逢甲大學學生報告 ePaper

校園無人車之 AI 強化學習餐飲派遣分析

Analysis of AI Reinforcement Learning Catering Dispatch for Unmanned Vehicles on Campus

作者：蔡怡玟、楊茗婷、陳玫婷
系級：運輸與物流學系三年乙班
學號：D0716278、D0716204、D0787881
開課老師：吳沛儒老師
課程名稱：專題研究
開課系所：運輸與物流學系
開課學年：一零九學年度 第二學期

中文摘要

現今外送服務平台興起，儼然成為一種主流，然而許多校園仍然禁止校外汽、機車進入校園中，導致訂餐者須親自到校門口取餐。因此，本研究目的為建構人工智慧之校園無人車派遣，並研擬相關物流策略，解決目前校園餐飲配送之困境。具體而言，本研究發展校園無人車之 AI 強化學習餐飲派遣方式，透過基本測試、敏感度分析與情境分析，探究不同物流派遣策略之影響。AI 校園無人車之分析結果顯示，以強化學習派遣之求解時間平均小於一秒，在派遣上具有可行性與有效性。物流策略分析結果指出多溫層無人車派遣策略較單一溫層佳，校園規模影響無人車配置規劃。進而，本研究藉由強化學習派遣分析，獲悉不同規模校園特性適宜之餐飲發車點數目、基本派遣車輛數、無人車容量、無人車類型。本研究成果在學術上可作為校園無人車強化學習相關研究之創新發展，實務上可解決餐飲外送無法進入校園之問題。

關鍵字：校園無人車、餐飲配送、強化學習、物流策略、車輛派遣

Abstract

Nowadays, with the rise of the delivery service platform, it has become a mainstream. However, many campuses still prohibit off-campus vehicles and

motorcycles from entering the campus, which leads to the fact that the diners have to pick up their meals in person at the school gate. Therefore, the purpose of this study is to build an artificial intelligence campus unmanned vehicle dispatch, and develop related logistics strategies to solve the current dilemma of catering distribution in campus. Specifically, this research develops AI reinforcement learning catering dispatch mode of campus unmanned vehicles, and explores the influence from different logistics dispatch strategies through basic test, sensitivity analysis and scenario analysis. The results from analysis of AI campus unmanned vehicles show that the solution time of the reinforcement learning dispatch is less than one second on average, which is feasible and effective in dispatch. The results of logistics strategy analysis show that the dispatching strategy of multi-temperature unmanned vehicles is better than those of single-temperature, and the campus scale affects the configuration planning of unmanned vehicles. Furthermore, through the reinforcement learning dispatch analysis, the number of catering departure points, the number of basic dispatched vehicles, the capacity of unmanned vehicles, and the type of unmanned vehicles suitable for different sizes of campus were obtained. The results of this research can be used as the innovation and development of the related research on campus unmanned vehicle reinforcement learning, and can solve the problem that the catering delivery can not enter the campus in practice.

Key words: campus unmanned vehicle, catering distribution, reinforcement learning, logistics strategy, vehicle dispatch.

目錄

表目錄

圖目錄

第一章、前言

1.1 研究動機

根據經濟部統計處資料，在 109 年 8 月，有外送和無外送之年增率分別為 5.1 與 1.3。除了商家能夠獲利，消費者也可以妥善利用額外時間，外送助餐飲業趨勢逐漸興起。雖然如此，但許多校園仍禁止校外汽、機車進入校園中，導致訂購者仍要到校園交接處取餐。行走時間除了走至指定門口、也需包含由指定門口至下一個目的地之時間，將整體時間拉長。

除此之外，每逢中午及傍晚用餐時刻，便會看到許多學生同時出校門進行餐點購買，在每間店面皆可看見顧客漫長等待之情況。因此，若中午及傍晚用餐時刻之尖峰時刻，事先在校園內分散一些客源，便能使校園內外部用餐者之「購買」動作更加有效率完成。

對於將門口視為發車點，再配送至各需求點之行為，以如何派遣無人車至各需求點進行餐飲配送視為最重要部分。不同校園規模中對於車輛派遣數目、行駛距離、發車點數目與車隊管理之相關車輛路徑規劃問題為無人車餐飲物流配送關鍵。

傳統車輛路徑問題以往之研究多用傳統求解法求出最佳解，但其演算時間較長、無法即時反應求解。因此近來趨勢多以使用強化學習法解決問題，進而提升效率與準確度。

綜上所述，本研究動機歸納如圖 1 所示。

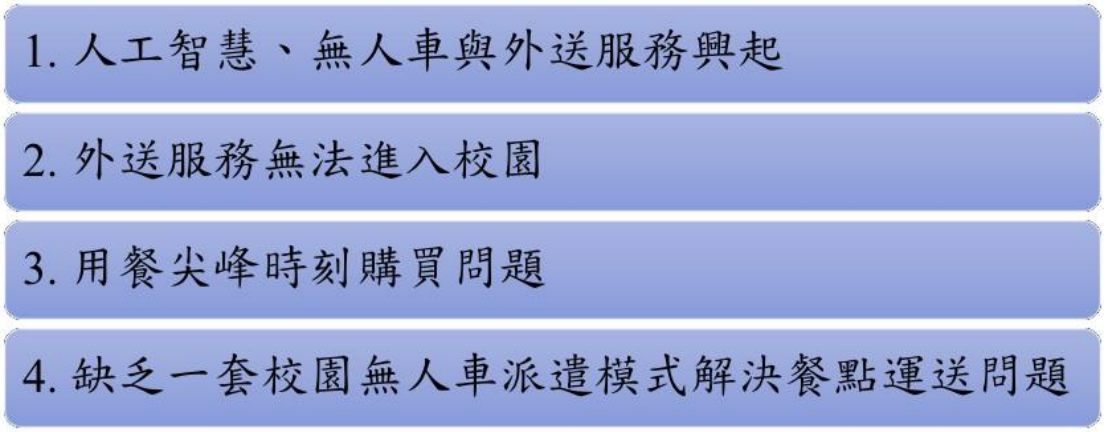


圖 1　研究動機

1.2 研究目的

本研究事先於校園內設置數個需求點，並經過派遣，使餐飲無人車經過該需求點、提供餐點給沒有時間到校外進行午餐購買之教師及學生購買，此構想可有效提升個人時間效率、靈活運用時間。

路線選擇也會大幅影響無人車行駛一趟所需之運輸時間，如同餐廳講求翻桌率，餐飲無人車也需在最小化時間之前提下盡量提升使用率，且因資料量過大，並非使用人工手算可得出，計算時需仰賴一般概念及相關數學模式，並帶入電腦

運算得出結果。經過研究探討、規劃不同情境進行敏感度分析與情境測試，透過情境設想探討發車點與需求端間之物流策略，包括多場站產生、車輛派遣規模數量等，以解決校園車輛路徑派遣問題並獲取校園無人車適宜之派遣策略。

本研究利用強化學習(Reinforcement Learning , RL)處理校園無人車派遣之車輛路徑問題(Vehicle Routing Problem , VRP)。透過自主學習，讓所選定代理人，藉由動態環境不斷學習及重複互動，得到最大化正獎勵值(例如最小化距離、時間等目標導向)，建立強化學習之神經網路，使無人車能夠順利、安全、無阻礙的在規劃之路線上行駛(Bogyrbayevay et al., 2020; Delarue et al., 2020; Kalakanti et al., 2019)。同時使用傳統方法進行相同求解，以最後結果對比兩者輸出效益及效率。綜上所述，本研究目的歸納如圖 2 所示。

圖 2 研究目的

1.3 研究範圍

本研究針對車輛路線進行相關分析與探討，使用虛擬環境作為車輛派遣配送之分析與探討，並以逢甲大學為模擬環境場地，設想服務對象之各種不同餐飲需求量之情境，並進行物流策略規劃。

本研究之課題、對象、空間、時間分以下四點說明：

1. 研究課題：透過強化學習求解無人車最佳路徑解，解決飯點時刻店家外買飯人潮壅擠問題。
2. 研究對象：各大樓整體用餐者所有需求視為一單位，並以各大樓作為需求端做估量。除此之外，以校門口為餐飲配送中心。本研究針對發車點與需求端間配送路徑及物流策略之研擬為主軸探討。
3. 研究空間：以一校園為例，包含大樓與校門口處餐飲配送中心所構成之範

圍。本研究主要以逢甲大學各大樓需求點進行路線模擬規劃及車輛路線問題中虛擬需求點位置與距離量測，透過研擬各種情境得到最佳物流策略。

4. 研究時間：本研究以用餐尖峰時間進行相關規劃，透過容量限制、車輛派遣數、車輛類型、需求量與發車點等參數分析，了解派遣最佳化模式，進而研擬物流策略，並進行餐飲配送規劃分析。

1.4 研究步驟

本研究之步驟主要著重在校園無人車派遣之問題特性與情境訓練分析，根據前述問題特性建構出校園無人車派遣之強化學習架構，並使用強化學習模型代碼進行測試，再與 Lingo 結果做比較，發展出一套最佳校園無人車派遣模式，最後修改此模型參數，並進行情境分析，研擬出校園無人車派遣之相關物流策略，如圖 3 所示。

