



國立臺北科技大學

工業工程與管理系碩士班

碩士學位論文

透過 PNDQN 求解司機隱性偏好之車輛途  
程問題

Optimizing Vehicle Routing by Incorporating Driver  
Preferences: A PNDQN Approach

研究生：詹郁琪

指導教授：鄭辰仰 博士

中 華 民 國 一 百 一 十 二 年 七 月

國立臺北科技大學  
研究所碩士學位論文口試委員會審定書

本校 工業工程與管理系 研究所 詹郁琪 君

所提論文，經本委員會審定通過，合於碩士資格，特此證明。

學位考試委員會

委

員：

陳 盈 彥

李 淑 芬

鄭 辰 仰

指導教授：

鄭 辰 仰

所 長：

黃 乾 佑

中 華 民 國    一 百 一 十 二   年   七   月   十 五   日

# 摘要

論文名稱：透過 PNDQN 求解司機隱性偏好之車輛途程問題

頁數：42

校所別：國立臺北科技大學 管理學院 工業工程與管理系 碩士班

畢業時間：一百一十一學年度 第二學期

學位：碩士

研究生：詹郁琪

指導教授：鄭辰仰 博士

車輛途程問題(Vehicle Routing Problem, VRP)為實務上常見的物流配送問題。現今大多數學者皆為了找出最佳路線解，卻忽視了司機因交通路況而改變路線之隱性偏好，因此最佳解並非在當前狀況中的最優解。在本文中將提出基於注意力機制之指針網路結合深度 Q 網路模型(an attention-based Pointer Network and Deep Q Network, PNDQN)，為了將司機過去行駛偏好路線經驗作為訓練資料，並考慮到新站點增加的問題。具體來說，本研究將使用具有注意力機制的指針網路來學習司機過去行駛經驗路線的特徵，進行預測後的路線結果會成為後續深度強化學習的環境。再接著運用深度 Q 網路根據制定好的策略，將新站點加入到路線解當中，求得最終當前最優解。

實驗結果證實，PNDQN 能有效的預測出司機過去經驗解，對於新站點問題進行比較後，在執行時間與路線結果中皆遠勝於其餘演算法。路線結果在六種情況中皆節省其餘演算法 19.09%路線長度，運算時間則節省了 82.26%。本研究可協助公司學習到物流司機經驗，將經驗量化後可傳承給新進員工，減少員工訓練時間，達到經驗傳承的目的。

關鍵詞：車輛途程問題、指針網路、深度強化學習、隱性偏好、司機經驗量化

# ABSTRACT

Title: Optimizing Vehicle Routing by Incorporating Driver Preferences: A PNDQN Approach

Pages: 42

School: National Taipei University of Technology

Department: Industrial Engineering of Management, College of Management

Time: July, 2023

Degree: Master

Researcher: Chan, Yu Chi

Advisor: Chen-Yang Cheng, Ph.D.

The Vehicle Routing Problem (VRP) is a prevalent logistics distribution issue. However, most researchers focus solely on finding the optimal route without considering drivers' implicit preferences, which can change due to traffic conditions. This paper introduces the attention-based Pointer Network and Deep Q Network (PNDQN) model, which leverages drivers' past route preference experiences as training data and addresses adding new stations. The study utilizes a pointer network with an attention mechanism to learn the features of drivers' past routes. It employs the Deep Q Network to incorporate new stations based on a predefined strategy. Experimental results demonstrate that PNDQN effectively predicts drivers' past experienced routes and outperforms other algorithms in terms of execution time and route quality for new station problems. The PNDQN model reduces route length by 19.09% compared to other algorithms in all six scenarios and achieves an 82.26% reduction in computation time. This research facilitates knowledge transfer from experienced drivers to new employees, reducing training time and ensuring experience inheritance in logistics operations.

Keywords: Vehicle Routing Problem, Pointer Network, Deep Reinforcement Learning, Implicit Preference



# 目錄

摘要.....	i
ABSTRACT.....	ii
目錄.....	iv
表目錄.....	vi
圖目錄.....	vii
第一章    緒論.....	1
1.1    研究背景與動機.....	1
1.2    研究目的.....	2
1.3    研究架構.....	3
第二章    文獻探討.....	4
2.1    車輛途程問題.....	4
2.2    指針網路.....	6
2.3    注意力機制.....	7
2.4    深度強化學習.....	8
第三章    研究方法.....	12
3.1    研究架構.....	12
3.2    數學模型.....	13
3.3    以指針網路學習司機隱性偏好路線.....	16
3.4    深度強化學習結合指針網路之 PNDQN 模型.....	20
3.5    實驗過程.....	23
第四章    實驗結果.....	26
4.1    情境一：求解具有歷史經驗城市點實驗結果.....	26
4.2    情境二：求解歷史經驗城市點與新站點實驗結果.....	33

4.3	案例說明與實驗結果.....	35
第五章	結論.....	37
5.1	研究結果與管理內涵.....	37
5.2	研究限制與未來發展.....	38
	參考文獻.....	39



## 表目錄

表 1 符號定義 .....	14
表 2 透過指針網路學習司機送貨經驗之虛擬碼 .....	19
表 3 訓練 1 類題庫模型評估表 .....	28
表 4 訓練 2 類題庫模型評估表 .....	28
表 5 Solomon 56 題國際標竿例題文獻已知最佳解 .....	29
表 6 PNDQN 模型對於 Solomon(1987)56 題標竿例題預測結果 .....	31
表 7 ACO、PSO、PNDQN 執行時間比較表 .....	32
表 8 PNDQN 模型測試 R101 之獎勵值收斂圖 .....	33
表 9 PNDQN、ACO、PSO 加入新站點比較表 .....	34
表 10 與現有論文結果比較表 .....	35





## 圖目錄

圖 1 強化學習架構圖 .....	9
圖 2 研究架構圖 .....	13
圖 3 傳統 Seq2Seq 模型 .....	17
圖 4 使用 PN 求解車輛途程問題範例 .....	18
圖 5 資料擴增示意圖 .....	20
圖 6 本研究於 DQN 各項元素範例圖 .....	22
圖 7 本研究 PNDQN 架構圖 .....	23
圖 8 情境一：使用 PN 求解具有歷史資料站點路線 .....	24
圖 9 情境二：使用 PNDQN 模型求解具有歷史資料與從未行駛城市集合點路線 .....	24



# 第一章 緒論

## 1.1 研究背景與動機

對於全球供應鏈當中，非常注重全面且迅速的了解各地消費者各種需求，並針對消費者指定計畫、協調、控制等，其中最需要注意項目為物流運輸。物流運輸扮演了供應端與消費端之間的橋樑，現今依靠著現代網路技術支援，實現了物流的整體化與快速效率。隨著科技的進步，物流運輸業更需要因應環境的變動來調整物流的分配與效率。物流成本包括了運輸成本、存貨持有成本、物流行政管理成本等，因此若能找出物流運輸路線的最佳路徑將會節省許多的運輸成本。

為了找出物流運輸路線最佳路徑，在過去的議題當中最經典也是目前最符合實務物流的問題就是車輛途程問題(Vehicle Routing Problem, VRP)。VRP 是經典的組合優化和整數規劃問題，也是 NP-Hard 問題。VRP 是由 Dantzig 和 Ramser(1959)所提出，在客戶集合中有不同的需求，給定配送具有容量限制的車子集合，組織適當的配送路線，讓車輛能有序地將貨物配送至客戶手上。

在過去的研究中，已經有許多的學者研究過相關議題，像是 Rathore 等人(2020)透過基因演算法(Genetic Algorithm, GA)解決具有軟時間窗的多站點取送貨 VRP、Daely 等人(2021)使用粒子群最佳化(Particle Swarm Optimization, PSO)解決動態 VRP 來生成及時路線解、Sheng 等人(2020)運用了指針網路(Pointer Networks, PN)來解決具有有限資源且有站點選擇優先順序之 VRP。但以上研究僅探討出理論上的最佳解，並非有考慮到實際路線情況與司機行為等因素，因此在實務上可能會造成司機行駛路線比理論上最佳解路線還要省時、省成本。

根據上述觀點，Srivatsa Srinivas 和 Gajanand(2017)說過在規劃車輛路線時，司機行為往往被忽視，像是交通擁堵程度、駕駛員疲勞等因素都會影響司機行駛路線選擇。Chen 等人(2021)說道，由於現實生活中的目標或成本矩陣非常複雜，因此具有經驗的

司機會比演算法所算出的最佳送貨路線還要更好。但若出現了新站點，而該站點沒有任何過去經驗，應遵循何種策略將該點結合到過去經驗內。所以既考慮司機過去經驗又可找出最佳路線解是本研究需要探討的問題主軸。

對於深度學習(Deep Learning, DL)應用於 VRP 的研究，Sheng 等人(2020)使用了 PN 來進行求解 VRP、Wu 等人(2021)使用具有注意力機制(Attention Mechanism)的神經網路結構來求解有容量限制的車輛途程問題(Capacitated Vehicle Routing Problem, CVRP)、Luo 等人(2023)透過深度 Q 學習網路(Deep Q Network, DQN)來求解 CVRP 等。本論文將以此作為動機，根據過去學者所使用的方法來結合司機經驗，並求得新站點加入後之最佳運送路線。

## 1.2 研究目的

在過去文獻中大多都是使用啟發式演算法來降低運送成本，卻未考慮到實務上司機過往的送貨經驗，本研究透過 PN 來學習過去司機送貨經驗，以及實務上若有新客戶下單時，還可透過深度 Q 學習網路來結合過去經驗來進行更動。

本研究主要貢獻如下：

- (1) 本研究將司機過去經驗作為特徵，透過 PN 將具有歷史經驗站點重新排序，以滿足司機隱性偏好之最佳路線
- (2) 本研究透過深度強化學習因應實務上新客戶需求，將司機過往經驗結合新站點，求得最短路徑解

## 1.3 研究架構

本研究將透過 PN 結合深度強化學習來滿足司機隱性偏好，求解出考慮司機過去經驗之路線解。本研究流程如下：

第二章為文獻探討，將了解學者們之相關研究

第三章為研究方法，將在此章節提出本研究系統架構

第四章為研究成果，根據以標竿例題作為輸入資料，匯入本研究模型進行求解，並說明本研究貢獻

第五章為結論與未來研究，在此說明本研究模型考慮司機隱性偏好進行求解，在未來可往哪種方向發展



## 第二章 文獻探討

此章節說明與本研究相關之文獻，分為四部分：第 2.1 節介紹車輛途程問題之相關應用，第 2.2 節介紹運用指針網路求解組合最佳化問題之相關文獻，第 2.3 節介紹使用注意力機制增進準確率之相關文獻，第 2.4 節介紹深度強化學習運用於 VRP 問題之相關文獻。

### 2.1 車輛途程問題

車輛途程問題(Vehicle Routing Problem, VRP)是由基本旅行銷售員問題(Travelling salesman problem, TSP)所延伸出來，考慮到了車輛數量與容量的限制，須以最小化成本滿足客戶需求安排最適配送路線，因此也是一種 NP-Hard 問題。VRP 由 Dantzig 和 Ramser(1959)所提出，每位客戶有各自的需求，需要組織出最佳配送路線。也有將其延伸之各式 VRP 問題，像是有容量限制的車輛途程問題(Capacitated Vehicle Routing Problem, CVRP)、有時間窗限制的車輛途程問題(Vehicle Routing Problems with Time Windows, VRPTW)等相關議題。

除了傳統意義上的 VRP 問題，面對到實務上更多的司機會透過自身經驗來更動傳統最佳解路線，因此過去使用演算法所求解 VRP 最佳解無法滿足司機隱性偏好，導致管理上會沒有依據來規劃出統一路線，使得過去最佳解並不適用於實際情況。最近學者也漸漸開始探討基於歷史資料學習隱性偏好作為求解目標。

