurface solute transport. *Reviews of Geophysics*, 28(3), 277-295.
- Barton, R. R., & Meckesheimer, M. (2006). Metamodel-based simulation optimization. *Handbooks in operations research and management science*, 13, 535-574.
- Bengio, Y., & Grandvalet, Y. (2004). No unbiased estimator of the variance of k-fold cross-validation. *Journal of machine learning research*, 5(Sep), 1089-1105.
- Chau, K. W. (2007). Application of a PSO-based neural network in analysis of outcomes of construction claims. *Automation in construction*, 16(5), 642-646.
- Choi, E., Bahadori, M. T., Schuetz, A., Stewart, W. F., & Sun, J. (2016, December). Doctor ai: Predicting clinical events via recurrent neural networks. In *Machine Learning for Healthcare Conference* (pp. 301-318).
- Cui, Y., Olsen, K. B., Jordan, T. H., Lee, K., Zhou, J., Small, P., ... & Levesque, J. (2010, November). Scalable earthquake simulation on petascale supercomputers. In *SC'10: Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis* (pp. 1-20). IEEE.
- Djurfeldt, M., Lundqvist, M., Johansson, C., Rehn, M., Ekeberg, O., & Lansner, A. (2008). Brain-scale simulation of the neocortex on the IBM Blue Gene/L supercomputer. *IBM Journal of Research and Development*, 52(1.2), 31-41.

Dooms, S., De Pessemier, T., & Martens, L. (2015). Offline optimization for user-specific hybrid recommender systems. *Multimedia Tools and Applications*, 74(9), 3053-3076.



Eisenhower, B., O'Neill, Z., Narayanan, S., Fonoberov, V. A., & Mezić, I. (2012). A methodology for meta-model based optimization in building energy models. *Energy and Buildings*, 47, 292-301.

Fonseca, D. J., Navarrese, D. O., & Moynihan, G. P. (2003). Simulation metamodeling through artificial neural networks. *Engineering Applications of Artificial Intelligence*, 16(3), 177-183

In *Proc. 8th International Workshop on Large-Scale Integration of Wind Power into Power Systems. Bremen, Germany* (pp. 472-478)

James, S. C., Johnson, E. L., Barco, J., & Roberts, J. D. (2017). Simulating current-energy converters: SNL-EFDC model development, verification, and parameter estimation. *Renewable Energy*.

Kennedy, J., & Eberhart, R. (1995, Nov 27). Particle swarm optimization. *IEEE ICNN'95 - International Conference on Neural Networks*, pp. 1942 - 1948 vol.4

Larose, C., Gagnon, R., Turmel, G., Giroux, P., Brochu, J., McNabb, D., & Lefebvre, D. (2009, October). Large wind power plant modeling techniques for power system simulation studies.

Madu, C. N. (1990). Simulation in manufacturing: a regression metamodel approach. *Computers & Industrial Engineering*, 18(3), 381-389.

Mehdad, E., & Kleijnen, J. P. (2018). Efficient global optimisation for black-box simulation via sequential intrinsic Kriging. *Journal of the Operational Research Society*, 69(11), 1725-1737



Mirjalili, S., Hashim, S. Z. M., & Sardroudi, H. M. (2012). Training feedforward neural networks using hybrid particle swarm optimization and gravitational search algorithm. *Applied Mathematics and Computation*, 218(22), 11125-11137

Palopoli, L., Passerone, R., & Rizano, T. (2011). Scalable offline optimization of industrial wireless sensor networks. *IEEE Transactions on Industrial Informatics*, 7(2), 328-339.

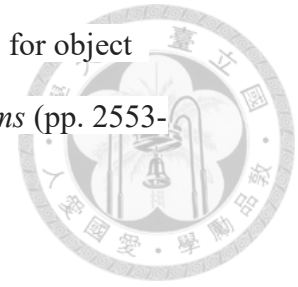
Ribau, J., Viegas, R., Angelino, A., Moutinho, A., & Silva, C. (2014). A new offline optimization approach for designing a fuel cell hybrid bus. *Transportation Research Part C: Emerging Technologies*, 42, 14-27.

Rodriguez, J. D., Perez, A., & Lozano, J. A. (2009). Sensitivity analysis of k-fold cross validation in prediction error estimation. *IEEE transactions on pattern analysis and machine intelligence*, 32(3), 569-575.

Simpson, T. W., Poplinski, J. D., Koch, P. N., & Allen, J. K. (2001). Metamodels for computer-based engineering design: survey and recommendations. *Engineering with computers*, 17(2), 129-150.

Sterling S. Olson, Jack C.P. Su, H. S. (2019). Turbulence-parameter estimation for currentenergy converters using surrogate model optimization.

Szegedy, C., Toshev, A., & Erhan, D. (2013). Deep neural networks for object detection. In *Advances in neural information processing systems* (pp. 2553-2561).



Valian, E., Mohanna, S., & Tavakoli, S. (2011). Improved cuckoo search algorithm for feedforward neural network training. *International Journal of Artificial Intelligence & Applications*, 2(3), 36-43.

Vlahogianni, E. I., Karlaftis, M. G., & Golias, J. C. (2005). Optimized and meta-optimized neural networks for short-term traffic flow prediction: A genetic approach. *Transportation Research Part C: Emerging Technologies*, 13(3), 211-234.

洪唯凱 (民 107). 電腦模擬系統下離線最佳化架構中後設模型和搜尋演算法之應用. 臺灣大學工業工程學研究所學位論文, 1-39.

陳陽益、許弘莒、蘇超偉、薛憲文、白俊彥、楊瑞源、李孟學、許城榕 (民 104). 貳級黑潮發電先導機組實海域船拖測試. 第 37 屆海洋工程研討會論文集, 739-744.