圖 3 研究步驟

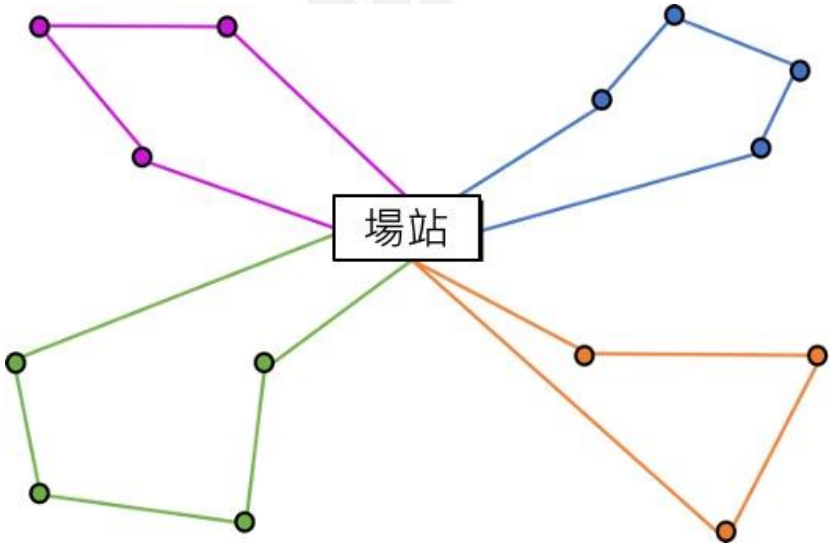
第二章、文獻回顧

本章節中，擬介紹與本研究相關之文獻回顧。架構如下，首先介紹車輛路徑問題及其使用之解決方法，再介紹強化學習重要元素、工作流程、強化學習演算法及其相關應用與平台。

2.1 車輛路徑問題

車輛路徑問題(Vehicle Routing Problem, VRP)為旅行推銷員問題(Travelling Salesman Problem, TSP)之延伸，為多個旅行推銷員問題之組合，目的為求解最佳化組合問題。例如：運輸成本最小化、運輸時間最短、運送路徑最短等，求解時間隨需求點數增多而時間增長，屬於 NP-Hard 問題。

圖 4 為車輛路徑問題示意圖。其基本概念為有一個場站(Depot)、X 台車行駛不同路線(Route)、N 個需求點(Customer)，事先將需求點分成不同群集，藉由車輛經過多組路線來滿足運輸行為，每個需求點皆需被車隊車輛經過一次，當全部需求點皆被通過後，車隊再回到場站，此為一次運輸行為。此外，車輛路徑問題具有容量限制、行駛里程、貨物需求量、發貨量等限制參數，並以車隊模式去做運行，組織出最適當之運輸路線，且車輛路徑問題大部分皆以 2 臺車以上作為路線運輸設計。



資料來源：本研究彙整

圖 4　車輛路徑問題示意圖

車輛路徑問題為車輛排程中之基礎模型，又可分為靜態車輛路徑問題(Static Vehicle Routing Problem, SVRP)與動態車輛路徑問題(Dynamic Vehicle Routing Problem, DVRP)。且車輛路徑問題並未能完全處理相關實際問題，例如時間、容量限制、多場站、非單一收發貨任務等，因此會演伸出其他延伸模型，這些皆屬 VRP 衍生類型之範疇。

在求解 VRP 過程中，有幾個經常使用之方法，例如：Sweep、OR-Tools、VRP 及強化學習。在過去文獻中(Ferrara,2018)便使用 Sweep 來解決 VRP 問題、Arun Kumar Kalakanti(2019)中以 TSP 組合模式來求解 VRP，OR-Tools 為現今最常使用求解 VRP 之工具；而強化學習為目前人工智慧最為重要之研究領域之一，其

有相當多演算法來因應決策者需求。

2.2 強化學習

強化學習(Reinforcement Learning , RL)，為機器學習其中一種學習方式。透過在不藉助監督者提供之完整指令下以訓練不需對人工智慧進行任何干預。除此之外，同時在沒有被寫入明確執行任務程式之情況下，使其不斷與環境產生重複性地互動，並隨著動態環境中外部條件變化而改變資料結構。總結而言，透過自主學習思考方式，將每一次產出之錯誤決策加以改善，直到足以反應任何情況，並做出相對應行動。以產出最佳結果及獲取最大報酬之策略為目標。(Lapan, 2019)

1. 強化學習 vs 監督式 vs 非監督式

強化學習、監督式學習與非監督式學習皆為機器學習之學習方式，但其訓練方式有所不同，以下為其相關介紹。

（1） 強化學習

強化學習設置獎勵系統，行動將帶來正回饋與負回饋，而無影響則會獲得中立獎勵。其目的為最大化獎勵值。在資料處理部分，所有資料以無標籤化方式進行分析。因強化學習有延時缺點，無法立即得知結果為正值或負值，需在未知領域和現有知識間進行權衡比較。因此，代理人前次行為影響下次機器決策輸入值，並不會完全為絕對正向結果。總結而言，強化學習為透過代理人觀察行動所帶來獎勵關聯性後，透過不斷學習、以漸進方式偏向正確方向，進而減少錯誤率。

（2） 監督式學習(Supervised Learning)

此種學習方法所使用之所有訓練資料將被進行標籤化，其標籤代表機器輸入至輸出感應憑據。監督是學習過程需仰賴大量事前人工分析，將資料中所有可能特質進行標記，為獨立式分布系統。使機器學習在輸出時，藉以用來判斷其誤差標準。因此，如果做了什麼樣之選擇，則將會立刻反饋給其算法。

（3） 非監督式學習(Unsupervised Learning)

非監督式學習與監督式學習完全相反，主要以電腦進行主導，所有資料無進行標註動作。為找出無標籤數據間內部關聯性，機器只能透過資料特徵來尋找、進而分類相對應值，並依照關聯性去歸類處理以及找出資料規律性，並依此形成集群。

2. 強化學習工作流程

圖 5 為強化學習於環境模擬中之基礎運作模型，以此流程來訓練人工智慧主體，主要包含為「代理人」與「環境」以及兩者間之溝通管道，即「觀察」、「行動」與「獎勵」，下述為其介紹(Emmanouil Tzorakoleftherakis, 2019)。

（1） 代理人(Agent)

代理人即訓練主體，透過環境根據代理人行動提供相對應「獎勵」，即第一管道資訊與環境「觀察」周圍發生情況提供之第二管道資訊，代理人透過前兩者得到之資訊，運用策略與訓練之演算法，以選擇最有效率之行動解決問題。

（2） 環境(Environment)

環境為一虛擬模型，此為根據代理人執行之行動，於環境中進行模擬，並根據其行動給予相對應之獎勵。

（3） 行動(Action)

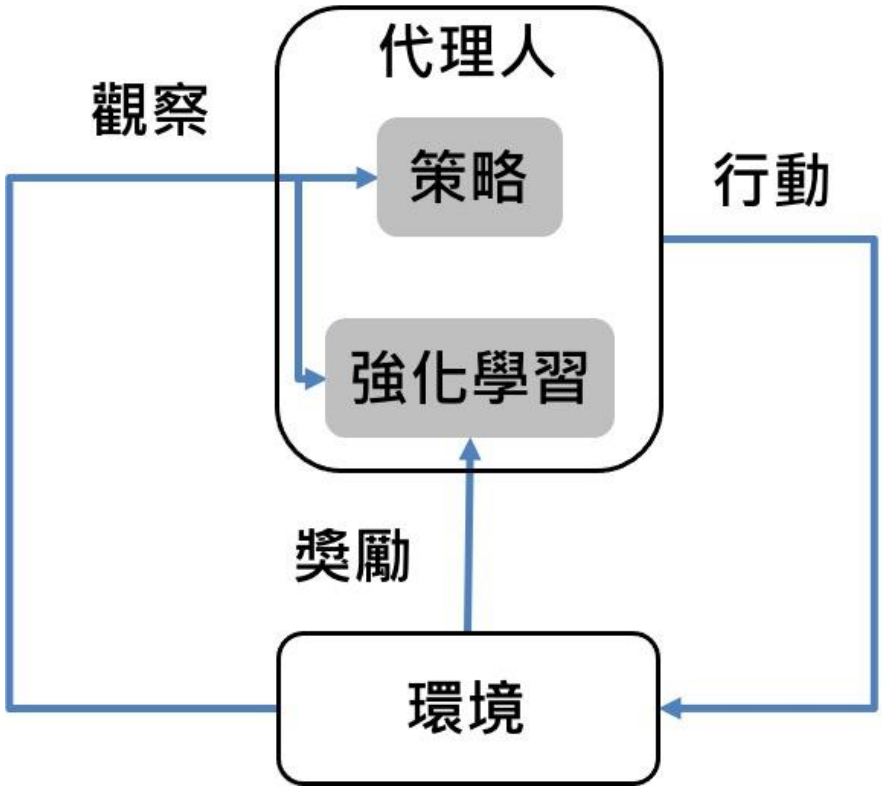
此為代理人根據環境狀態與策略，於環境中採取相對應之行動，以取得最大化獎勵值為目的。行動可分為離散型(Discrete)與連續型(Continuous)，前者主要為講述代理人可採取之行動為有限且互斥之行為集合，例如：向左或向右移動。後者主要是講述連續行動具有一定之附加價值，例如：操控汽車將包含車輪旋轉角度與方向。

（4） 獎勵(Reward)

獎勵值主要為顯示代理人行動表現之結果，並從環境中獲得不定值。獎勵之回饋值，其值可正可負，代理人將再根據獎勵值顯示結果做相對應行動調整。

（5） 觀察(Observation)

主要為說明周圍發生之情況，也稱為狀態(State)，透過觀察提供給代理人第二資訊管道，且與獎勵提供之資訊相互應，因此代理人可根據此資訊執行不同之行動措施。



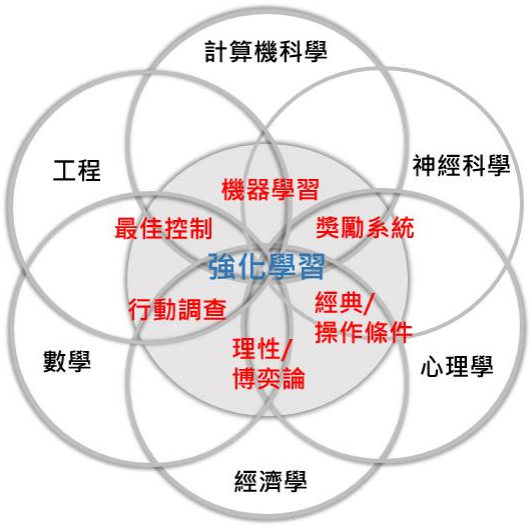
資料來源：TeraSoft(2020)

圖 5 強化學習概要圖

強化學習為許多應用領域中心思想，具有通用性和靈活性，不同領域中都能從其獲得利益，分別為機器學習、工程、神經科學、心理學、經濟學與數學，如圖 6 所示。此六大領域為強化學習最具關聯性學科，在某些特定主題進行決策