在過去的文獻當中已經有學者使用各種方法來進行求解，像是 Quirion-Blais 和 Chen(2021)基於案例推論(Case-based reasoning, CBR)的方法來儲存司機經驗，在根據調查與檢索的方式調整過去經驗路線來設計新路線履行訂單。該結果與 BoneRoute 算法相比，CBR 所求得路線成本遠高於 BoneRoute 求得成本 18.4%，但結果更趨近於先前司機行駛路線經驗相似度約 58.84%。Chen 等人(2021)設計出一種逆向優化的公式在開發了

一種基於乘法權重的更新的學習算法，透過學習司機經驗來推導更加合適的成本矩陣並以過去的決策經驗中學習成本矩陣。該研究比較了 185 個驗證實例的平均距離與平均相似度，結果表明學習後平均距離從 17.95 降低到 11.24、平均相似度落在 57.62%~71.84%。Zhao 等人(2023)建立了司機進行配送時客戶之間的時間依賴性的車輛路徑動態規劃模型(a Time-Dependent Vehicle Routing model considering the differences among paths, TDVRP-DP)，並開發了三種演化式算法(NSGA-II、NSGA-III 和 RVEA)，來求解配送易腐產品路線時，能夠最小化總成本並最小化客戶不滿。研究結果顯示，在不同規模的 VRP 當中，計算結果可節省大約 50%的時間，卻無法提高解決方案的正確性。Mandi 等人(2021)自行建構神經網路模型來估計點與點之間的轉移機率，並透過加權馬可夫計數法的差異，考量了司機行駛的隱性偏好，來求解實務上的 CVRP 問題。該實驗根據某公司 39 周資料以星期做區分，透過節線差異的計算公式後，其實驗結果與實際路線差異為 18.04。

經由上述學者研究結果，可得出以下結論：

- (1) 為了讓研究更加符合現實情境，越來越多學者著手研究考慮司機行駛偏好的 VRP 問題
- (2) 無論使用演算法或神經網路來進行研究，其結果都與實際情況皆有差異，甚至無法求得更好的結果
- (3) 在實務上也會出現新客戶的情況，而目前文獻未考慮到歷史經驗與新站點結合，因此本研究將兩者因素皆納入考慮，開發出新模型求解

## 2.2 指針網路

PN 是由 Vinyals 等人(2015)所提出，為了改善傳統序列到序列(Sequence to Sequence, Seq2Seq)的缺點，因此透過 PN 可透過指針指向輸入序列，解決 Seq2Seq 因長度過長而訊息丟失的情況。

目前 PN 也廣泛的運用到了求解 VRP 問題當中，例如 Ma 等人(2019)提出 Graph Pointer Networks(GPN)結合 Pointer Network 的架構來處理旅行商問題(Travelling salesman problem, TSP)，並運用強化學習來解決擁有時間窗的 TSP 問題。其研究結果與最佳解差距大約在 5.9%~12.8%，執行時間 200 秒，比 Concorde 求解器時間快 85%。Sheng 等人(2020)為解決具有有限資源且有站點選擇的優先順序 VRP，目標為透過安排不同優先順序的站點且構建車輛行駛路線取得最大化的總效益。車輛路徑集合相關的目標函數值由指針神經網路結合深度強化學習，可在訓練後短時間內獲得一個很好的可行解決方案。與差分學習法(Differential evolution, DE)、基因演算法(Genetic Algorithm, GA)進行比較，可看出該研究之模型的求解時間與效能皆遠勝於其餘演算法。Kong 等人(2022)為了求解無人機物流配送路線問題，開發了一種基於注意力機制的指針網路模型(attention-based pointer network model, A-Ptr-Net)，來解決無人機行駛路線優化問題。後續與混合整數線性規劃模型(mixed integer linear program, MILP)、模擬退火法(simulated annealing, SA)、掃描覆蓋法(sweep coverage, SC)、加權目標式掃描覆蓋法(weighted targets sweep coverage, WTSC)、行動邊緣運算法(mobile-edge computing, MEC)相比，無論無人機容量如何變化，A-Ptr-Net 的效能始終高於其餘模型，例如當無人機容量為 30 時，與最佳解差異僅只有 0.17%。

統整上述研究可發現，指針網路可有效求解 VRP 問題，而加入注意力機制的指針網路則可以更加準確地預測出符合實際狀況之最佳解，因此本研究選擇使用結合注意力機制之指針網路來學習司機行駛經驗路線，作為後續深度強化學習的環境。



## 2.3 注意力機制

注意力機制(Attention Mechanism)廣泛應用於機器翻譯、語音辨識等領域，最初是由Bahdanau等人(2014)提出在RNN Encoder-Decoder的架構中加入注意力機制。該篇研究方向為自然語言處理(Natural Language Processing, NLP)，在自然語言處理中，當句子過長時，原來重要的資訊會被後續資訊覆蓋，導致結果不好。而注意力機制則會找出句子中最富有資訊量的部分，並結合上下文向量(context vector)來進行預測，因此過去的資訊也會有部分保留下來，藉此提高預測準確度。Luong等人(2015)則更詳細的提出兩種注意力機制的結構，分別為全注意力機制(global attention)與區域注意力機制(local attention)。

研究證實，全注意力機制可將所有資訊納入考量卻很花費計算成本，區域注意力機制僅針對部分資訊提高運算效率，卻可能丟失某些重要資訊，之後研究可針對問題特性，選擇將使用全注意機制或區域注意力機制。Wu等人(2021)開發了一種基於自注意力的深度架構作為策略網絡，有效解決了 TSP 和 CVRP。Xin等人(2021)介紹了多解碼器注意方法(Multi-Decoder Attention Model, MDAM)模型，透過自注意力機制架構，結合強化學習求解TSP與CVRP問題，透過六種不同規模大小問題與傳統強化學習、Concorde等進行比較。由實驗結果可看出，該模型在六種情境下求解時間遠勝於其餘模型，求解結果也與實際結果差異 1~2%左右。Mo等人(2023)提出一種基於注意力機制的成對指針網路，使用司機的歷史行駛路線數據作為訓練資料，將城市點進行分段路線訓練。最後研究結果顯示，各四段預測路線與實際結果準確度大約落在 22%~24%左右。

以上學者使用了注意力機制加強模型，可提升模型預測準確度，也能夠成功求解VRP。但以上學者距離實際結果有一定的差異，並未能夠更加精準的預測出司機過去經驗，因此本研究將透過資料擴增、調整編碼器與解碼器等手法，來提高注意力機制結合指針網路之預測準確度。



## 2.4 深度強化學習

機器學習包括了不同類型的學習方式，主要透過資料的性質與期望的成果可以採用以下四種方式，分別為：監督式學習(supervised learning)、非監督式學習(unsupervised learning)、半監督式學習(Semi-supervised Learning)及強化學習(reinforcement learning)。監督式學習就是事先將訓練資料設定好標籤(label)，讓模型去學習特徵，後續能準確預測輸入資料。非監督式學習是將訓練資料丟入，未經過任何標記後，機器將使用所有相關且可存取的資料來進行識別類型與相關性。半監督式學習則是指輸入少量的標籤資料來強化未標籤化的資料，藉此來提升強化機器的學習力與準確性。最後一種強化學習則是無須給予任何訓練資料，單純透過與環境不斷地互動，來學習所得到的策略與規劃。

強化學習中主要有 5 個元素需要建構，分別為：環境(environment)、狀態(state)、動作(action)、代理人(agent)、獎勵(reward)，其架構如圖 1。強化學習需先自行架構出問題所需環境，接著環境中會回饋出當前狀態給與代理人，而代理人透過環境所回饋的獎勵值與選擇策略來決定下一次行動，行動後再度提供當前動作給環境。此時環境會根據上一次狀態轉移到當前動作來計算出所獲得的獎勵值，若獎勵值為正，則表示該動作執行為佳，建議可持續往該方向前進；反之，獎勵為負，則給予懲罰(penalty)，表示該動作執行會帶來不好的情況，因此給予懲罰值遠離該方向。有了獎勵值後，環境會將獎勵值輸入至演算法中，讓演算法更新至目前選擇決策並根據新的狀態來進行下一次動作選擇。

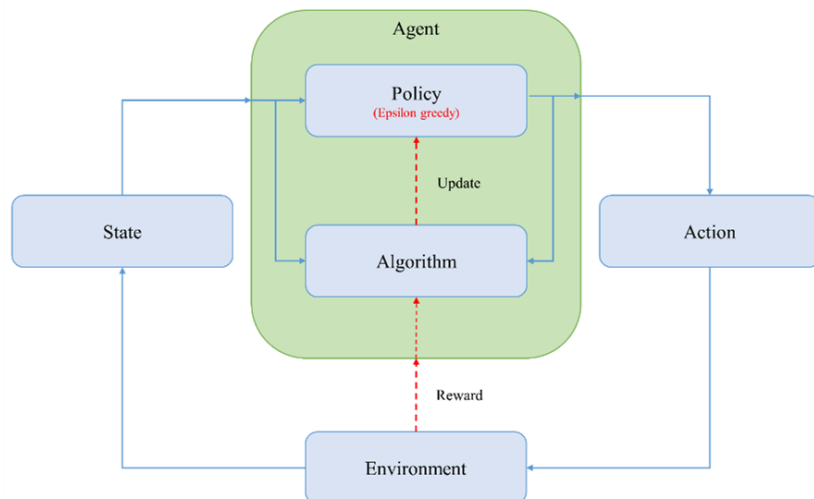


圖 1 強化學習架構圖

強化學習透過上述流程，經過了每一步的選擇策略，將該獎勵值所記錄下來，如同透過馬可夫決策過程(Markov decision process, MDP)建立模擬模型，如同公式(1)。將每一次迭代的過程與獎勵值紀錄，最終找出獎勵值最大的決策流程。

$$S_1, a_1, r_1, S_2, a_2, r_2, \dots, r_t \quad (1)$$

在很多現實的決策問題中，許多問題的狀態空間和行動空間非常大，例如在處理圖像、語音或自駕車等問題時，傳統的強化學習方法難以處理高維度的問題，因為需要大量的資源和時間進行搜索和學習。通常也需要進行大量的試錯，尤其是當動作空間很大或環境反饋的延遲性高時，這樣才可以找到最佳策略，導致學習效率低下。這對於傳統強化學習負擔很重，因傳統強化學習須利用窮舉法將所有動作列舉出來，但當問題規模擴大時，使用窮舉法會導致情況太多而造成維度爆炸的問題。

深度強化學習不僅僅可以解決維度爆炸問題，能夠處理高維度的狀態和行動空間。也能夠使用深度神經網路來自動提取特徵，從而有效處理複雜的狀態表示。這使得模型能夠更好地理解 and 表示輸入的結構和內在關係。深度強化學習模型能夠學習通用的策略和行為模式，並具有良好的泛化能力。這意味著在面對新的環境或類似的問題時，模型能夠快速適應和表現良好，而無需重新訓練。而 VRP 為 NP-hard 問題，因此深度強化學習可以更加有效的求解複雜度更高的 VRP 問題。