時因強化學習概念而在做法上會有高度重疊性，其目的皆為以「如何優化決策以實現最佳結果」為主。

機器學習(Machine Learning, ML)領域以強化學習為瞭解研究主體在給予不完整之觀察數據之情況，會以何種行動進行回應做為重點核心，因此會借用機器學習中之許多機制、技巧和技術，才得以完成這項指令。工程(Engineering)領域主要研究最佳化控制，採取一連串最佳行動來獲得最佳結果。神經科學(Neuroscience)領域為人類透過大腦研究出強化學習算法之獎勵系統，並在強化學習模型中使用。心理學(Psychology)領域主要是研究強化學習在各種條件下會反應之動作，例如：經典條件反射和操作性條件反射等。經濟學(Economics)領域為研究理性博弈論，在不同環境之變化條件下與不完整之知識數據，如何才能得到最大之獎勵。數學(Mathematics)領域為研究運籌學，主要是找尋並達到最佳之條件。



資料來源：Silver (2018)

圖 6 應用強化學習多種領域

3. 強化學習平台

（1） OpenAI Gym and Universe

A. OpenAI Gym

此程式平台較常用於建置、評估和比較強化學習演算法之好壞。其可解決模型於環境測試過程中所遇到之問題，使應用之演算法將得到較好求解值。且此平台相容於各種框架之演算法，所以相當簡單好懂，對代理人之模式架構上並無要求，為 RL 提供了相當不錯之介面。且其含有遊戲介面，對於 RL 將有助於得出通用性更強之算法。

B. OpenAI Universe

此為 OpenAI Gym 之擴展平台，提供簡易至複雜與即時環境等，且能完全控制多個遊戲環境。其環境複雜程度恰似戰略遊戲，對於決策之時間有其要求，且因為其為遠端桌面來運行程式，所以不會動用到程式和新原始碼或 API，就可將任何程序轉換為 Gym 環境。因此，大多以此平台作為訓練與評

估代理人之效能。

（2） RoboSchool

此平台為基於 Bullet 所開發之物理引擎，提供了 12 種環境模式，其中包含傳統且類似於 MuJoCo 場景、交互控制以及多智能體控制場景等。此平台目前所包含之環境皆為 OpenAI Gym 形式接口，並用於模擬機器人之控制。

（3） DeepMind Lab

DeepMind Lab 為一項 AI 代理研究與 3D 遊戲平台。提供豐富科幻視覺場景模擬環境，其操作使智能體以 3D 形式移動方式環顧四周。此平台可視為執行各種 RL 演算法之實驗室，客製化與擴充性程度相當優異。

（4） Project Malmo

此為一款由 Microsoft 研究員 Katja Hofmann 建立於 Minecraft 創世神遊戲上之人工智能實驗和研究平台，其對於自行修改環境之彈性有益，且整合了較複雜環境，並用於協同 AI 挑戰賽。其允許程式設計者可使用比標準值 Minecraft 更快地速度來處理相對應之場景。且 Malmo 為多智能體 RL 演算法開源平台，要求各智能體之間相互合作，並將協同 AI 達到最佳。但是 Malmo 目前只提供 Minecraft 遊戲環境，此為與 Open AI Universe 不同處。

（5） ViZDoom

此平台為一款類似於以毀滅戰士遊戲（Doom）為基礎之 AI 平台，提供以多智能體競爭博弈之環境。除此之外，並用於測試多重代理與競賽型環境之優劣。但是此平台只能使用於毀滅戰士遊戲虛擬環境中進行後台彩現與單一/多重玩家模式。

（6） Amazon SageMake

為雲端第一個受管 RL 服務，可使開發人員透過受管對 RL 進行建立、訓練以及部署。Amazon SageMake 支援多種架構與多元模擬環境。除此之外，與 AWS RoboMaker 新機器人服共同進行整合。

4. 馬可夫決策過程(Markov Decision Processes, MDP)

馬可夫決策過程目的為解決各種最佳化之問題，且大部分強化學習問題也都用 MDP 模型表示。馬可夫決策過程為處理並決策馬可夫鏈(MP)為一隨機轉移機率問題之數學模型。因此，其演算過程皆具隨機性，且下一個狀態轉移只與當下之狀態有關，與前一狀態無關(hankla, 2020)。

馬可夫決策過程是由五個元件所組成，包含狀態(State, s)、行動(Action, a_t)、轉移機率(Transition Probability)、獎勵機率(Reward Probability)及折扣因子(Discounted Factor)。首先，狀態(s)為一有限且可實際處於其中之狀態集合;而行動(a_t)為所有可執行行動集合;轉移機率為從某個狀態移動到另一個狀態之機率;獎勵機率則為當代理人(Agent)執行行動(a_t)後，從某個狀態轉移到另一個狀態所收到之獎勵機率。最後，折扣因子以γ表示之，是決定目前與未來獎勵之重要程度，用來使預期獎勵(R_t)最大化(陳信宇，2010)。

公式(1)為在狀態(s)下，採取行動(a_t)後，轉移至下一個狀態s'的機率。t 是指時間。

$$P_{a_t}(s, s') = P(s_{t+1} = s' \mid s_t = s\,, a_t = a) - (1)$$

而公式(2)為在狀態(s)下，執行行動(a_t)後，狀態轉移到下一狀態s'之預期獎勵值。

$$R_{a_t}(s,s') = \mathrm{E}\{r_{t+1}|s_t = s, a_t = a, s_{t+1} = s'\} - (2)$$

為了使預期獎勵能最大化，遂加入折扣因子，如公式(3)所示。

$$R_t = r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \cdots = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1} - (3)$$

5. 策略函數(Policy Function)

主要目標為盡可能多收集累積獎勵之回傳值，進而找出一項最佳策略，使得該策略所獲之預期獎勵最大化(Huang, 2016; Lapan, 2019; Ravichandiran, 2019)，其符號常以π做表示。公式(4)為策略函數之行動對於所有可能狀態下之機率公式。

$$\pi(\mathrm{a}|\, s) = P[A_t = a \mid S_t = s] - (4)$$

6. 價值函數(Value Function)

隨著所選之策略**π**不同，價值函數也會有所不同，分別為狀態-價值函數及狀態-行動價值函數。

（1） 狀態-價值函數(State-Value Function)

公式(5)為其狀態-價值函數之定義(Ravichandiran, 2019) ，通常以 V(s)做表示。

$$V^{\pi}(s) = E_{\pi}[R_t \mid s_t = s] - (5)$$

公式(6)表示在狀態(s)下，根據策略π後，所得到之預期獎勵值，可以$V^{\pi}(s)$表示。而顯現出來之狀態值越高代表狀態愈好。

$$V^{\pi}(s) = E_{\pi}[\sum_{k=0}^{\infty} \gamma^k r_t + k + 1 \mid s_t = s] - (6)$$

（2） 狀態-行動價值函數(Action-Value Function)

公式(7)為 Q 函數之定義(Ravichandiran, 2019)。也稱為 Q 函數，大多以 Q(s)來表示。意指在使用策略π後，於當前之狀態中執行指定行動之價值。

$$Q^{\pi}(s,a) = E_{\pi}[R_t \mid s_t = s, a_t] - (7)$$

7. 強化學習常用演算法模式

強化學習於 VRP 和 TSP 之車輛路徑問題所建構的模型中，相對於其他相關聯之強化學習方法，較常出現並應用之演算法模式有貝爾曼方程式、Q–Learning 法、DQN、策略梯度法、Sarsa 法，此部分將於後面做詳細介紹。

2.3 強化學習相關模式

在本節中，將會介紹強化學習中之相關應用方法，了解方法之相關介紹、運作流程與其應用。

2.3.1 貝爾曼方程(Bellman Equation)

貝爾曼方程亦稱為動態規劃方程(Dynamic Programming Equation)。由美國應用數學家理查·貝爾曼(Richard Bellman)提出，用於簡化強化學習或求解馬爾可夫決策過程(Kumar, 2020)，意指找到最佳策略與價值函數。

公式(8)為貝爾曼方程中，利用 Q 函數最大值便能求得最佳價值函數。此公式運用 Q 函數最大值找到最佳價值函數$V^*(s)$之公式(Ravichandiran, 2019)。以最佳價值函數$V^*(s)$做為目標式，較能得到最高值，使 Q 函數同時也為最大值狀態。

$$V^*(s) = max_a Q^*(s,a) - (8)$$

公式(9)為求解出最佳價值函數，因此，將 Q 函數以貝爾曼方程表示，得到 Q 函數貝爾曼方程式(Ravichandiran, 2019)。

$$Q^\pi(s,a) = \sum_{s\prime} P^a_{ss\prime}\left[R^a_{ss\prime} + \gamma \sum_{a\prime} Q^\pi(s\prime, a\prime)\right] - (9)$$

公式(10)為貝爾曼最佳方程式(Bellman Optimality Equation) (Ravichandiran, 2019)，為公式(9)帶入公式(8)所求。

$$V^*(s) = max_a \sum_{s\prime} P^a_{ss\prime}\left[R^a_{ss\prime} + \gamma \sum_{a\prime} Q^\pi(s\prime, a\prime)\right] - (10)$$

貝爾曼最佳方程式(Bellman Optimality Equation)為一遞迴方程式，可透過動態規劃(Dynamic programming, DP)求解貝爾曼最優性方程式(Bellman Optimality Equation)，找到最佳價值函數及最佳策略(Ravichandiran, 2019)。動態規劃(Dynamic Programming)為最常見之優化算法，用於求解複雜問題有效方法。即為透過將問題簡化成數個子問題，在尋找子問題解同時，進而求解出整個問題最優解。在求解子問題時，將其解儲存於表格中，因此，若反覆出現相同子問題時，並不需要重新計算，直接採用先前表格求解值。透過不斷重覆利用已儲存之子問題解，以節省計算時間(Silver, 2017)。

動態規劃有兩種演算法，分別為價值迭代(Value Iteration)和策略迭代(Policy Iteration)，以下擬做相關介紹。

1. 價值迭代(Value Iteration)

價值迭代為透過遞迴方式找出一最佳價值函數 V(s)，並從最佳價值函數中找出最佳策略。其過程首先初始化一隨機價值函數 Q(s,a)，再計算每一狀態-行動之值。除此之外，並從中選擇最大 Q 值作為當前狀態之價值函數，並進行更新。透過反覆執行以上步驟直到價值函數收斂。本研究引述 Sudharsan Ravichandiran