因此 Qiu 等人(2022)提出使用多端口注意力機制(multi-head attention mechanism)結合運用編碼器-解碼器(Encoder-Decoder)的深度強化學習來解決送貨安裝上門路線規劃的問題。作者們設計了多端口注意力機制能提供了更加廣泛的資訊提取能力，並使用 Beam Search 演算法作為解碼器在最佳解與計算效率時間上取得平衡。在後續研究上套用到一家公司的真實案例，實驗表明比起 LKH(Kernighan-Lin heuristic)演算法來說，對於規模大的問題計算時間還要快速。Luo 等人(2023)開發一種高效的編碼器-解碼器模型，稱為殘差圖卷積編碼器和基於多個注意力的解碼器(the Residual Graph Convolutional Encoder and Multiple Attention-based Decoders, RGCMA)所組成，為 CVRP 實例構建多個解決方案，以提高候選解決方案的質量。其結果顯示，RGCMA 在小規模例題中可完美獲得最佳解，而大規模問題的求解差異與時間則略遜強化學習多重優化策略(Policy Optimization with Multiple Optima for Reinforcement Learning, POMO)。Park 等人(2022)本文運用 DQN 來模擬車輛於仿真交通道路上行駛。本研究透過在路口設置路側設備(road side unit, RSU)及時採集車輛資訊回傳給霧計算(Fog Computing)，當霧服務器獲取路口檢測到的物體資訊透過專用短程通信技術(Dedicated Short Range Communications, DSRC)傳遞給範圍內的所有車輛。本研究比較了最早期限優先(Earliest Deadline First, EDF)、優先選擇請求最多(Most Requested First, MRF)、DQN，結果顯示當車輛密度高、服務請求多樣化時，需要一種能適應環境的算法，DQN 在不同的到達率環境上表現最佳。Chen 等人(2022)本文運用了深度 Q 學習方法求解使用無人機與車輛應用於當日送達(Same-day delivery, SDD)訂單。本研究透過深度 Q 學習來解決複雜度更多的維度爆炸問題，該動作空間分成了多個子集，其內容有無人機、車輛、服務並考慮車輛與無人機的分配順序。本研究將通過深度 Q 學習做出的決策，並使用啟發式方法來管理對特定車輛和無人機的分配。其結果模擬各種規模變化後，皆可以做出有效的決策並收斂。

統整以上研究，Raza 等人(2022)也提到解空間會隨著問題輸入規模的增加呈指數增長，因此尋找最優解需要非常高的計算量。目前大多數研究都沒有考慮到客戶環境與隨機現實因素來進行模擬設置。所以本研究將透過具有注意力機制之指針網路並結合深度

強化學習來求解帶有司機隱性偏好之車輛途程問題並考慮新站點增加的問題，藉由此模型提供給公司記錄資深員工經驗，將經驗量化後傳授給新進員工，減少訓練時間與提高運送準確度。



## 第三章 研究方法

本章節說明如何透過PNDQN求解車輛途程問題，根據過去歷史路線資料建立指針網路模型並說明如何將其路徑特徵進行學習訓練，接著考慮到新站點問題，說明如何結合深度強化學習來將新站點插入。第 3.1 節說明車輛途程問題之相關定義，第 3.2 節公式化相關的變數及限制式，第 3.3 節說明如何從過往資料中來學習物流司機的行駛隱性偏好，第 3.4 節介紹開發之PNDQN模型，第 3.5 節說明本研究實驗過程。

### 3.1 研究架構

目前物流公司最主要規劃配送路線皆以最短路線、最省成本為主，但這對於物流司機而言，會因為實務上所遇到情況而自行改變行駛路線。這就違反了原本規劃的配送路線，因為原始的規劃配送路線並未考慮到實務上所發生的狀況。為了能讓物流司機在最先開始就接收到考慮過實務需求之最佳化路線，因此本研究將研發出結合司機過去經驗與未曾運送過之新站點加入的模型，以開發出最適合物流司機行駛路線解。

本研究建構 PNDQN 模型，可透過 PN 來抓取過去物流司機送貨特徵，再使用深度強化學習來插入從未有歷史經驗的新站點，其架構圖如圖 2。一開始先將司機過去送貨路線透過資料擴增再進行獨熱編碼(One-Hot Encoding)，以提高城市點與點之間相關性。接著做為訓練資料丟入到 PN 網路中訓練，就可將城市點資訊丟入 PN 網路後，預測出過去行駛路線解。若有新站點需要納入考量時，則會將過去行駛路線做為深度強化學習的環境，透過經驗回放機制(Experience Reply)將現今狀態、現今動作、現今獎勵值、下一次狀態皆儲存。再來從記憶體隨機抽取固定數量的資料來丟入神經網路來進行訓練。透過 Epsilon greedy (簡稱  $\epsilon$ -greedy)策略來選擇下一次動作並將該動作傳遞給獎勵函數，則可以獲得下一次狀態與獎勵值。最後輸出 Q 值並選擇最高數值做為現今動作更新至環境中，重複到設定好的學習次數時便輸出最終路線解。

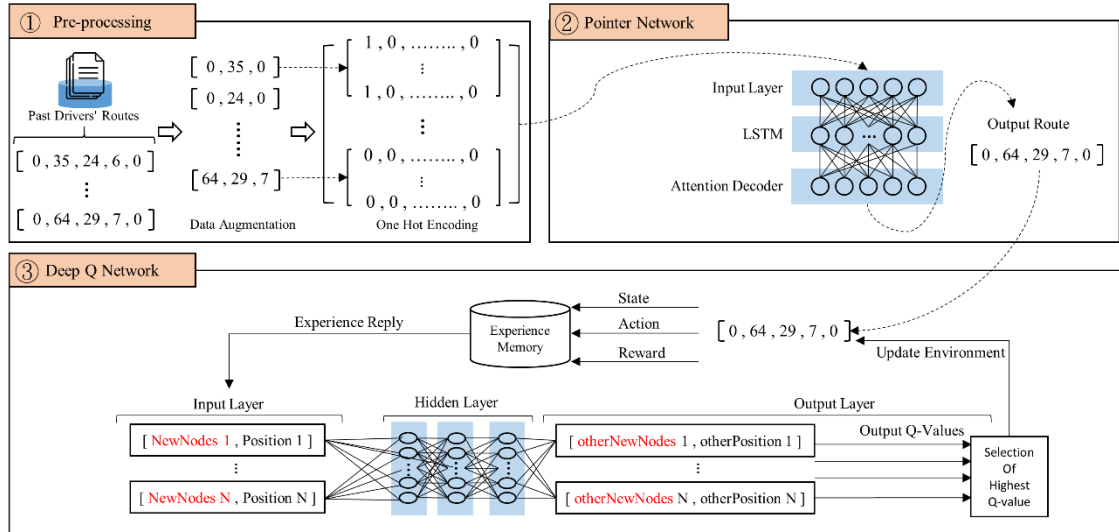


圖 2 研究架構圖

## 3.2 數學模型

本章節將介紹本研究建立求解目標行駛最小化路線的數學模型，求解了具有時窗限制及容量限制之車輛途程問題，本研究依循 Solomon(1987)所建構之硬時間窗之車輛途程問題。表 1 為本研究數學模型之符號定義，接著說明本研究數學模型完全基於 Solomon(1987)的論文設定，因後續實驗會與該論文所提出之國際標竿例題 (Benchmark) 為實驗案例。以下為本研究的相關假設：該情境僅只有一個場站(depot)，已知所有城市地點、時間窗、需求數量，已知車輛容量限制。除了基本設定以外，對於車輛途程問題也有以下假設：每位城市僅只會被一輛車服務，能服務之車輛數量，車輛剩餘容量需大於服務城市需求才可配送，每位城市有各自的服務時間，所有城市服務完成後會回到場站。

表 1 符號定義

$N = \{n   n = 0, 1, \dots, n\}$	城市點集合
$V = \{v   v = 1, 2, \dots, v\}$	車輛數集合
$D_{ij} \forall i, j \in N, i \neq j$	城市 <i>i</i> 到城市 <i>j</i> 之間距離
$X_{ijv} \forall i, j \in N, \forall v \in V, i \neq j$	$\begin{cases} 1 & , \text{當車輛} v \text{從城市} i \text{到城市} j \text{時} \\ 0 & , \text{其他} \end{cases}$
$y_{iv}$	為 0 或 1；若車輛 <i>v</i> 服務城市 <i>i</i> 時其值為 1，反之為 0
$q_i \forall i \in N$	城市 <i>i</i> 的需求量
$C$	每輛車輛最大負載容量限制
$t_{ij} \forall i, j \in N, i \neq j$	從城市 <i>i</i> 到城市 <i>j</i> 行駛時間
$t_{0i} \forall i \in N$	場站至城市 <i>i</i> 的旅行時間
$b_i \forall i \in N$	抵達城市 <i>i</i> 的時間
$s_i \forall i \in N$	城市 <i>i</i> 的服務時間
$l_i \forall i \in N$	城市點 <i>i</i> 最早能抵達時間窗限制
$u_i \forall i \in N$	城市點 <i>i</i> 最晚能抵達時間窗限制
$l_0$	最大可服務之時間窗下限
$u_0$	最大可服務之時間窗上限
$t_0^v \forall v \in V$	場站服務時間
$T$	為一大常數，表最大可服務時間



本研究數學模型建構如下：

### Objective Function

$$\text{Minimize } \sum_{i \in N} \sum_{j \in N} \sum_{v \in V} D_{ijv} X_{ijv}$$

### Constraints

$$\sum_{v \in V} y_{iv} = \begin{cases} V & i = 0 \\ 1 & i \in N \end{cases} \quad (2)$$

$$\sum_{i \in N} q_i y_{iv} \leq C_v \quad v \in V \quad (3)$$

$$\sum_{i \in N} X_{ijv} = y_{jv} \quad j \in N, v \in V \quad (4)$$

$$\sum_{j \in N} X_{ijv} = y_{iv} \quad i \in N, v \in V \quad (5)$$

$$b_j \geq b_i + s_i + t_{ij} - (1 - X_{ijv})T \quad (i, j) \in N, v \in V \quad (6)$$

$$b_i \leq u_i \quad i \in N \quad (7)$$

$$b_j = \text{Max}\{l_i, b_i + s_i + t_{ij}\} \quad (i, j) \in N \quad (8)$$

$$l_i \leq t_{0i} \quad i \in N, \quad (9)$$

$$l_0 \leq t_0^v \leq u_0 \quad v \in V, l_0 \geq 0 \quad (10)$$

第一目標函數為總車輛數最小，第二目標函數為總路線距離最小化。限制式(2)為確保每條路徑皆由場站出發，且路線最後一定會回到場站，每個城市點僅拜訪過一次。限制式(3)為車容量限制。限制式(4)與限制式(5)為限制每個城市點只會被進入與離開一次。限制式(6)為限制服務時間的順序，防止子路徑生成。限制式(7)與限制式(8)為硬時間窗限制。限制式(9)則定義了車輛的出發時間。限制式(10)為設定所有城市點最大最小時間窗限制。



### 3.3 以指針網路學習司機隱性偏好路線

傳統的序列到序列(Sequence to Sequence, Seq2Seq)是透過編碼器(Encoder)與解碼器(Decoder)兩個循環神經網路(Recurrent Neural Network, RNN)或是長短期記憶模型(Long Short-Term Memory, LSTM)所組成。編碼器會將輸入序列編碼轉換成上下文向量(context vector)，將當下輸入的文字以及上下文保留下來，再將其傳遞給解碼器生成輸出序列，並透過本次輸出文字當作下一次輸入，讓機器學習能夠抓取到上下文之間的關係特徵，使得輸出序列能更加符合預測結果。

Seq2Seq 是由學者 Sutskever 等人(2014)所提出，首先將輸入序列  $x = \{x_1, x_2, \dots, x_T\}$  輸入至 RNN 或是 LSTM 模型中，在每個時間點上輸出隱藏層狀態(hidden states)，計算方式如公式(11)。其中  $W$  為權重值，透過權重計算出輸入序列與前一步隱藏層之間的影響關係。最終由隱藏層狀態求解出輸出序列  $y = \{y_1, y_2, \dots, y_{T'}\}$ ，如公式(12)。但輸入序列長度  $T$  有可能會與輸出序列長度  $T'$  不同，因此在解碼器階段會計算最大條件機率來辨別輸出序列，如公式(13)。其中， $p(y_t|v, y_1, \dots, y_{t-1})$  是透過 *softmax* 函數求解出該機率，最後在每個序列後端加入結束符號(end-of-sentence symbol, EOS)作為序列結尾，幫助模型定義可變長度序列分布，如圖 3 所示。舉例來說，圖 3 中 City A、B、C 作為序列輸入資料，透過編碼器轉換成隱藏向量，在經由解碼器將向量轉成輸出序列並使用前一步的輸出(City B)作為下一次的輸入(City B)。

$$h_t = \text{sigm}(W^{hx}x_t + W^{hh}h_{t-1}) \quad (11)$$

$$y_t = W^{yh}h_t \quad (12)$$

$$p(y_1, y_2, \dots, y_{T'} | x_1, x_2, \dots, x_T) = \prod_{t=1}^{T'} p(y_t | v, y_1, \dots, y_{t-1}) \quad (13)$$