所整理之價值迭代步驟，如圖 7 所示(Ravichandiran, 2019)。

資料來源：用 Python 實作強化學習 使用 TensorFlow 與 OpenAI Gym

圖 7 價值迭代步驟

2. 策略迭代(Policy Iteration)

策略迭代包含策略評估（Policy Evaluation）及策略改進（Policy Improvement）兩部份。其過程為，給定一隨機策略函數並將其初始化，在此策略下計算出價值函數，並進行策略評估，評估其是否為最佳函數。再者，若此價值函數並非為最佳，則進行策略改進，透過採取貪婪(Greedy)方法計算此價值函數來不斷更新策略，直到策略收斂，找到最佳策略為止。本研究引述 Sudharsan Ravichandiran 所整理之策略迭代步驟，如圖 8 所示(Ravichandiran, 2019)。

資料來源：用 Python 實作強化學習 使用 TensorFlow 與 OpenAI Gym

圖 8 策略迭代步驟

2.3.2 Q 學習(Q-Learning)

Q-Learning 為一決策過程，為值迭代法之變形，以反覆學習求得最佳行動值(王俊勝、駱聖文，2019)。為一種 TD 演算法(Sutton, 1988)。「行動值」(Value Of Action)為當在「狀態(s)」上執行「行動(a)」時，可獲得「總獎勵」$Q_{s,a}$，即為 Q 值。此一系列行為便稱為 Q 學習，為一「資料庫」概念(Lapan, 2019)。

「狀態值」(Value Of State)可用$V(s)$來定義，指透過馬可夫獎勵過程所得到平均期望值，其定義如公式(11)。γ_t為這回合在時步t所取得之「區域獎勵」(Lapan, 2019)。

$$V(s) = \mathbb{E}[\sum_{t=0}^{\infty} \gamma_t r^t] - (11)$$

依照公式(11)，可用總獎勵$Q_{s,a}$定義狀態值$V(s)$，如公式(12)所示。意味著某「狀態值」等於由該「狀態」所選擇最大行動值。

$$V(s) = \max_{a \epsilon A} Q_{s,a} — (12)$$

Q-Learning 通常會以表格化方式顯示，稱為表格 Q 學習，如表 1 所示。

表 1 表格 Q 學習

	行動1(無人車往右方移動)	行動2(無人車往左方移動)
狀態1(初始)	-3	9
狀態2(第二次)	-5	15
依序	依序	依序

舉例而言，情境假設在某校園中，若有一台無人車在運送貨物，路徑選擇有往左和往右兩個選項，並將目的地設在最左方。因此，選擇右方道路之行動會離目標地越來越遠。在無人車每次選擇中，逐漸了解到往左可以愈加靠近目的地，且可完成一次運輸任務，便會在每次選擇中加以修正，即為 Q-Learning 精神，無人車即為代理人，校園為環境，觀察則為狀態，具有明確性，且選擇哪種行為之行動一定會成功，經驗值即為獎勵，一但選擇便會結束該回合，同時開啟新回合，讓無人車繼續學習，並將獎勵值化為數值來回饋，其公式為(13)所示，其中折扣因子γ(Discounted Factor)位於 0~1 之間，代表代理人之前瞻性。若γ=0，代表除了當下獎勵外，其餘皆不重要，而γ=1，則代表所有獎勵皆很重要，需要考慮點就更多。因此狀態越遠，γ指數越大，代表對當下狀態而言越不重要(Ravichandiran, 2019)。

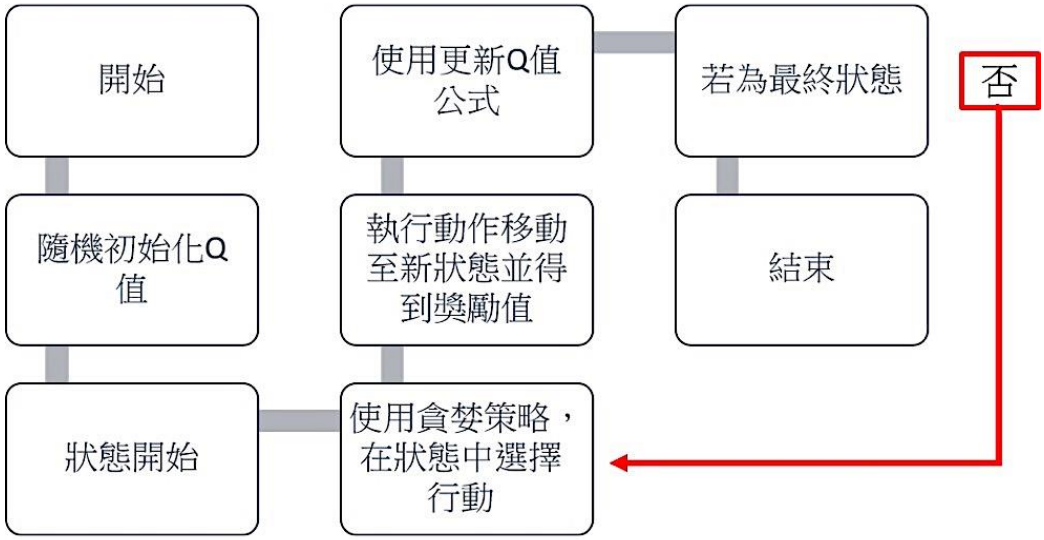
$$Q_{s,a} = \mathbb{E}\left[R_{t+1} + \gamma \max_{a'} q^*(s', a') | s, a\right] — (13)$$

若要使 Q-Learning 循環結束，便需要不斷更新 Q 值，常見手法為使用混合方法，如公式(14)，以近似值方式來更新$Q_{s,a}$。α代表學習速率(Learning Rate)，介於 0~1 之間，將新舊之 Q 值平均用以做更新前一個狀態之 Q 值(Lapan, 2019)。

$$Q_{s,a} \leftarrow (1-\alpha)Q_{s,a} + \alpha(r + \gamma max_{a' \epsilon A} Q_{s',a'}) — (14)$$

在開始一個 Q 學習時，先以隨機數值初始化 Q 函數，表示狀態世代開始，

並使用貪婪策略公式，在狀態中選擇行動，由此獲得直接獎勵，同時轉移至新狀態。除此之外，在獲得獎勵中加上此狀態長期值，便為最佳解。接下來再使用公式更新前一個狀態 Q 值，若為最終狀態，則結束，否則需由貪婪策略開始重複執行行動，直至結束，其操作順序如圖 9 所示(Ravichandiran, 2019)。



資料來源：用 Python 實作強化學習 使用 TensorFlow 與 OpenAI Gym

圖 9 Q學習操作步驟

2.3.3 策略梯度

策略梯度以基於策略(Policy-based)做為演算法，優點為可在一個連續環境下挑選行動。其透過參數θ得到最佳策略，以計算每個狀態對應之行動機率，並透過獎勵值(Reward)左右神經網絡之反向傳遞。

策略梯度主要根據當前之狀態來選擇行動。利用神經網絡輸入狀態(State)，神經網絡就會輸出狀態中每個行動之機率，透過反向傳遞讓神經網絡達到收斂，並且利用獎勵值確定此行動是否應該在下次增加被選之概率。當一個行動得到越多獎勵，即會增加其下次出現機率。反之，當一行動得到越少獎勵，便會降低其出現機率(莫煩，2016)。為了使獎勵最大化，可採用策略梯度算法對參數進行優化，此僅介紹蒙特卡洛策略梯度(REINFORCE)及演員-評論者算法(Actor-Critic)兩種策略梯度算法。

1. 蒙特卡洛策略梯度（REINFORCE）

蒙特卡洛策略梯度使用蒙特卡洛方法，進行隨機均勻抽樣，利用樣本估計得到之累積折扣回報來估計真實回報值。其求解出梯度（∇J）不存在誤差（bias），但具有方差(variance)，如公式（15）所示(Weng, 2019)。首先隨機初始化狀態 s，利用蒙特卡洛之採樣得到一個完整之回合(episode)，並儲存這些完整回合之(s, a, r, s')，計算每個時間步 t 之累積回報值及預期獎勵G_t，透過梯度上升法來更新策略參數 θ，以利最小化損失，重複採樣行動直到收斂為止。

$$\nabla J \approx \Xi[Q(s,a)\nabla log\pi(a|s)] \quad —（15）$$

2. 行動者-評論者演算法（Actor-Critic）

行動者-評論者演算法結合策略梯度和 Q-Learning。策略梯度屬於行動者，基於概率選擇行動；Q-Learning 屬於評論者，以評判行動之得分，再依據評論者評分選取最佳行動。優點為能在連續行動區間選取適合之行動，又可進行單步更新，相較單一策略梯度能夠有效率地更新。除此之外，行動者-評論者演算法透過評論者進行行動評估，便使得收斂更加困難(莫煩，2017)。

2.3.4 深度 Q 網路(Deep Q Network, DQN)

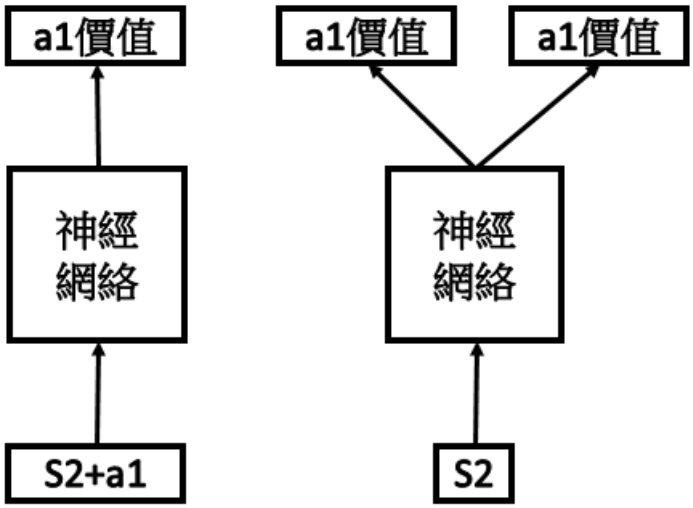
傳統強化學習演算法中，因有許多種狀態(State)且過於複雜，所以表格中無法儲存並應對每一個狀態與行為(Action)所擁有之 Q 值(莫煩，2017)。