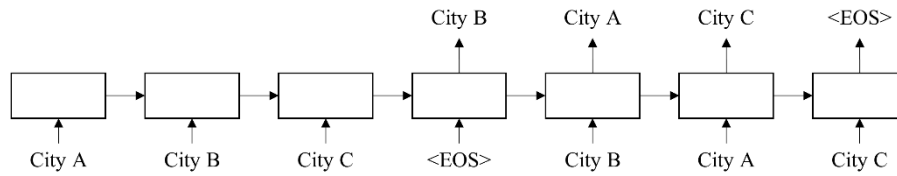


圖 3 傳統 Seq2Seq 模型

但在 Seq2Seq 模型中，編碼器將輸入序列字串壓縮成一個固定長度的上下文向量，會無法有效的表達出序列訊息，當字串長度過長時，會由於向量長度固定，讓已編碼完成的資訊被後來資訊覆蓋，丟失資訊，導致效果極差。而且 Seq2Seq 是將輸入序列的每個元素都賦予相同的權重，因此在每個時間序列時常常讓模型性能下降，導致準確率不高。

因此學者 Vinyals 等人(2015)提出透過「指針」結合注意力機制(Attention Mechanism)來改善 Seq2Seq 缺點。PN 的解碼器是根據輸入序列中的位置生成指標或選擇位置，因此 PN 可以將輸出序列中的某些元素映射到輸入序列中的特定位置。如圖 4， $P_0$  為出發站點， $P_1 \sim P_3$  作為輸入城市點序列，輸入各城市點座標  $(x_n, y_n)$  至 PN 中， $\Rightarrow$  與  $\Leftarrow$  即為 Seq2Seq 中的結束符號(end-of-sentence symbol, EOS)作為輸入與輸出序列結尾。透過隱藏層計算出輸出序列機率指針，透過機率指針指向輸入序列，進而生成機率最大之正確解答，如紅色箭頭即為當前時間步所預測出機率最大的輸出結果。由於輸出序列皆映射到輸入序列位置，各指針對應輸入序列每個元素，預測每一步時都會尋找當前輸入序列中最大權重的元素，因此輸出序列可以適應輸入序列的長度變化，無須設定輸出詞彙表。

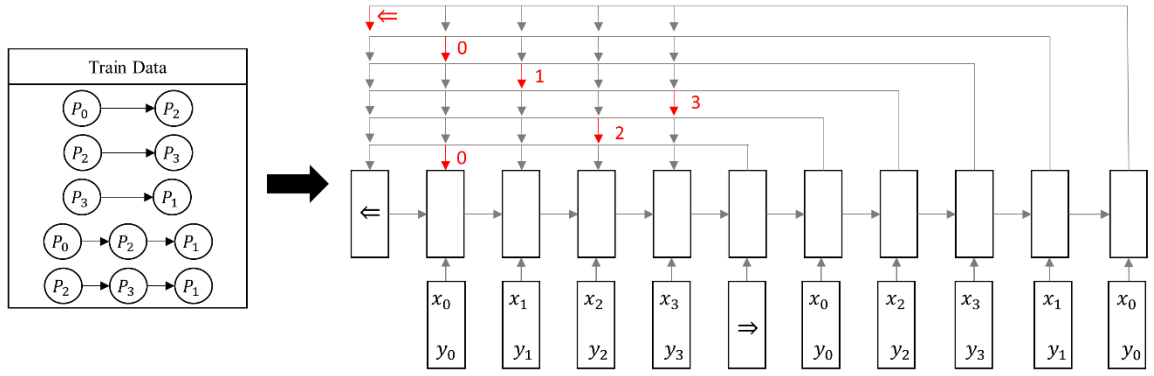


圖 4 使用 PN 求解車輛途程問題範例

注意力機制(Attention Mechanism)廣泛應用於機器翻譯、語音辨識等領域，最初是由Bahdanau等人(2014)提出在RNN Encoder-Decoder的架構中附加上注意力機制，藉此保留過去重要資訊，提高預測準確度。加入注意力機制後編碼器與解碼器計算方式如下：

$$u_j^i = v^T \tanh(W_1 e_j + W_2 d_i) \quad j \in (1, \dots, n) \quad (14)$$

$$a_j^i = \text{softmax}(u_j^i) \quad (15)$$

公式(14)中， $e_j$ 為編碼器的隱藏狀態會將額外的資訊傳播給解碼器， $d_i$ 為解碼器的隱藏狀態，模型會透過解碼器的隱藏狀態來進行預測， $v$ 、 $W_1$ 、 $W_2$ 為模型可學習之參數。透過此公式能獲得模型將輸入序列進行歸一化的注意力編碼向量，再經由公式(15)的 $\text{softmax}$ 函數取得機率最大之預測結果( $a_j^i$ )。而PN會透過該注意力機制生成指針，來指向輸入序列，求得機率最大的輸出結果，因此將公式(15)改寫為公式(16)。

$$p(C_i | C_1, \dots, C_{i-1}, P) = \text{softmax}(u^i) \quad (16)$$

PN專門針對輸出結果為離散型並且會對應輸入資料位置的問題，例如本研究為將各城市座標點作為輸入序列，透過PN結合注意力機制獲得在每一個時間步中，指針所指向的輸入序列相對應權重最大之元素組合。本研究將透過此種特性學習司機過去送貨路線之經驗，用PN預測出與過去送貨路線近似解。

本研究將引入PN串接至後續強化學習，透過PN來學習司機過去送貨經驗，以單一路線來進行規劃，藉以建構出強化學習所需環境。表 2 說明PN來學習司機過去經驗之訓練虛擬碼，最一開始初始化PN參數並設定好迭代次數(*epoch*)。在每一次訓練過程時，會根據一開始設定的批量大小(*B*)，將每條路經(*N*)中的顧客城市點集合 $I = \{i_1, i_2, \dots, i_n\}$ 作為輸入資料。透過指針網路結合注意力機制進行預測，生成出預測結果序列 $P = \{p_1, p_2, \dots, p_n\}$ 。接著與實際司機行駛路線序列 $O = \{o_1, o_2, \dots, o_n\}$ 進行交叉熵(Cross Entropy)計算預測序列與實際序列之間的誤差值，透過反向傳遞(Backpropagation)並經由本研究所使用之激勵函數Adam來更新網路權重，使得參數更新時會加穩定。

表 2 透過指針網路學習司機送貨經驗之虛擬碼

---

**Algorithm 1 : Past Driver Delivery Routes by Pointer Network**

---

```

1  Initialize Model(Pointer Network) parameters  $\theta$ 
2  For  $epoch = 1, 2, \dots, E$  do
3       $i_n \leftarrow Input()$   $n \in \{1, 2, \dots, B\}$ 
4       $o_n \leftarrow output(i_n)$ 
5       $p_n \leftarrow Model.Predict(i_n)$ 
6       $L \leftarrow Cross\ Entropy(p_n, o_n)$ 
7       $\theta \leftarrow Adam(L)$ 
8  end

```

---

本研究也考慮車輛容量、時間窗等多種限制加入模型內，使用自行開發之輸出層結構，將多種限制考慮後輸出不違反限制之預測結果。以容量限制為例，下一個城市點所需求容量加上目前載重容量，不得超出車輛容量限制。以時間窗為例，每個城市點皆有時窗上下界限，而車輛需要在時窗限制之內抵達城市，否則無法服務該城市點。因此本研究在輸出層設定若違反上述條件的城市點，則將預測結果機率調整為 0，這樣即可避免模型預測時會選擇到違反條件的城市。

為了讓模型在學習時能有更多訓練資料，本研究透過資料擴增的手法，將輸入資料進行切分擴展，讓模型在訓練時能夠根據更多樣本數來學習，調整正確之參數來進

行預測。方法如下：將司機過去送貨經驗路線進行切片取值，設定某數( $m$ )作為分割數，將一條路線進行兩兩一組( $m = 2$ )或三三一組( $m = 3$ )以此類推，並且城市前後順序不可更動。例如圖 5，原始司機行駛路線為 $[0, 74, 1, 25, 10, 0]$ ，經由分組可將路線拆分為 9 條路線，並且將拆分出的 9 條路線作為本研究的訓練資料。透過此種方法加強城市點之間的關連性，讓模型能學習到城市點彼此的相互前後關係。

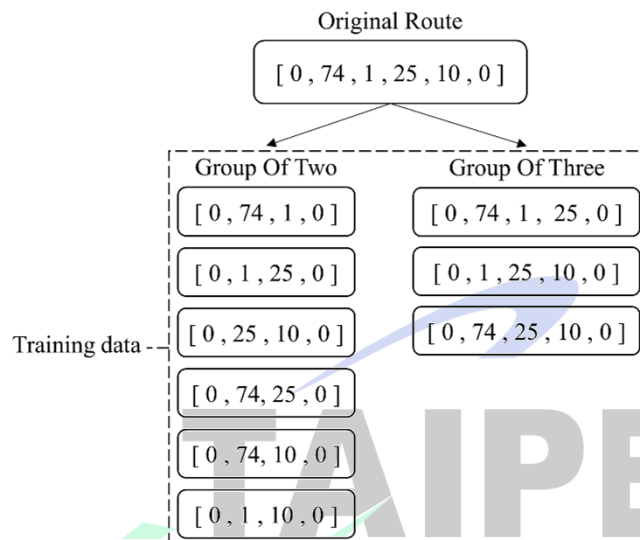


圖 5 資料擴增示意圖

### 3.4 深度強化學習結合指針網路之 PNDQN 模型

對於實務上來說，車輛途程問題往往會遇到該時段路線塞車、施工等情形需要司機根據經驗繞路，這方面可透過 PN 來進行學習。但若是有新站點加入時，司機並沒有過去送貨經驗，因此需要藉由強化學習來輔助考慮新站點的加入，規劃出當前最佳路線解。強化學習可透過選擇最大獎勵值來選擇新站點應該如何加入過去經驗路線，讓新的路線能結合過去經驗與新站點加入。因此本研究將應用強化學習的演算法來求解更加符合實務問題之車輛途程問題。

強化學習將會有代理人(Agent)與環境(Environment)交互作用，根據制定的狀態(State)與動作(Action)來獲得獎勵值(Reward)，藉由此種演算法來求得最佳路線結果。以本研究為例，代理人即為司機車輛、環境則為使用 PN 求解司機過去經驗路線、狀態

為目前新站點應插入過去經驗路線位置、動作為其餘新站點可插入過去經驗路線位置。

在Q-Learning中最大的問題在於若狀態與動作過多時，會發生Q表維度爆炸而無法紀錄。因此本研究會使用深度強化學習DQN，讓神經網路來儲存Q表內所需要的資料，像是當前狀態、當前動作、獎勵值、下一次狀態，將所需資訊透過神經網路來進行學習，可解決維度爆炸的問題。

DQN是一種無模型(model-free)的演算法。一開始並未給提供任何有關於該例題的預訓練模型，因此不需要先去了解該例題的運作模式，相較於基於模型的演算法(Model-based)，無模型演算法不需要事先建立一個確切的環境模型，可以更加容易地應用於現實世界的問題。無模型是透過反覆試驗(trial and error)讓代理人通過不斷地試錯，更新它對不同狀態和動作的估計值(Q值)，從而不斷調整其策略以達到最大化期望總收益的目標。透過此方法，代理人可以逐漸學習到在不同環境下哪些動作是有利的，並且可以在不斷探索和利用之間找到平衡點，以達到最優策略。甚至能將所選擇的組合儲存後，可考慮到更加長遠的價值來決定是否捨棄過久或當前的獎勵值組合，進而保留最佳獎勵值之策略。以下為DQN學習模型相關元素基本介紹：