傳統強化學習演算法中，因有許多種狀態(State)且過於複雜，所以表格中無法儲存並應對每一個狀態與行為(Action)所擁有之 Q 值(莫煩，2017)。

為了處理這項問題，使用結合神經網絡與 Q-Learning。圖 10 為其兩種形式之模式：(1)狀態+行為→神經網絡→行動 Q 值；(2)狀態→神經網絡→行動 Q 值(莫煩，2017)。並套用公式(16)，其為 Q-Learning 之更新規則，找到狀態-行動組之最佳 Q 值，此方法即是 Deep Q Network(Ravichandiran, 2019)。

$$Q(s,a) = Q(s,a) + \alpha\big(\mathrm{r} + \gamma \mathrm{maxQ}(s'a') - Q(s,a)\big) \quad —(16)$$

其中$Q(s,a)$為 Q Function 之預測值，用於根據狀態估計獎勵。從狀態 s 和行動 a 計算預期之未來值(莉森揪，2018)。其中，$\mathrm{r} + \gamma \mathrm{maxQ}(s'a')$為目標值。



資料來源：莫煩(2017)

圖 10　神經網絡作用

公式(17)優化目的公式，旨在最小化損失函數，其函數值=目標值-預測值之平方，並更新權重θ，直至將損失降到最低(Ravichandiran, 2019)。其中，$y_i = r + \gamma max_{a'} Q(s', a'; \theta)$ 為目測，$Q\big(s, a\ ; \theta\big)$為預測值、$\theta$為神經網絡。

$$\mathrm{Min}\ \ \mathrm{Loss} = (y_i - Q\big(s, a\ ; \theta\big))^2 —(17)$$

DQN 運作方式目的中，為學習當前與過去經驗之緩衝區(shura_R, 2018)。代理人執行某項行動a，於此緩衝區進行訓練以及新舊數據比較後，將較適宜之新狀態數據s'取代舊狀態經驗數據s，此即為經驗回放(Experience Replay)。於緩衝區隨機選取經驗，可解決神經網絡經驗之間因相關性而產生之過度擬合(Overfit)，使神經網絡更加有效率(Ravichandiran, 2019)，此即為回放緩衝(Replay Buffer)。

公式(17)目標 Q 值與預測 Q 值皆採用相同之θ網路來計算，所以兩者之間會有相當程度之發散。因此，此部分將兩者之網路獨立，公式(18)為此修正損失函數。將目標 Q 值之參數改為θ'，預測 Q 值則會運用隨機梯度下降法來學習正確之θ權重，藉此穩定訓練過程(Lapan, 2019; Ravichandiran, 2019)。

$$\text{Min Loss} = (r + \gamma max_{\alpha'} Q(s', a'; \theta') - Q(s, a\ ; \theta))^2 \text{—(18)}$$

雙層 DQN(Double DQN)由於基本 DQN 傾向於高估 Q 值，當在估計狀態 s 之所有 Q 值中含有雜訊，與實際不符，估計之行動會比最佳行動來得高，因而可能選到次佳策略(Lapan, 2019)，因此本研究提出優先經驗回放或重播緩衝區針對基本 DQN 經驗回放之抽樣問題進行改善，以及提出競爭網路結構針對基本 DQN 之 Q 函數應用架構有更好之訓練方式。公式(19)為基本 DQN 計算目標 Q 值之公式，將原本為目標網路之θ'用來選擇行動(Ravichandiran, 2019)。

$$Q(s, a) = r + \gamma max_{\alpha'} Q(s', a'; \theta') \text{ —(19)}$$

公式(20)為雙層 DQN 修改之公式，使用訓練網路進行評估所選之行動，可完整解決此 Q 值高估之問題(Lapan, 2019; Ravichandiran, 2019)。

$$Q(s, a) = r + \gamma Q(s, armaxQ(s, a; \theta^-)\ ;\ \theta') \text{ —(20)}$$

1. 優先經驗回放或優先重播緩衝區

由於基本 DQN 經驗回放之轉換為單一隨機抽樣策略屬於高度相關，且環境大多數為穩定狀態，所以不會因為選用不同行動而有變化。因此於雙層 DQN 提出樣本「優先等級」制度，並區分為公式(22)與公式(23)兩種優先權類型，兩式分別計算完後，將會使用公式(21)，進行優先權轉換(Schaul, 2015; Ravichandiran, 2019)。

$$P_i = \frac{P_i}{\sum_k P_k} \text{ —(21)}$$

（1）比例優先權(Proportional Prioritization)

$$P_i = (\delta_i + \epsilon)^a \text{—(22)}$$

以下為公式(22)註解。其中P_i為轉移 i 優先權;δ_i為轉移 i 之 TD 誤差，TD

誤差較高，賦予優先權就較高;ϵ為大於 0 之常數;a為優先權大小，當a為零時，此為與基本 DQN 異同之均勻分布採樣。

（2）排序優先權(Rank-based Prioritization)

$$P_i = (\frac{1}{rank(i)})^a \text{—(23)}$$

其中 rank(i)代表轉移 i 在緩衝區之位置。

2. 競爭網路結構

由前述 Q 函數之介紹可知價值函數 V(s)為代理人於某個狀態 s 中之良好程度；而優勢函數 A(a)代表其相較於其他行動，代理人執行行動 a 之良好程度。此部分競爭 DQN 之 Q 函數則為兩者之加總。如公式(24)，可發現其相比於基本 DQN 之架構更能精準估計 Q 值(Ravichandiran, 2019)。

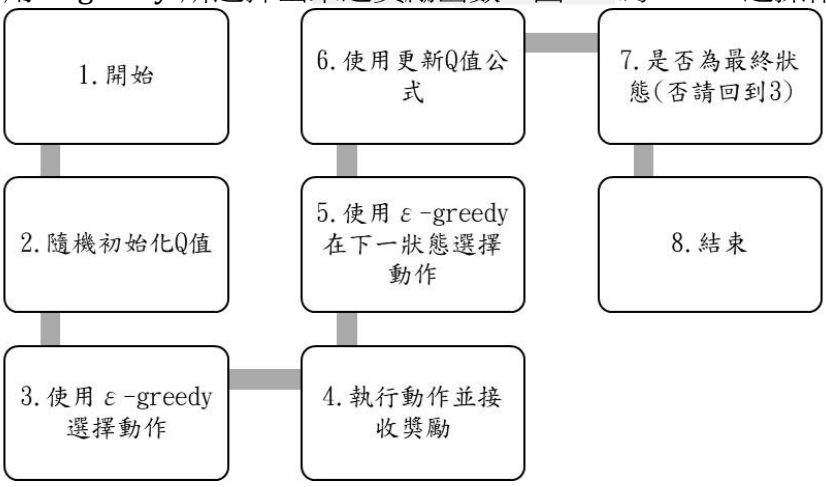
$$Q(s,a) = V(s) + A(a) \text{ —(24)}$$

2.3.5 Sarsa

Sarsa 演算法由「當前策略狀態(State)- 當前策略行動(Action)-當前策略獎勵(Reward)-下一回合狀態(s')-下一回合行動(a')」首字母縮寫詞而來，同時也代表其運作方式，由以下公式(25)更新 Q 值。

$$Q(s,a) = Q(s,a) + \alpha(r + \gamma Q(s',a') - Q(s,a)) \text{—(25)}$$

Sarsa 演算法步驟與 Q-Learning 大同小異，差別只在於更新 Q 值時，Sarsa 選擇採用 ε -greedy 所選擇出來之獎勵函數，圖 11 為 Sarsa 之操作流程。



資料來源：用 PYTHON 實作強化學習：使用 TensorFlow 與 OpenAI Gym

圖 11 Sarsa 操作流程

表 2 為 Q-learning 與 Sarsa 之相關內容與差異性彙整。

表 2 Q-learning 與 Sarsa 之相關內容表

	Q-learning	Sarsa
學習策略	Off-policy	On-policy
RL學習方法	Model-free RL(Value-based RL)	Model-free RL(Value-based RL)
產生新動作策略	ε-greedy隨機貪婪策略	ε-greedy隨機貪婪策略
差異點	訓練時較有彈性，前進下一回合選擇以ε-greedy作為選擇依據	前進下一回合只選最大獎勵函數
更新Q值公式	$(1-\alpha)Q_{s,a}+\alpha(r+\gamma max_{\alpha' \epsilon A}Q_{s',a'})$	$Q(s,a)+\alpha(r+\gamma Q(s'a')-Q(s,a))$
策略	策略性挑選(只要訓練走累加獎勵值最大即可)	只走正確之路(最佳行為)
演算法	單步(one step)演算法-只跟當期與下一回合狀態-行動值有關	單步(one step)演算法-只跟當期與下一回合狀態-行動值有關

資料來源：本研究彙整

強化學習中其中一種方法為 Model-free RL，又稱為無模型強化學習，可直接通過環境做訓練，因不用通過模型，因此通用性高，但因環境複雜且有時難以預估，因此學習較困難，執行效率也因需判別多而較低。

Model-free RL 又區分為 Policy-based RL、Value-based RL、混合型 RL。Value-based RL 意思為以訓練出一個價值函數來做為評論者，以評斷策略之好壞，並且利用此函數來改善目前執行狀態，以找到最佳之獎勵總合之長期價值函數。此方法可計算所有累積之獎勵值，且可在每個回合進行評估，待全部價值皆評估出來便可得到策略，稱為確定策略(Deterministic Policy)，意旨挑選最大價值函數；但現實生活中，卻常以隨機策略(Stochastic Policy)來得到更高之獎勵值，如同人之學習，最好選擇不一定會得到最好結果，在學習路程上所累積之經驗值也會不同，所得到之長期獎勵值也不一定比都選擇最好來的低。

2.4 強化學習應用旅行商問題與車輛路徑問題

此章節為彙整本研究所探討過去對於強化學習應用旅行商問題與車輛路徑問題之研究論文，找出使用狀態、行動、獎勵值、問題種類、強化學習方法、優劣及結果，並彙整如表 3 所示。

1. Hu et al. (2020)為研究多重旅行商問題(MTSP)，目的為最大程度地減少所有銷售人員旅行之子行程之總和或最小化最長之子行程。使用多主體學習方法，進一步共享圖神經網路與分部式策略網路來產生小型 TSP 之近似最優解決方案，此學習策略最大值建構出最近似最優解之解決方案。
2. Weijian et al. (2020)使用 Open Graph Gym 環境促進強化學習，來解決圖優化之問題，並結合深度強化學習及圖嵌入來得到結果，以解決較大尺寸之圖型，並同時減少事件次數來找到問題之最佳解決方案。對於不同類型之圖形問題，強化學習算法有不同之表示形式，例如：最小頂點覆蓋率(MVC)和最大剪切率(MAX)，以實現高性能和高質量之計算圖形，並找到最佳之解決方案，但缺點為：若要研究另一種圖形問題，需要修改圖型組件才可支持新之

圖嵌入方法。

3. Delarue et al. (2020)開發基於價值函數之深度強化學習方法來解決容量限制車輛路徑問題，並且與傳統 OR-Tools 相比，求解速度平均差異達到 1.7%，且可用更簡單之神經結構獲得結果。
4. Kool et al. (2019)研究 VRP、TSP(包括傳統 TSP、PCTSP、SPCTSP)、OP(定向問題)，在多節點之情形下，以推動一學習啟發式方法，並得到高度優化與專業化之結果，同時也發現使用確定性貪婪展開所得之展開結果比使用值函數來訓練模型更有效。
5. Kalakanti et al. (2019)以 Q-Learning 與 SWEEP 法來對解決 VRP 做比較，並提出針對 VRP 之強化學習求解器(RL SolVeR Pro)，其研究顯示方能獲得更好或相同水平之結果。缺點為未能很好運用於研究中其中一種時間環境-隨機時間環境上。
6. Patrick et al. (2018)使用強化學習來生成刀具打印路徑，避免使用傳統分支定界或線性編程算法所產生之次優且有缺點之結果，最後結果可以使用較少之列印珠且使用零件上性能也最佳。其缺點為不能保證能夠使用更複雜之結構，而在製作過程中，若發生列印珠斷裂之情形，需花費更長時間，且幾何圖形無法接連續。
7. Nazari et al. (2018)使用強化學習解決車輛路徑問題(VRP)，不需要明確計算矩陣便可求解，通過觀察獎勵信號並遵循可行性規則；及應用策略梯度算法優化參數，即可為從給定分佈中採樣之問題實例找到依連續動作且接近之最優解決方案，且只需要透過一次網路回饋之過程來更新。若使用掩蔽方案時，則每個客戶都必須被拜訪一次。
8. Dai et al. (2017)使用圖嵌入結合強化學習，首先圖嵌入添加節點至模擬環境路網上，再由 Q-Learning 進行評估及訓練，使用貪婪策略步步求解、建構解決辦法，證明研究框架之優化問題可以應用於圖形上，此學習策略為有效之最小頂點覆蓋和旅行商問題算法。
9. Bello et al. (2017)使用策略梯度法來訓練神經網路，以解決組合優化問題，來訓練一遞迴神經網路求解 TSP，無須進行大量工程及啟發式設計仍可得到接近解，但此系統中，主動搜索所需花費時間較長。
10. 胡尚民與沈惠璋(2020)目標為建構最小化路徑之電動車路徑優化問題(EVRP)，其限制包含路徑總時間限制、載重量限制、電池容量限制，並使用策略梯度法訓練模型。分析結果表明，該演算法總體上具有更好尋優性能，能夠提升效率，如總路徑更短、車輛數更少。
11. 鍾玉峰等人(2019)以台灣海洋模式數據，為台灣海域附近行駛船隻構建最佳化路徑，透過 Q-Learning 與 DQN 來進行學習，能更彈性面對環境、提升學習效率，且可利用圖形發現學習結果，確保前進至目的地之行動累積報酬最大值(包括最小化路徑、最節省燃料)。未來可加入如海流流向、流速、風速等因素來擬真更真之環境、規劃不同需求之最佳船型路徑，或者將範圍縮小至特定研究區域，讓路徑能更符合現實。
12. 張震等人(2015)為解決收貨範圍內重疊之區域應由哪輛運輸車運送貨物，並尋求每輛運輸車之平均收取貨物時間之最短路徑，來對運輸車輛路徑優化。以多智能體強化學習方法為基底求解，使用 Q-Learning 及 Boltzmann 策略

選擇車輛行駛之方向，並用 Q-Learning 更新最大累積回報之頻率，因多智能體強化學習會造成穩定性不足，該研究才使用以 Q 學習方法調整最大累積之回報的頻率，來得到最佳路徑。

表 3 強化學習應用旅行商問題與車輛路徑問題

作者	年份	狀態	行動	代理人	獎勵	RL方法	問題種類
Yujiao Hu et al.	2020	每個代理人被分配到需訪問之城市數	代理人必須且只能訪問一次	銷售人員	策略預期最大化	圖嵌入(GNN)、分佈式策略網絡(DisPN)	MTSP(多重車輛路徑問題)
Weijian Zheng et al.	2020	OpenGraphGym	代理隨機或根據策略從節點集中選擇節點	圖形	方案質量之證明	圖形嵌入、OpenGraphGym組件、Q-Learning	圖形優化問題
Wouter Kool et al.	2019	節點數以n=20、50、100來訓練		電腦	NA	確定性貪婪策略	TSP、PCTSP、SPCTSP、VRP、OP
Arthur Delarue et al.	2020	城市數(11、21、51)，而車輛通行能力Q隨都市數量而變化，Q=20、30、40	隨機抽取n個位置，並將隨機選擇其中一個都市作為車輛段，其餘n−1個都市之需求從(1,2，…，9中統一抽樣)	神經網路體系結構	一條使即時成本（路徑長度）和新狀態之值函數之和最小容量之可行路徑更新	策略梯度法	CVRP
K. Kalakanti et al.	2019	1. 客戶環境：C-Type（集群客戶）、R-Type（均勻分佈客戶）、RC-Type（R-Type和C-Type的混合） 2. 問題規模(客戶數)：200、400、600、800、1000		電腦		Q-Learning法	VRP
Steven D. Patrick et al.	2018	基於神經網絡(NN)之隨機點而生成的路徑	根據評論家之打分學習要點和創建填充模式之不同策略	參與者	嘗試在點上實現NN並得到分數	RNN法	刀具打印路徑問題

作者	年份	狀態	行動	代理人	獎勵	RL方法	問題種類
Mohammadreza Nazari et al.	2018	隨機選擇每個節點之需求為(1，…，9)中之一個離散數		電腦		策略梯度法(Actor network、Critic network)	VRP
Hanjun Dai et al.	2017	空間中向量節點數	將節點嵌入μ_v之相應p維節點	電腦	行動v並轉換到新狀態S'後成本函數之變化	S2V-DQN、Q-Learning	TSP
Irwan Bello et al.	2017	分別以TSP20、50和100個節點，生成一個1000個圖的測試集，在$[0，1]^2$中隨機且均勻地繪製點	給定一組輸入點s，學習當前策略所得到之期望旅行長度	神經網路	旅遊長度最小化做為獎勵訊號	策略梯度法	TSP
胡尚民與沈惠璋	2020	節點數設置128、學習率10^{-4}、訓練及大小設置為200000、迭代次數20		神經網路	使用REINFORCE算法估算路徑長度期望，從而減少梯度的方差	策略梯度法	EVRP
張震等人	2015	運輸車輛的目前位置各個目的地是否收到貨物	運輸車輛在單位時間內之行駛方向（向左、向右、向上、向下）	所有直接進行合作的運輸車輛	若車輛沒有經過a地點，表示車輛沒有收到a地點之貨物，給予模型一個正整數，然後將a地點設為已收取貨物之狀態。若車輛已經過a地點或沒有到達任何地點，就給予一個負整數	Q-Learning、Boltzmann策略	最佳路徑
鍾玉峰等人	2019	1. Q-learning:10*10網格環境靜態網格地圖 2. Deep Q-Network:129*129台灣海洋模式建構之網格地圖建立		電腦	船隻最佳航行路徑(Q-learning)-負報酬以更新Q值，並用deep Q-Network更新目標定義(Double DQN)	Q-learning法、DQN法	最佳路徑

2.5 綜合評析

本章節將文獻探討所彙整之結果分析如下。

1. 車輛路徑問題

目的為車輛組合最佳化問題，具有限制條件參數，為多個 TSP 之組合，使目標函數可得到最大化求解。依照問題特性又可延伸成各種車輛路徑問題，例如容量限制車輛路徑問題。基本概念為一場站與多個需求點所構成之路線問題，當全部點皆通過時即回到場站，此為一運輸活動。

2. 強化學習

強化學習為機器學習一種，介於監督式與非監督式學習間。包含四個元素-觀察、環境、行動、獎勵，透過不斷互動，使代理人在環境不同狀態下產生不同行動，透過行動得出相關獎勵值，最終目的為求得所有行動所產生最大化獎勵值，將獎勵行為分為好與壞，依此給予正回饋或負回饋，最後藉由採取行動所帶來之關聯反覆訓練，使代理人行為越加正確並得到最好之訓練行為。

在強化學習中，最重要觀念為馬可夫決策過程，為所有強化學習之基礎方針，包含狀態、行動、轉移機率、獎勵機率及折扣因子。並透過策略函數與價值函數解決最佳化問題。

以如何優化決策實現最佳結果為主軸，延伸至不同領域產生不同利益結果，證實強化學習為靈活運用之中心思想。演化至今，使用強化學習之應用平台也愈加廣泛，舉凡 OpenAI Gym and Universe、RoboSchool、DeepMind Lab、Project Malmo、ViZDoom、Amazon SageMake 等。

強化學習相關應用法有數種，例如貝爾曼方程式、Q-Learninig、策略梯度、DQN 法，根據模式不同會產生不同演算法與其優缺點，沒有哪種方法是最好的，不同應用會有不同合適模式。

強化學習應用於 TSP 與 VRP，不同論文有不同看法，其共同觀念皆為-強化學習可以將複雜求解過程減化，提高準確率並減少求解時間，再將自身要求輸入模式中得到最佳解。