- A. 代理人(Agent)：主體執行者。以本研究設定代理人為「物流司機」。
- B. 環境(Environment)：將問題建構出環境，讓代理人在裡面進行學習，透過不同的狀態(State)來改變環境現況。以本研究設定環境為「使用PN預測出具有過去司機送貨經驗之站點排序」。如圖 6 中的環境，過去經驗路線加上第一次狀態選擇後，當前環境為[ 0, 35, 74, 1, 25, 10, 0 ]。
- C. 狀態(State)：即為代理人目前所在位置，作為選擇下一次動作的依據。以本研究設定狀態為「目前新站點插入位置」，以陣列型態表示。如圖 6 中的狀態，當前狀態為新站點 35 插入至第一個空格位置，因此表示[ 35, 1 ]。
- D. 動作(Action)：讓代理人從當前狀態中選擇的一個可行的決定，這決定可讓代理人從當前狀態轉移至下一個狀態。以本研究設定動作為「其餘未選擇新站點插入



剩餘位置組合」，以陣列型態表示。如圖 6 中的動作，當前動作選擇列表有剩餘新站點 46 可插入到剩餘空格位置，因此表示[ 46 , 2 ]至[ 46 , 10 ]。

- E. 獎勵(Rewards)：當代理人從當前狀態執行一個動作後，將該動作回饋給環境即會獲得數值反饋。反饋數值即為獎勵值，可能有正、負或零的情況，這可代表代理人執行動作後表現是否優良，並藉此做為修改策略之基礎。以本研究設定獎勵為「插入新站點後之總行駛距離」。

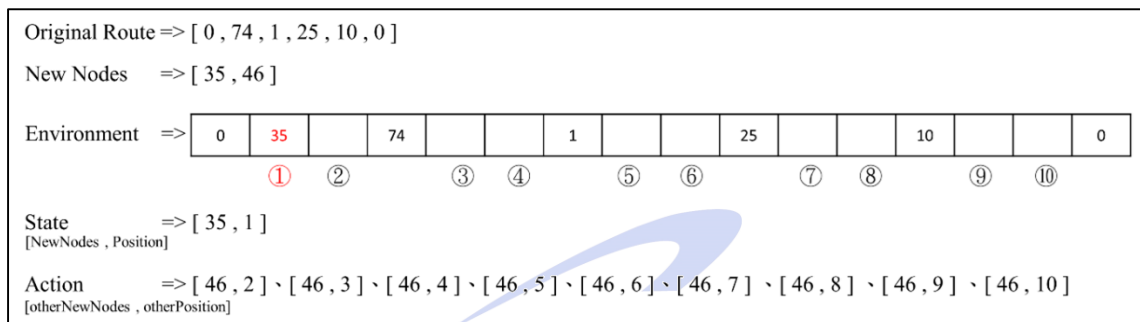


圖 6 本研究於 DQN 各項元素範例圖

本研究使用DQN作為求解未有經驗之新站點加入車輛途程問題演算法，透過DQN的方式在取樣完一整個迭代過程後，將該組合儲存並學習。而本研究將結合過去經驗路線解，開發具有司機過去行駛經驗並解決加入新站點之模型，DQN架構圖如圖 7 所示。從圖 7 可看出本研究環境為使用PN預測出的過去經驗路線解，最初狀態將設定為[ 0 , 0 ]表示起始狀態( $S_1$ )，接著會透過Q網路進行學習。本研究採取 $\epsilon - greedy$ 策略，讓演算法有 $\epsilon$ 的機率選擇隨機動作，另外會有 $1 - \epsilon$ 的機率選擇神經網路輸出的Q值向量中，Q值最高的動作( $\max Q(S_t, a_t)$ )。執行完動作( $a_t$ )後，即可從環境中獲得新狀態( $S_{t+1}$ )與獎勵值( $r_t$ )。為了避免災難性失憶的發生，因此加入了經驗回放的機制，將網路進行批次更新。會將所選擇到的狀態與動作儲存成記憶串列，像是[當前狀態, 當前動作, 當前獎勵值, 下一次狀態]匯入到設計的記憶體，當記憶體填滿時會將最新的資料取代最舊的資料。當數量資料數量超出本研究自訂批次時，會隨機選擇其中的資料組合成子集丟入目標Q網路內進行反向傳播，透過loss function計算Q網路輸出值

$(Q(S_t, a_t))$ 與目標Q網路輸出值 $(r_t + \gamma \max_{a_{t+1}} Q(S_{t+1}, a_{t+1}))$ 差距，進而更新目標Q網路參數。

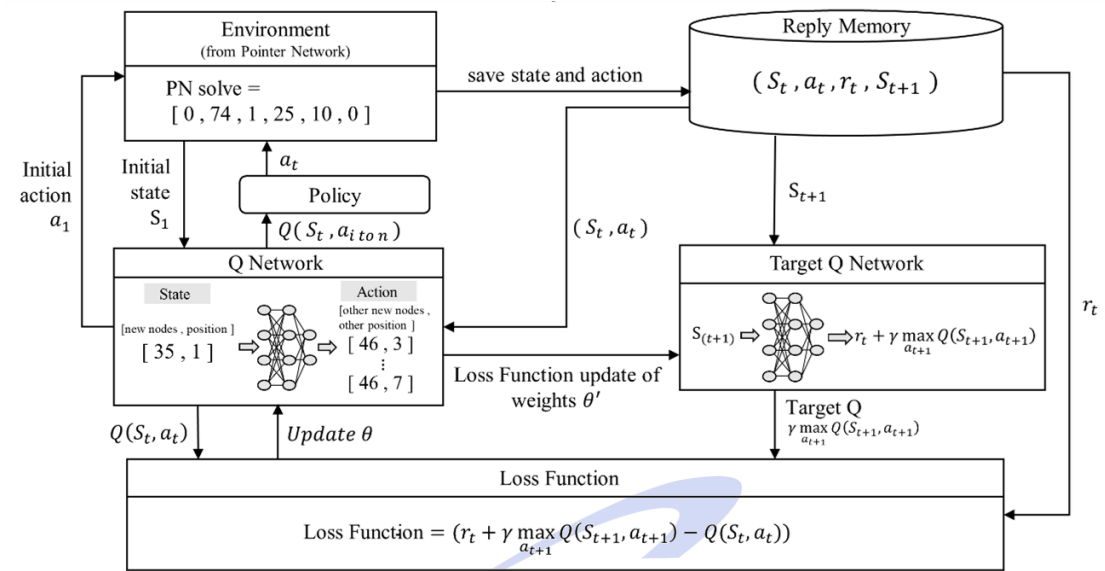


圖 7 本研究 PNDQN 架構圖

### 3.5 實驗過程

#### (一) 情境一：使用PN求解具有歷史資料站點路線

隨著時間的推移，許多老客戶會逐漸成為公司的歷史資料，留下了豐富的客戶交易紀錄，提供給公司規劃豐富的送貨路線經驗。本研究將使用這些路線作為訓練資料，提供給指針網路進行學習，讓指針網路學習到該路線特徵，生成後續預測路線權重依據。

如圖 8 所示，本研究一開始將所有城市點  $P = \{P_0, P_1, P_2, P_3\}$  的資訊作為輸入資料。再將司機過去行駛路線透過資料擴增的方式來生成出訓練資料，例如過去經驗路線為  $[P_0, P_2, P_3, P_1, P_0]$ ，透過資料擴增把路線更改成  $[P_0, P_2]$ 、 $[P_2, P_3]$ 、 $[P_3, P_1]$ 、 $[P_0, P_2, P_3]$ 、 $[P_2, P_3, P_1]$ 、 $[P_0, P_2, P_1]$ 、 $[P_0, P_3, P_1]$  等路線為訓練資料。接著透過PN學習到城市點相互前後關係特徵，藉此預測出過去經驗路線解。



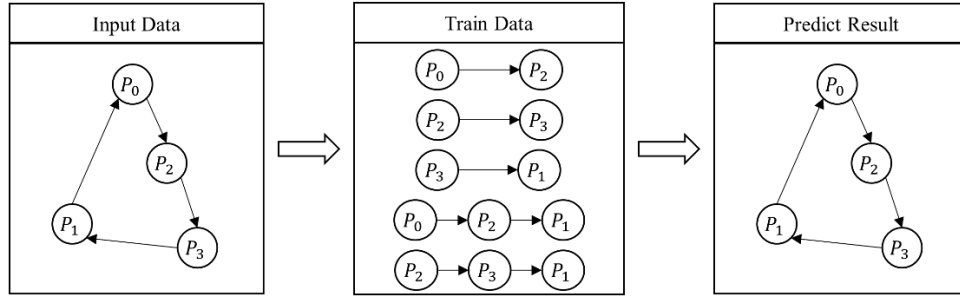


圖 8 情境一：使用 PN 求解具有歷史資料站點路線

## (二) 情境二：使用 PNDQN 模型求解具有歷史資料與從未行駛城市集合點路線

在現實生活中，當有新客戶提出運送要求時，公司內部並沒有過去送貨經驗，因此單純的 PN 已經無法滿足現實問題，所以本研究設計出 PNDQN 模型能夠處理該情境。如圖 9 所示，在環境的部分將繼承 PN 所預測出來的結果  $[0, 2, 3, 1, 0]$ ，在根據新站點集合  $[4]$  並設定選取動作之策略，本研究選取動作策略為「當新站點加入路線時，總路線長度最短」。接著將環境改寫成在各站點之間加入新站點數量的空格並將該空格位置進行編號，藉由該機制保留新站點加入路線的每一種可能性。初始狀態為  $[0, 0]$  表示目前尚未有新站點插入。而動作則有  $[4, 1]$ 、 $[4, 2]$ 、 $[4, 3]$ 、 $[4, 4]$  可選擇，第一個元素為新站點編號、第二元素為空格編號，因此  $[4, 1]$  意味著新站點 4 會插入到第 1 個空格位置中。獎勵則會計算新站點加入該空格後，一整條路線長度總長作為獎勵值。對於路線來說長度總長越短越好，因此以圖 9 為例，根據本研究制定的策略，本次動作會選擇  $[4, 1]$ ，最短總長度為 20 單位。經由反覆迭代到指定次數之後，則會規劃出司機最佳行駛路線，因此最終結果為  $[0, 4, 2, 3, 1, 0]$ 。

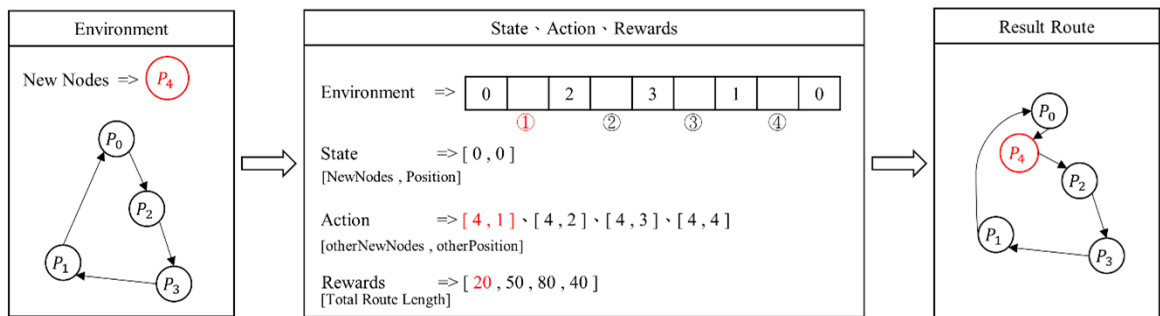


圖 9 情境二：使用 PNDQN 模型求解具有歷史資料與從未行駛城市集合點路線

本研究透過PN特性來訓練過去司機行駛經驗，在考慮到實務上的新站點加入問題，透過DQN來學習並計算出獎勵值，藉由本研究開發模型規劃出司機既可遵循過去經驗解，又可兼顧新站點指派到最適位置路線結果。本研究將於下一章節說明案例實驗結果。