綜合而言，本研究與上述參考文獻最大不同點為-使用強化學習結合車輛路徑問題來進行校園無人車餐飲配送活動，因此，本研究具有其研究價值，擬提供未來各校園能夠實施進入校園中，減免運送外食進校園之不便。

第三章、研究方法

3.1 問題特性

本研究根據組合最佳化問題配置模型，主要為解決用餐時刻教師與學生無空閒時間且有購買餐點需求，並提供無人車配送服務，以此為目的建構強化學習之模型。本研究目標為無人車在其配送過程中顧客需求不超過車輛容量且達到流量守衡之限制情況下，其行駛距離以及相關成本使用，能最大限度降低其運輸成本，並使無人車行程時間與車輛載荷能達到最小變化，期望能透過此反覆採樣配送路線找出最優解決方案。

本研究研擬出顧客資訊端、店家資訊端以及後端平台整合資訊端等物流配送系統，此系統可於手機 APP 供顧客選購較熱門餐點與其相關資訊，並即時追蹤無人車位置與配送至智慧櫃時間；店家可根據前一天訂單資訊，提供相對應餐點數量於訂購時間放置無人車進行配送；而無人車動態位置則將由後端平台整合資訊端進行處理並規劃路線，進而得知各大樓使用頻率及智慧櫃之周轉率，以作為日後增減無人車及智慧櫃數量之參考。因此，此資訊端規劃主要為使顧客可享受更完善服務，並完成如圖 12 無人車配送路線。本研究範圍主要以車輛路徑規劃為主，由路線角度出發做整體物流配送構想，且情境假設資訊端已完善情況下。

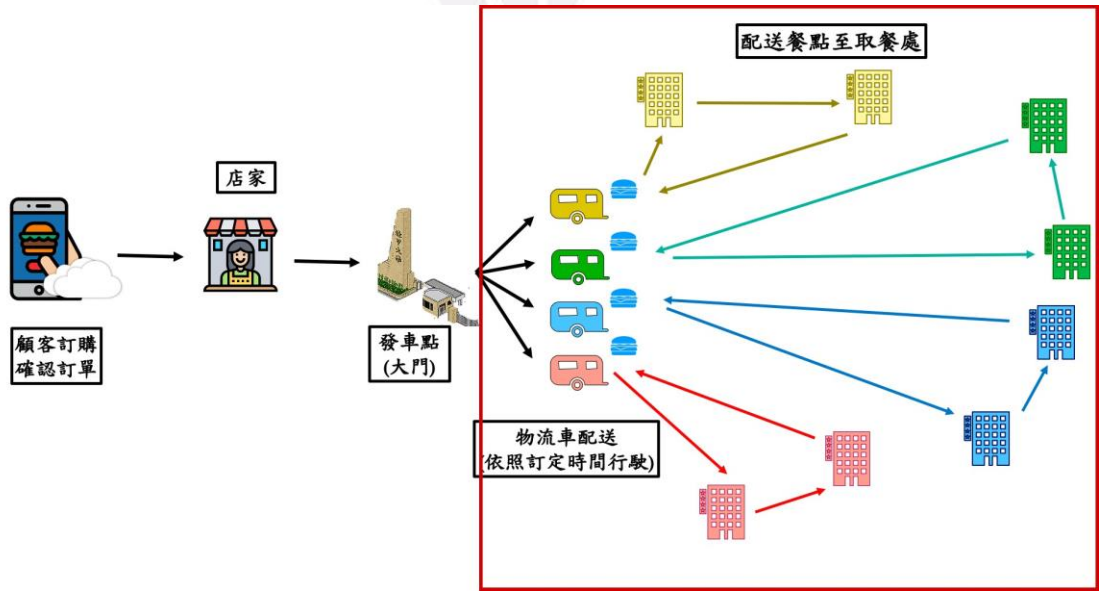


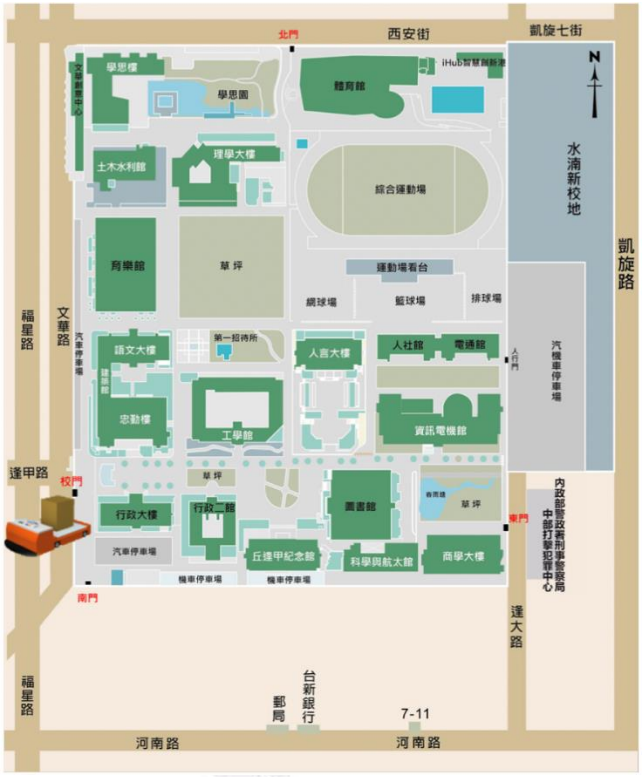
圖 12　無人車配送路線

3.2 校園無人車派遣之強化學習架構

本研究以無人車作為操作代理人，運用策略迭代演算法解決 CVRP 模型問題，並根據其模擬環境透過神經網路得出相關行動結果，其中無人車配送之模擬環境將擬定為 N 棟教學大樓。圖 13 為模型中，無人車配送餐點範圍，即為逢甲大學校門口與各教學大樓之校園位置圖，無人車將會於各個教學大樓之指定位置等候客戶前來領取餐點。

本研究以逢甲大學校園為主要路線規劃場地，將載有一定餐飲數量之無人車由校門口送至各個教學大樓之路徑，作為 CVRP 之組合最佳化問題，使用基於組

合動作價值之強化學習之策略迭代算法，來建構 CVRP 模型。並以指針網絡為基礎架構，並找出近乎最優解決方案。



資料來源： 逢甲大學官網(2020)

圖 13 逢甲大學校園平面圖

本研究模型擬以校門口為一節點 i，無人車從校門口 i 至教學大樓 j 之距離，以**Δij**表示，其中 i、j> 0，且 i、j 皆與顧客需求 di 相關。於模型中，本研究擬先假設提供無人車至各個教學大樓之路線數量為無限，且逢甲校園內之教學大樓 j 擬以座標方式進行定位，此定位方式將便於無人車在模擬或是實際上路時，可準確得知配送方向。而模型策略 n 進行方式為當無人車以一定容量從校門口 i 裝載便當後，即可隨機選取一條路線作為前往指定教學大樓 j 進行配送。此隨機性策略$\pi(s)$擬定義為$\pi(a \mid s) = P[A_t = a \mid S_t = s]$，指「行動 a」在「所有可能狀態 s」中之機率分佈。且於形式上，策略$\pi(s)$=arg max$_a$Q(s,a)，表示每個「狀態 s」中具有 Q 之「行動 a」(Lapan, 2019)。所有餐點皆配送完畢後，無人車擬尋找路線返回校門口 i 進行待機。上述操作流程將如圖 14 所示，此流程圖為本研究預設路線，路線優先順序擬根據模擬結果而持續變更其行經路線，直至最佳化路徑選擇。

本研究於模型中所預設公式擬參考何柱等人研究，公式(26)表示此模型中之目標式，即最小化行駛距離；公式(27)表示每個節點皆有一輛車配送顧客所需之餐點；公式(28)表示每輛車可配送 1 條以上路線，直至將指定地點之餐點配送完畢；公式(29)表示此神經網路中流量守恆條件；公式(30)表示每輛車所需配送之餐點不超過其容量限制；公式(31)則為防止無人車於此路線配送流程中，出現孤立子環之約束條件(何柱等人，2019)。

$$\min \sum_{i=1}^{M} \sum_{j=1}^{N} \Delta ij x_{ijk} \text{—(26)}$$

$$s.t\sum_{i=1,i\neq j}^{M}\sum_{k=1}^{V}x_{ijk}=1,\forall j\in\{1,2,\ldots.,N\}\text{—(27)}$$

$$\sum_{j=1}^{N}x_{ijk}\leq 1,\forall i\in\{1,2,\ldots.,M\},\forall k\in\{1,2,\ldots.,V\}\text{—(28)}$$

$$\sum_{i=1,i\neq j}^{M}x_{ijk}=\sum_{i=1,i\neq j}^{M}x_{jik},\quad \forall j\in\{1,2,\ldots.,N\},\forall k\in\{1,2,\ldots.,V\}\text{—(29)}$$

$$\sum_{i=1}^{M}d_i\,x_{ijk}\leq Q\text{—(30)}$$

$$\sum_{k=1}^{V}\sum_{i\in S}\sum_{j\in S,i\neq j}x_{ijk}\leq|S|-1,\forall S\subseteq\{1,2,\ldots.,M\}\text{—(31)}$$

其中，k 為此模型派遣之無人車

x_{ij}為決策變數，表示無人車 k 是否有從 i 點行駛至 j 點

此實驗策略π(s)結果也將根據圖 14 無人車配送餐點至各教學大樓選擇之最短路徑進行獎勵制度，模型設定之獎勵區間如圖 15 將提供正負回饋，並進行儲存。當無人車從校門口至指定教學大樓配送餐點路線為正確道路，將給予 1 分正回饋值；稍微偏離主要道路則給予 0.5 分正回饋值；而無人車完全偏離系統預設道路就會給予 1 分負回饋值。