## 第四章 實驗結果

本章節提出之PNDQN模型可否求解具隱性偏好之車輛途程問題，使用Solomon(1987)所設計之 56 題具時窗限制之車輛途程問題(Vehicle Routing Problem with Time Window, VRPTW)國際標準測試例題及Mandi等人(2021)所提供之範例資料進行分析，並分成 3 小節加以說明。第 4.1 節透過PNDQN展示情境 1 之實驗結果，並使用資料擴增來加強學習特徵；第 4.2 節模擬有新站點資料的情況，用以測試情境 2 之實驗結果；第 4.3 節介紹案例之相關數據及說明沿用該案例之評估指標，透過該指標來證實PNDQN模型之成效。設備方面，本研究使用之CPU為Intel(R) Core(TM) i7-9750，記憶體為 8G，系統為 64 位元之作業系統電腦。

### 4.1 情境一：求解具有歷史經驗城市點實驗結果

為了保證PNDQN模型實驗結果準確性，本研究將使用Solomon(1987)所設計之 56 題VRPTW的國際標準測試例題來做為測試資料，驗證模型求解的準確度、速度與穩定性。該國際標準測試例題提供了一個用於評估算法效能的基礎資料集，其中根據城市分布情形分為三類問題實例：

- (1) C類：指城市點呈現聚集在一起的狀態，也稱為群聚式分布，通常出現在城市或城市周邊地區。在這種情況下，城市點之間的距離比較近，交通路線較為密集，需要考慮到城市交通限制等因素。
- (2) R類：指城市點呈現完全隨機分布的狀態，也稱為完全隨機式分布，通常出現在人為開發較少的地區，例如郊區或農村地區。在這種情況下，城市點之間的距離比較遠，交通路線較為疏鬆。

(3) RC類：指城市點呈現混合完全隨機式分布和群聚式分布的狀態。這種情況下，城市點之間的距離既有比較近的，也有比較遠的，交通路線密度和疏鬆程度均有所不同。

該題庫提供了每個城市點的X座標、Y座標、城市點需求、最早可進入時間、最晚可進入時間、服務時間的資訊，每個題庫皆擁有 100 城市點。接著在每種類型中都再分為 1、2 類，1 類時間窗區間與車容量較小，2 類時間窗區間與車容量較大，因此所有題庫以下簡稱：C1、C2、R1、R2、RC1、RC2。

在神經網路中，超參數是指在模型訓練之前需要手動設置的一些模型參數，例如學習率、批次大小等。超參數的選擇會直接影響神經網路的訓練速度、收斂性和泛化能力。如果超參數設置得不當，可能會導致模型無法訓練或者產生過度擬合等問題。例如：學習率設置過大，可能會導致模型無法收斂。因此，超參數的選擇和優化是神經網路中非常重要的一個設定。超參數對模型的影響是固定的，且在模型訓練過程中不會發生改變，因此超參數的選擇對於神經網路的性能和準確性也具有重要影響。因此，如何調整並選擇出最佳參數組合是非常重要的課題。

如同上述所說，超參數組合可透過手動或自動的方式來嘗試哪種組合為最佳解。但使用手動調整需花費大量的時間且有可能遺漏的情況發生，因此本研究將使用自動化調整超參數的方法來協助找到最佳組合。本研究在參數組合選擇上使用了貝葉斯參數優化(Bayesian Optimization)的方式，找出該神經網路最佳超參數組合。首先，先將所需設定之超參數設定範圍，接著使用貝葉斯參數優化將這些超參數在該範圍內，找出對於模型來說結果準確度最佳解。本研究測試超參數如下：神經網路層層數搜索範圍[1,2,3]，各層神經網路層的神經元個數搜索範圍[5,512]、學習率搜索範圍[1e-6,1e-1]、激活函數[relu, sigmoid, softmax]、學習優化器[adam, SGD, Momentum]，每次訓練批次大小搜索範圍[2,64]。

最後，本研究透過貝葉斯參數優化找出的最佳超參數組合如下：使用LSTM層編碼與解碼層各一層、各層神經元個數為 158、學習率為1e-4、激活函數使用了softmax、

學習優化器則使用 $adam$ 、訓練批次大小為 16。本研究透過題庫C1、C2、R1、R2、RC1、RC2 這六種國際標竿例題來檢測本模型預測準確度與穩定性。基於表 3 和表 4，類型 1、2 題庫的分析，可以發現PNDQN模型在訓練過程中展現出穩定的效果和高度的準確率，在最終準確率PNDQN模型皆趨近於 100%，可證實PNDQN模型為有效模型。

表 3 訓練 1 類題庫模型評估表

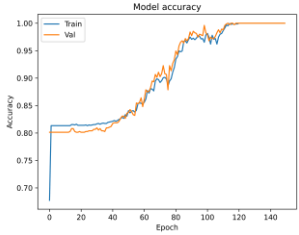
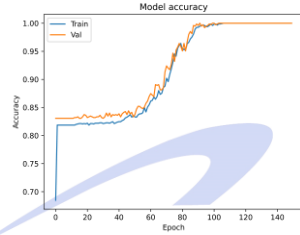
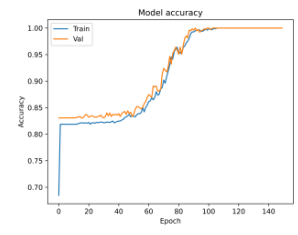
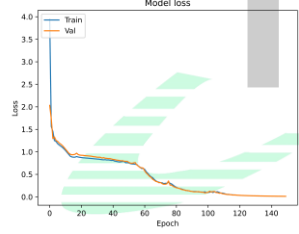
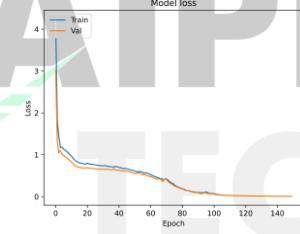
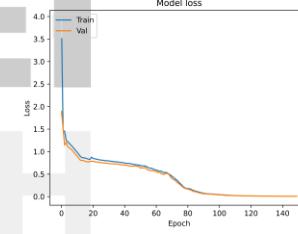
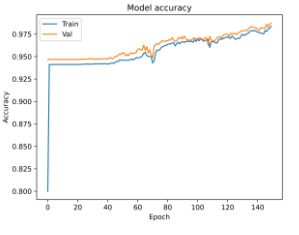
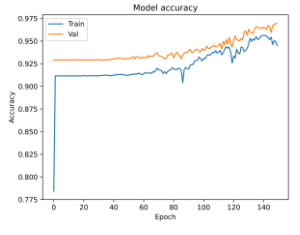
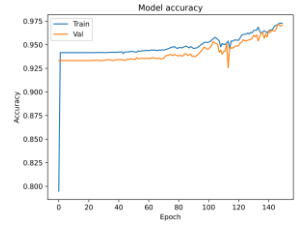
	C1	R1	RC1
準確函數 Accuracy			
損失函數 Loss			

表 4 訓練 2 類題庫模型評估表

	C2	R2	RC2
Accuracy			

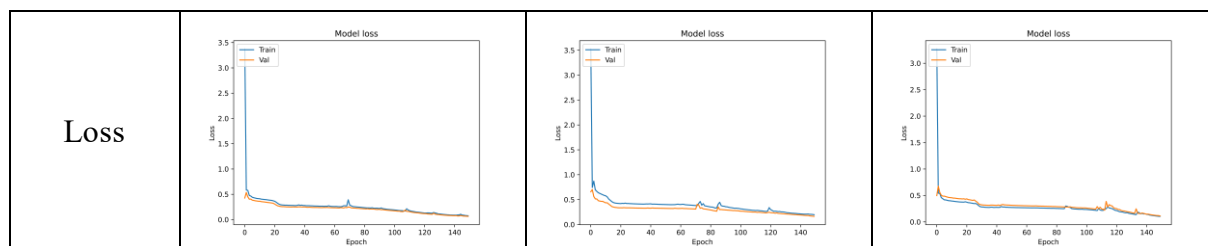


表 5 為Solomon(1987) 56 題國際標竿例題文獻已知最佳解。本研究將使用該表所提供之最佳路線解作為司機過去行駛路線，並在後續中比較預測結果是否完全符合該路線，以證實PNDQN模型能夠透過過去經驗來預測司機行駛路線。

表 5 Sololmon 56 題國際標竿例題文獻已知最佳解

Problem	Vehicle	Distance	Best Solutions by Heuristics	Problem	Vehicle	Distance	Best Solutions by Heuristics
C101	10	828.94	Rochat和 Taillard(1995)	C201	3	591.56	Rochat和 Taillard(1995)
C102	10	828.94	Rochat和 Taillard(1995)	C202	3	591.56	Rochat和 Taillard(1995)
C103	10	828.06	Rochat和 Taillard(1995)	C203	3	591.17	Rochat和 Taillard(1995)
C104	10	824.78	Rochat和 Taillard(1995)	C204	3	590.6	Rochat和 Taillard(1995)
C105	10	828.94	Rochat和 Taillard(1995)	C205	3	588.88	Rochat和 Taillard(1995)
C106	10	828.94	Rochat和 Taillard(1995)	C206	3	588.49	Rochat和 Taillard(1995)
C107	10	828.94	Rochat和 Taillard(1995)	C207	3	588.29	Rochat和 Taillard(1995)
C108	10	828.94	Rochat和 Taillard(1995)	C208	3	588.32	Rochat和 Taillard(1995)
C109	10	828.94	Rochat和 Taillard(1995)				
R101	19	1645.79	Homberger(2000)	R201	4	1252.37	Homberger和 Gehring(1999)
R102	17	1486.12	Rochat和 Taillard(1995)	R202	3	1191.70	Rousseau 等人(2002)
R103	13	1292.68	Li和 Lim(2003)	R203	3	939.54	Mester(2002)
R104	9	1007.24	Mester(2002)	R204	2	825.52	Bent和 Van Hentenryck(2004)
R105	14	1377.11	Rochat和 Taillard(1995)	R205	3	994.42	Rousseau 等人(2002)
R106	12	1251.98	Mester(2002)	R206	3	906.14	Schrimpf 等人(2000)
R107	10	1104.66	Shaw(1997)	R207	2	893.33	Bent和 Van Hentenryck(2004)
R108	9	960.88	Berger和 Barkaoui(2004)	R208	2	726.75	Mester(2002)
R109	11	1194.73	Homberger和 Gehring(1999)	R209	3	909.16	Homberger(2000)
R110	10	1118.59	Mester(2002)	R210	3	939.34	Mester(2002)
R111	10	1096.72	Rousseau 等人(2002)	R211	2	892.71	Bent和 Van Hentenryck(2004)
R112	9	982.14	Gambardella 等人(1999)				
RC101	14	1696.94	Taillard 等人(1997)	RC201	4	1406.91	Mester(2002)
RC102	12	1554.75	Taillard 等人(1997)	RC202	3	1367.09	Czech和 Czarnas(2002)

RC103	11	1261.67	Shaw(1998)	RC203	3	1049.62	Czech和 Czarnas(2002)
RC104	10	1135.48	Cordeau 等人(2001)	RC204	3	798.41	Mester(2002)
RC105	13	1629.44	Berger和 Barkaoui(2004)	RC205	4	1297.19	Mester(2002)
RC106	11	1424.73	Berger和 Barkaoui(2004)	RC206	3	1146.32	Homberger(2000)
RC107	11	1230.48	Shaw(1997)	RC207	3	1061.14	Bent和 Van Hentenryck(2004)
RC108	10	1139.82	Taillard 等人(1997)	RC208	3	828.14	Ibaraki 等人(2001)

表 6 為PNDQN模型在預測Solomon(1987)C1、C2、R1、R2、RC1、RC2 之 56 題國際標竿例題時，所預測出的路線距離、執行時間、已知最佳解等比較結果。透過表 6 可看出，僅只有在C208、R207、R208、RC208 這四種題庫中，PNDQN模型所找出的解答與目前最佳解有落差，但其餘題庫皆可 100%準確預測。在執行時間上皆小於 1 秒即可求得解答，證明PNDQN模型可用極快的速度且高的準確度來求解有過去行駛經驗之車輛途程問題。





表 6 PNDQN 模型對於 Solomon(1987)56 題標竿例題預測結果

Test	PNDQN_Distance	Best_Distance	Run Time(s)	Test	PNDQN_Distance	Best_Distance	Run Time(s)
C101	828.9366	828.9366	0.8222	R112	982.1392	982.1392	0.6509
C102	828.9366	828.9366	0.6581	R201	1252.3707	1252.3707	0.6913
C103	828.0646	828.0646	0.6699	R202	1191.7025	1191.7025	0.7030
C104	824.7765	824.7765	0.6493	R203	939.5029	939.5029	0.7572
C105	828.9366	828.9366	0.6665	R204	825.5188	825.5188	0.7341
C106	828.9366	828.9366	0.6608	R205	994.4272	994.4272	0.7567
C107	828.9366	828.9366	0.6631	R206	906.1416	906.1416	0.7562
C108	828.9366	828.9366	0.6560	<b>R207</b>	<b>973.4007</b>	<b>890.6078</b>	<b>0.7407</b>
C109	828.9366	828.9366	0.6604	<b>R208</b>	<b>952.5636</b>	<b>726.8223</b>	<b>0.7436</b>
C201	591.5563	591.5563	0.7437	R209	909.1629	909.1629	0.7483
C202	591.5563	591.5563	0.7407	R210	939.3722	939.3722	0.7433
C203	591.1732	591.1732	0.7413	R211	885.7109	885.7109	0.7347
C204	590.5985	590.5985	0.7459	RC101	1696.9487	1696.9487	0.6643
C205	588.8757	588.8757	0.7460	RC102	1554.7468	1554.7468	0.6604
C206	588.4926	588.4926	0.7437	RC103	1261.6714	1261.6714	0.6628
C207	588.2860	588.2860	0.7412	RC104	1135.4790	1135.4790	0.6629
<b>C208</b>	<b>654.0137</b>	<b>588.3235</b>	<b>0.7456</b>	RC105	1629.4359	1629.4359	0.6715
R101	1650.7986	1650.7986	0.6896	RC106	1424.7333	1424.7333	0.6723
R102	1486.8586	1486.8586	0.6711	RC107	1230.4771	1230.4771	0.6664
R103	1292.6749	1292.6749	0.6623	RC108	1139.8211	1139.8211	0.6473
R104	1007.3101	1007.3101	0.6557	RC201	1406.9396	1406.9396	0.7543
R105	1377.1105	1377.1105	0.6722	RC202	1365.6446	1365.6446	0.7486
R106	1252.0301	1252.0301	0.6604	RC203	1049.6238	1049.6238	0.7555
R107	1104.6555	1104.6555	0.6594	RC204	798.4632	798.4632	0.7424
R108	960.8753	960.8753	0.6486	RC205	1297.6474	1297.6474	0.7674
R109	1194.7336	1194.7336	0.6568	RC206	1146.3169	1146.3169	0.7416
R110	1118.8381	1118.8381	0.6664	RC207	1061.1442	1061.1442	0.7374
R111	1096.7261	1096.7261	0.6525	<b>RC208</b>	<b>844.7179</b>	<b>828.1411</b>	<b>0.7307</b>
Average Run Time(s)							0.7035

為了確認PNDQN模型在求解車輛途程問題是否有相當的準確性，本研究亦與蟻群演算法 (Ant Colony Optimization, ACO) 和粒子群演算法 (Particle Swarm Optimization, PSO)來進行比較。ACO為根據Othman等人(2018)設定之超參數來進行測



試，超參數組合如下：迭代次數 $E=150$ ，螞蟻數 $s=20$ ，費洛蒙之重要性 $\alpha=1$ ，距離之重要性 $\beta=1$ ，費洛蒙揮發係數 $\rho=0.05$ 。PSO為根據Attard(2021)所設定之超參數來進行測試，超參數組合如下：迭代次數 $G=1000$ ，總粒子數 $n=1500$ ，慣性權重 $w=10.9$ ，加速常數為 $c_1=1$ 、 $c_2=1$ ，速度更新之亂數為 $rnd=[-1,1]$ 。本研究將以C101、C201、R101、R201、RC101、RC201 這六個題型中，在已知最佳解裡最長的路線來進行檢測。本次檢測方式為將最長路線之城市點作為輸入資料，設定每台車容量、硬時間窗等資訊後，在丟入到各模型當中分別執行 10 次，記錄路線結果、行駛時間。透過表 7 可看出，在這六種題型中分別將最長路徑的城市點輸入後，所有演算法在預測同一條路線解時，執行結果與最佳解一致。再根據每跑 10 次平均時間來進行比較，可知道以PNDQN模型所執行時間遠勝於ACO、PSO。因此使用PNDQN模型可預測出司機過去行駛過路線解，具有優異的求解品質，而求解時間也遠勝於其於演算法。

表 7 ACO、PSO、PNDQN 執行時間比較表

Test	Nodes of Longest Path	ACO Average Time (s)	PSO Average Time (s)	PNDQN Average Time (s)
C101	13	1.2182	1.5135	<b>0.6715</b>
C201	35	4.6332	5.2263	<b>0.7439</b>
R101	8	0.7106	0.7946	<b>0.6541</b>
R201	30	3.5213	3.8962	<b>0.6941</b>
RC101	10	0.9306	1.1007	<b>0.6651</b>
RC201	27	3.5032	3.7991	<b>0.7108</b>

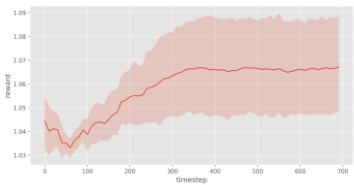
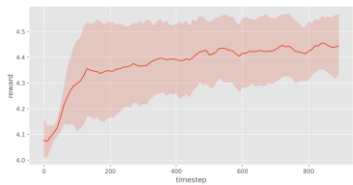
根據上述實驗可得知，PNDQN模型在對於司機過去行駛路線的預測上，皆有不錯的求解品質及速度。因此本研究將該預測結果匯入後續深度強化學習之環境依據，以便求解新站點加入之車輛途程問題。

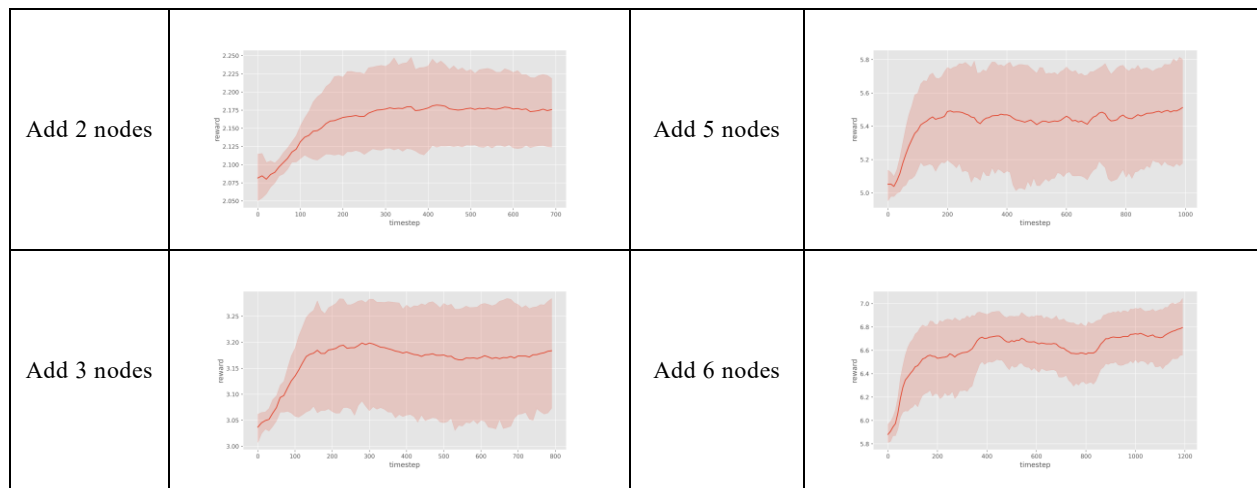
## 4.2 情境二：求解歷史經驗城市點與新站點實驗結果

本研究使用Solomon(1987)56 題國際標竿例題來模擬新站點加入司機過去經驗路線解情形，基於以求解出過去司機行駛路線解，在透過深度強化學習來取得新站點插入位置的最佳策略。由於C2、R2、RC2 類型之題庫皆屬於車容量大、路線較長、總路線較少，難以從中挑選出其餘站點作為新站點插入到該路線上，因此本研究將針對標竿例題的R101 題庫中，挑選出一條路線做為測試例題，在隨機選取 1~6 個站點作為新站點資料。最後透過PNDQN模型來將新站點插入到最佳位置，在盡量符合時間窗與車輛容量限制下，求得距離最小化路線解。

在深度強化學習中，評估模型是否已經收斂是一個重要的問題。通常會透過觀察獎勵值(Reward)是否已經收斂來評估模型的性能。獎勵值反映了代理人在學習過程中的表現，因此獎勵值的收斂圖是評估模型收斂性的一個關鍵指標。本研究使用PNDQN模型測試R101 題組之模型評估如表 8 所示。本研究針對不同情境設置不同的迭代次數，並記錄每跑 10 次的獎勵值變化情況。為了更直觀地呈現資料，本研究採用平滑化處理的方法進行展示。在表 8 中，本研究使用了淺色部分來表示在該時間段內執行 10 次之獎勵值的最大和最小區間，同時使用實線來表示在該時間段內執行 10 次之獎勵值的平均值。

表 8 PNDQN 模型測試 R101 之獎勵值收斂圖

R101			
Condition	The reward metric plot	Condition	The reward metric plot
Add 1 node		Add 4 nodes	



根據上述實驗結果，PNDQN模型求解上皆有不錯的收斂結果。因此本研究將以PNDQN模型、ACO、PSO來進行新站點加入結果比較。例如表 9 為在題組R101 中的最長路線加入 1~6 個新站點，透過不同的模型與演算法，各跑 10 次取得距離平均數。從表 9 可看出僅只有在加入 4 個新站點的例題中，PNDQN模型求解出的平均距離略輸ACO、PSO。其他例題皆遠勝於ACO、PSO，因此證明PNDQN模型求解品質優於其餘演算法。

表 9 PNDQN、ACO、PSO 加入新站點比較表

Add Node	ACO、PSO	PNDQN
	Average Distance	Average Distance
1	183.3800	<b>153.9789</b>
2	194.1658	<b>182.8209</b>
3	229.7542	<b>217.7462</b>
4	<b>253.3142</b>	261.0389
5	297.4977	<b>286.0351</b>
6	354.0608	<b>346.7065</b>

### 4.3 案例說明與實驗結果

本研究根據Mandi等人(2021)之論文於 39 周內收集的 201 條每日路線做為資料集。總城市點為 73 個，且已知城市點之間行駛距離。本研究以此作為訓練資料，透過(1)無資料擴增(2)有資料擴增的兩種方式來丟入PNDQN模型內進行比較，在訓練與測試資料上皆分別以訓練資料的 75%與 25%進行拆分。

本研究沿用該篇論文中的評估指標作為比較基準，評估指標為節線差異(Arc Difference, AD)。如公式(17)所示，分母為真實路線的總節線數，分子為測試路線與預測路線中的差異節線數量，以百分比顯示。透過該評估指標可表示百分比越小，與真實解越接近。

$$AD(\%) = \frac{\text{the set difference of the arc sets of the test and predicted solutions}}{\text{the total number of arcs in the actual routing}} \times 100\% \quad (17)$$

本研究將分別以(1)無資料擴增(2)有資料擴增與Mandi等人(2021)自行開發之模型來進行比較。Mandi等人(2021)資料是將每天跑的路線以星期做區分後，再進行預測。所以表 10 為原論文預測結果、無資料擴增使用PNDQN模型進行預測、有資料擴增使用PNDQN模型進行預測之比較結果表。

表 10 與現有論文結果比較表

AD(%)	Mandi等人(2021)	PNDQN	
	(Neural networks)	(Without Data Augmentation)	(With Data Augmentation)
Monday	<b>23.62</b>	29.96	24.23
Tuesday	<b>25.82</b>	27.43	29.54
Wednesday	<b>20.48</b>	21.43	22.45
Thursday	<b>21.17</b>	28.18	25.91
Friday	18.08	17.77	<b>11.68</b>
Saturday	<b>0.00</b>	<b>0.00</b>	<b>0.00</b>
Sunday	17.11	15.21	<b>11.06</b>
Overall	18.04	20.00	<b>17.84</b>

由表 10 可得知PNDQN模型在星期一到四的預測結果中略輸Mandi等人(2021)自行開發之模型，但在星期五至日PNDQN模型在有資料擴增的情況下遠勝於其餘模型，因此以平均整體結果來說，PNDQN模型在有進行資料擴增的情況之下，與實際路線誤差為 17.84%，勝於Mandi等人(2021)自行開發模型。

本研究主要目標是透過PNDQN模型預測出司機過去的行駛路線，並加入新站點後求得最短距離的路線解。本研究的應用範圍為物流領域，旨在協助公司更有效地管理供應鏈，提高物流效率。透過對司機過去的行駛經驗進行分析，本研究可以量化老員工物流司機的經驗，並將其傳承給新員工，以提高新員工的經驗水平。總的來說，本研究提供了一個有效的解決方案，幫助公司在管理供應鏈和物流方面能夠更加精確、高效、可持續，同時也幫助公司實現員工經驗的傳承和知識的累積，提高公司整體的競爭力。



## 第五章 結論

### 5.1 研究結果與管理內涵

本研究主要透過PNDQN模型來學習司機行駛經驗並考慮新站點加入問題，以指針網路與深度強化學習的特性來求解實務問題。對於實務上物流司機經常行駛非最佳解路線，因為路段封閉或現實其餘因素等，導致物流司機會依照自身經驗來規劃出當前最優解。PNDQN模型證實能夠有效將過去司機行駛路線記錄並儲存至神經網路中進行學習訓練，因此後續僅需要輸入城市點資料即可輸出過去行駛路線，準確度近乎為 100%。PNDQN模型亦可證實當新站點加入後，須考慮到過去行駛經驗與新站點之間的關係，可求解出符合兩種條件之行駛路線。其求解出來的路線結果在六種情況中與其餘演算法(ACO、PSO)相比皆節省 19.6%路線長度，運算時間則節省了 82.26%。在物流領域中，員工經驗的傳承是個重要的問題，而本研究提供了一個有效的解決方案。PNDQN模型基於深度學習技術將老員工的行駛路線進行了量化處理，能夠實現經驗傳承，讓新進員工能夠快速地上手。其次，能夠提高物流效率，減少運輸時間和成本，提高客戶滿意度。此外，透過對路線特徵的分析和提取，公司可以針對員工的強項進行專業的培訓，提高員工的技能水平和專業素質。

## 5.2 研究限制與未來發展

本研究雖考慮了過去經驗與新站點的結合，但也有以下三點沒有考慮到，未來可以加入更加完善該實驗：

- (1) 即時性時間窗：目前本研究所設定之時間窗為固定時間窗，因此未考量到實務上動態時間窗的限制。對於未來實驗可加入動態時間窗的考量，更加貼近實務情形。
- (2) 行駛新站點後成為過去經驗：本研究是透過行駛最短距離來插入新站點，並未考慮司機偏好路線。有可能司機會根據自身經驗將該站點重新調整至適當排序位置，因此建議可在深度強化學習中在加入司機判斷經驗作為特徵來插入新站點。
- (3) 加入即時路況：本研究並未考慮到即時路況，建議未來實驗可搭配即時路況、天氣等因素來進行規劃。可以確保往年司機行駛路線已不符合現有路況的發生，並讓新進司機更加能夠依照規劃出的路線行駛，節省大量的即時重新規劃路線之時間。



## 參考文獻

- [1] Attard, A. (2021). Particle swarm optimization for the vehicle routing problem with time windows. University of Malta,
- [2] Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.
- [3] Bent, R., & Van Hentenryck, P. (2004). A two-stage hybrid local search for the vehicle routing problem with time windows. *Transportation Science*, 38(4), 515-530.
- [4] Berger, J., & Barkaoui, M. (2004). A parallel hybrid genetic algorithm for the vehicle routing problem with time windows. *Computers & operations research*, 31(12), 2037-2053.
- [5] Chen, L., Chen, Y., & Langevin, A. (2021). An inverse optimization approach for a capacitated vehicle routing problem. *European Journal of Operational Research*, 295(3), 1087-1098.
- [6] Chen, X., Ulmer, M. W., & Thomas, B. W. (2022). Deep Q-learning for same-day delivery with vehicles and drones. *European Journal of Operational Research*, 298(3), 939-952.
- [7] Cordeau, J.-F., Laporte, G., & Mercier, A. (2001). A unified tabu search heuristic for vehicle routing problems with time windows. *Journal of the Operational research society*, 52(8), 928-936.
- [8] Czech, Z. J., & Czarnas, P. (2002). Parallel simulated annealing for the vehicle routing problem with time windows. In *Proceedings 10th Euromicro workshop on parallel, distributed and network-based processing* (pp. 376-383). IEEE.
- [9] Daely, P. T., Aruan, Y. J., Lee, J. M., & Kim, D.-S. (2021). Dynamic VRP Optimization Using Discrete PSO in Edge Computing Environment. In *2021 International Conference on Information and Communication Technology Convergence (ICTC)* (pp. 654-656). IEEE.
- [10] Dantzig, G. B., & Ramser, J. H. (1959). The truck dispatching problem. *Management science*, 6(1), 80-91.
- [11] Gambardella, L. M., Taillard, É., & Agazzi, G. (1999). MACS-VRPTW: A multiple ant colony system for vehicle routing problems with time windows. In: *Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale*.
- [12] Homberger, J. (2000). *Verteilt-parallele Metaheuristiken zur Tourenplanung: Lösungsverfahren für das Standardproblem mit Zeitfensterrestriktionen*. Fernuniv. Hagen, Diss., 2000 (1. Aufl.). Gabler Edition Wissenschaft. Wiesbaden: Dt. Univ. Verl.[ua].

- [13] Homberger, J., & Gehring, H. (1999). Two evolutionary metaheuristics for the vehicle routing problem with time windows. *INFOR: Information Systems and Operational Research*, 37(3), 297-318.
- [14] Ibaraki, T., Kubo, M., Masuda, T., Uno, T., & Yagiura, M. (2001). Effective local search algorithms for the vehicle routing problem with general time windows. In *Proc. 4th Metaheuristics Internat. Conf.*
- [15] Kong, F., Li, J., Jiang, B., Wang, H., & Song, H. (2022). Trajectory optimization for drone logistics delivery via attention-based pointer network. *IEEE Transactions on Intelligent Transportation Systems*.
- [16] Li, H., & Lim, A. (2003). Local search with annealing-like restarts to solve the VRPTW. *European Journal of Operational Research*, 150(1), 115-127.
- [17] Luo, J., Li, C., Shang, S., Zheng, Y., & Ju, Y. (2023). An Efficient Encoder-Decoder Network for the Capacitated Vehicle Routing Problem. Available at SSRN 4395195.
- [18] Luong, M.-T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.
- [19] Ma, Q., Ge, S., He, D., Thaker, D., & Drori, I. (2019). Combinatorial optimization by graph pointer networks and hierarchical reinforcement learning. *arXiv preprint arXiv:1911.04936*.
- [20] Mandi, J., Canoy, R., Bucarey, V., & Guns, T. (2021). Data driven vrp: A neural network model to learn hidden preferences for vrp. *arXiv preprint arXiv:2108.04578*.
- [21] Mester, D. (2002). An evolutionary strategies algorithm for large scale vehicle routing problem with capacitate and time windows restrictions. In *proceedings of the conference on mathematical and population genetics*, University of Haifa, Israel.
- [22] Mo, B., Wang, Q. Y., Guo, X., Winkenbach, M., & Zhao, J. (2023). Predicting Drivers' Route Trajectories in Last-Mile Delivery Using A Pair-wise Attention-based Pointer Neural Network. *arXiv preprint arXiv:2301.03802*.
- [23] Othman, W., Wahab, A., Alhady, S., & Wong, H. (2018). Solving vehicle routing problem using ant colony optimisation (ACO) algorithm. *International Journal of Research and Engineering*, 5(9), 500-507.
- [24] Park, S., Yoo, Y., & Pyo, C.-W. (2022). Applying DQN solutions in fog-based vehicular networks: Scheduling, caching, and collision control. *Vehicular Communications*, 33, 100397.
- [25] Qiu, H., Wang, S., Yin, Y., Wang, D., & Wang, Y. (2022). A deep reinforcement learning-based approach for the home delivery and installation routing problem. *International Journal of Production Economics*, 244, 108362.
- [26] Quirion-Blais, O., & Chen, L. (2021). A case-based reasoning approach to solve the vehicle routing problem with time windows and drivers' experience. *Omega*, 102, 102340.

- [27] Rathore, N., Jain, P., & Parida, M. (2020). A MATLAB-Based Application to Solve Vehicle Routing Problem Using GA. In *Advances in Simulation, Product Design and Development* (pp. 285-298): Springer.
- [28] Raza, S. M., Sajid, M., & Singh, J. (2022). Vehicle Routing Problem using Reinforcement Learning: Recent Advancements. In *Advanced Machine Intelligence and Signal Processing* (pp. 269-280): Springer.
- [29] Rochat, Y., & Taillard, É. D. (1995). Probabilistic diversification and intensification in local search for vehicle routing. *Journal of heuristics*, 1, 147-167.
- [30] Rousseau, L.-M., Gendreau, M., & Pesant, G. (2002). Using constraint-based operators to solve the vehicle routing problem with time windows. *Journal of heuristics*, 8, 43-58.
- [31] Schrimpf, G., Schneider, J., Stamm-Wilbrandt, H., & Dueck, G. (2000). Record breaking optimization results using the ruin and recreate principle. *Journal of Computational Physics*, 159(2), 139-171.
- [32] Shaw, P. (1997). A new local search algorithm providing high quality solutions to vehicle routing problems. APES Group, Dept of Computer Science, University of Strathclyde, Glasgow, Scotland, UK, 46.
- [33] Shaw, P. (1998). Using constraint programming and local search methods to solve vehicle routing problems. In *Principles and Practice of Constraint Programming—CP98: 4th International Conference, CP98 Pisa, Italy, October 26–30, 1998 Proceedings 4* (pp. 417-431). Springer.
- [34] Sheng, Y., Ma, H., & Xia, W. (2020). A pointer neural network for the vehicle routing problem with task priority and limited resources. *Information Technology and Control*, 49(2), 237-248.
- [35] Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations research*, 35(2), 254-265.
- [36] Srivatsa Srinivas, S., & Gajanand, M. (2017). Vehicle routing problem and driver behaviour: a review and framework for analysis. *Transport reviews*, 37(5), 590-611.
- [37] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.
- [38] Taillard, É., Badeau, P., Gendreau, M., Guertin, F., & Potvin, J.-Y. (1997). A tabu search heuristic for the vehicle routing problem with soft time windows. *Transportation Science*, 31(2), 170-186.
- [39] Vinyals, O., Fortunato, M., & Jaitly, N. (2015). Pointer networks. *Advances in neural information processing systems*, 28.
- [40] Wu, Y., Song, W., Cao, Z., Zhang, J., & Lim, A. (2021). Learning improvement heuristics for solving routing problems. *IEEE transactions on neural networks and learning systems*, 33(9), 5057-5069.

- [41] Xin, L., Song, W., Cao, Z., & Zhang, J. (2021). Multi-decoder attention model with embedding glimpse for solving vehicle routing problems. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 35, pp. 12042-12049).
- [42] Zhao, F., Si, B., Wei, Z., & Lu, T. (2023). Time-dependent vehicle routing problem of perishable product delivery considering the differences among paths on the congested road. *Operational Research*, 23(1), 5.