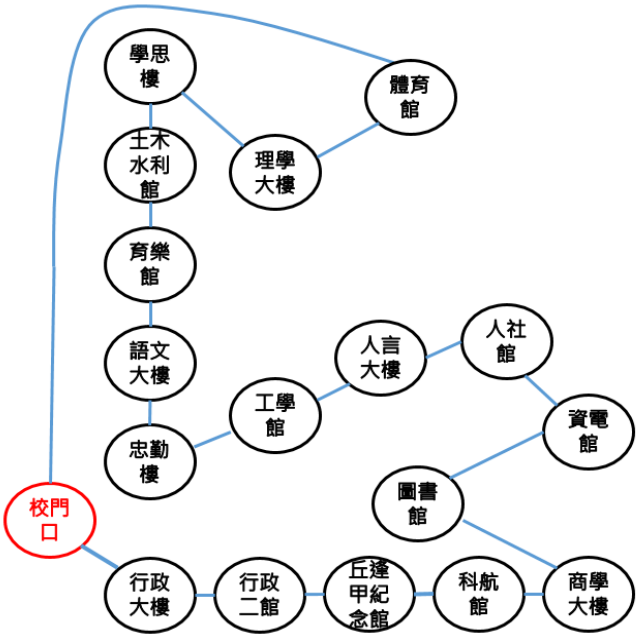


圖 14　初擬路線流程圖

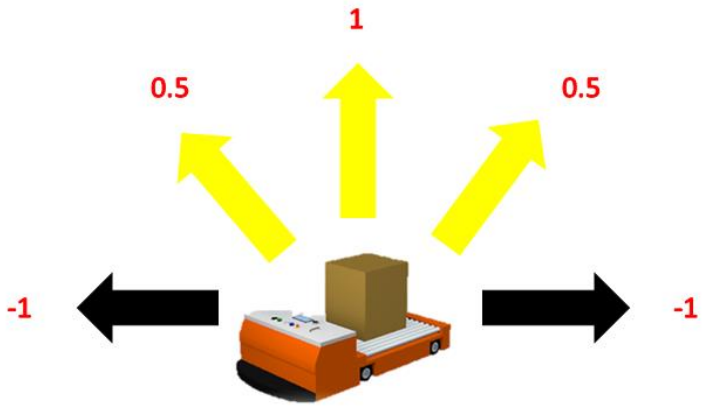


圖 15　初擬獎勵區間

此獎勵方式稱為「信賴域策略最佳化」(Trust Region Policy Optimization, TRPO)，也就是給定一項約束值，來藉此確保該模型代理人處在正確道路區域內。並持續加入約束來改進該策略，使新舊策略間之 KL 散度得以小於指定常數 δ。最後將此模型使用優勢目標值$A_{\theta_{old}}$將 Q 值$Q_{\theta_{old}}$進行替代，最終可得出公式(32)與(33)目標函數。因此，本研究擬獎勵最大化方式，來更新模型參數 (Ravichandiran, 2019)。

$$\text{maximize}_{\theta} \quad E_s\pi_{\theta_{old}}\,,a\pi_{\theta_{old}}\left[\frac{\pi_\theta(a|s)}{\pi_{\theta_{old}}(a|s)}A_{\theta_{old}}(s,a)\right] \text{—(32)}$$

$$\text{subject to} \quad E_s\pi_{\theta_{old}}\left[DKL\left(\pi_{\theta_{old}}(\cdot|\,s)\parallel\pi_{\theta_{old}}(\cdot|\,s)\right)\right]\le\delta\text{—(33)}$$

本研究將模型不斷反覆進行訓練如圖 16 架構圖所示，期望最終決定策略路線能夠根據此「信賴域策略最佳化」之獎勵模式，找出一條能夠使無人車在容量限制之情況下，得以最小化車輛行駛距離。

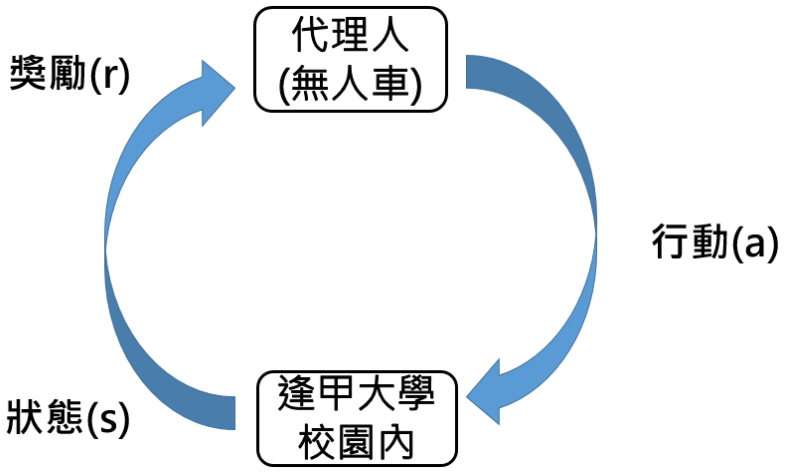


圖 16　研究架構圖

本研究於建構模式部分，首先根據所設置參數與環境限制建構模型，並使用策略迭代測試環境，不斷地進行測試形成迴圈，找出車輛路徑最佳解，最後根據分析結果研擬出相關路線規劃策略，如圖 17 所示。

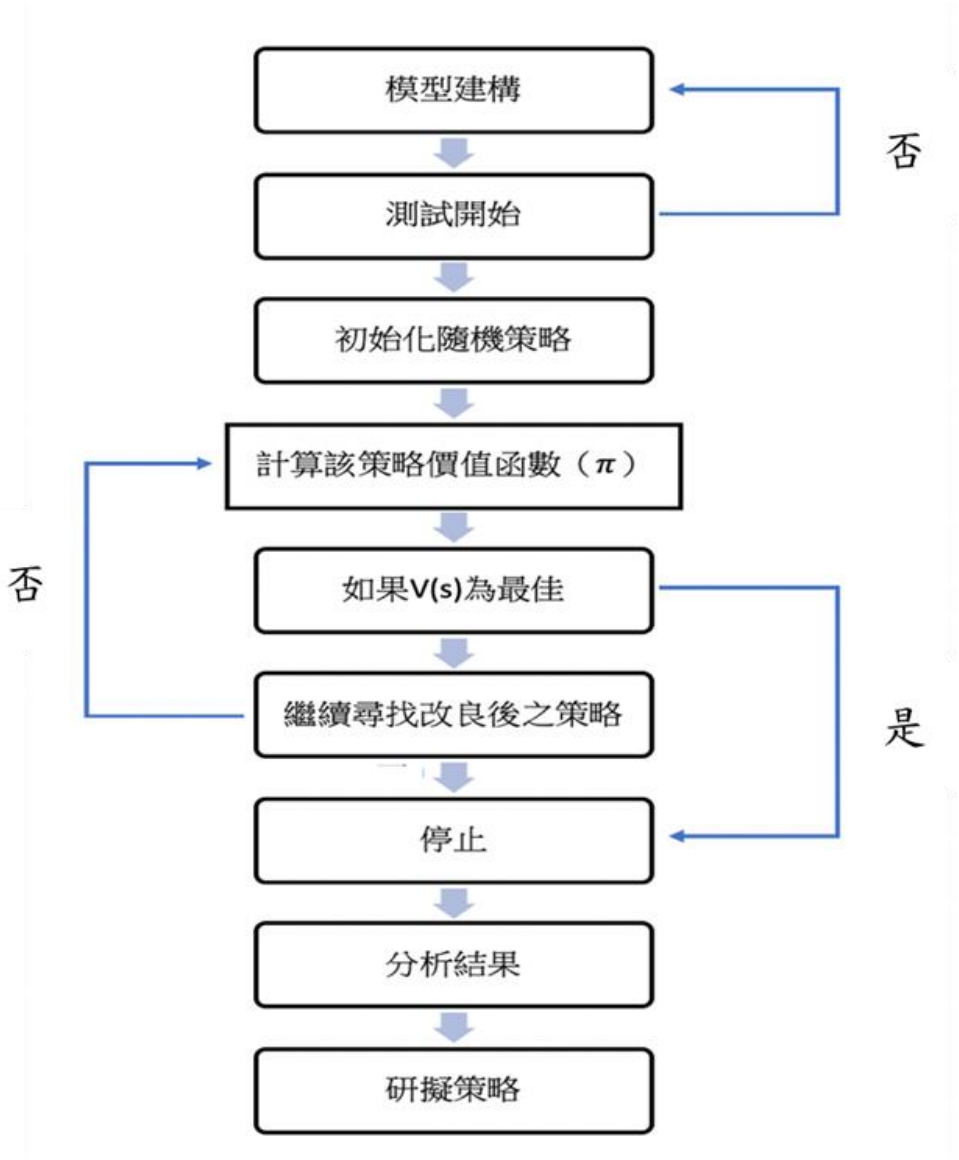


圖 17　車輛路徑策略分析

本研究透過使用策略迭代法撰寫 python 程式基礎模型，並以貪婪模式進行測試。本研究在初期分析階段，透過調整不同函數與參數值進行分析。除此之外，為測試該模型實驗結果與傳統求解法之求解差異，擬將最佳解與 Lingo 進行結果比較，得出強化學習應用於校園無人車車輛路徑問題之優劣。

第四章、結果分析與討論

4.1 基本測試與分析

4.1.1 測試說明

本研究以 Rintarooo(2021)之研究為基礎建構校園無人車之 AI 強化學習餐飲派遣分析，並以逢甲大學各大樓座標為需求點模擬需求點參考，並使用 NEO(network and emerging optimization)之公開文件檔進行大樓位置定位及容量轉換，共設置 16 個需求點及一個發車點(校門口)，並將發車點設為編號 1，需求點 1 為編號 2，其餘依序往後編號，如圖 18 所示，表 4 為各需求點之間距離。

本研究將一個便當設置 500 公克，車輛容量設置 25 公斤，一趟共可載 50 個便當數。本研究使用電腦效能 Intel(R)Core(TM)i7-4770CPU@3.40GHz 3.40GHz 進行訓練。鑑於電腦性能問題，最高訓練批量(Batch Size)只能使用批量大小為 256，訓練迴圈(Epoch)為 20 迴圈做訓練與分析模型，訓練步數設置 10000 步，目的在求解出最小化模擬距離及測試時間，以作為日後校園路徑派遣之訓練模型。本研究將訓練結果與 Lingo 做比較，做為最佳解判斷標準。表 5 為 16 個需求點隨機設置需求數，並以公斤單位作為需求參考進行測試分析。

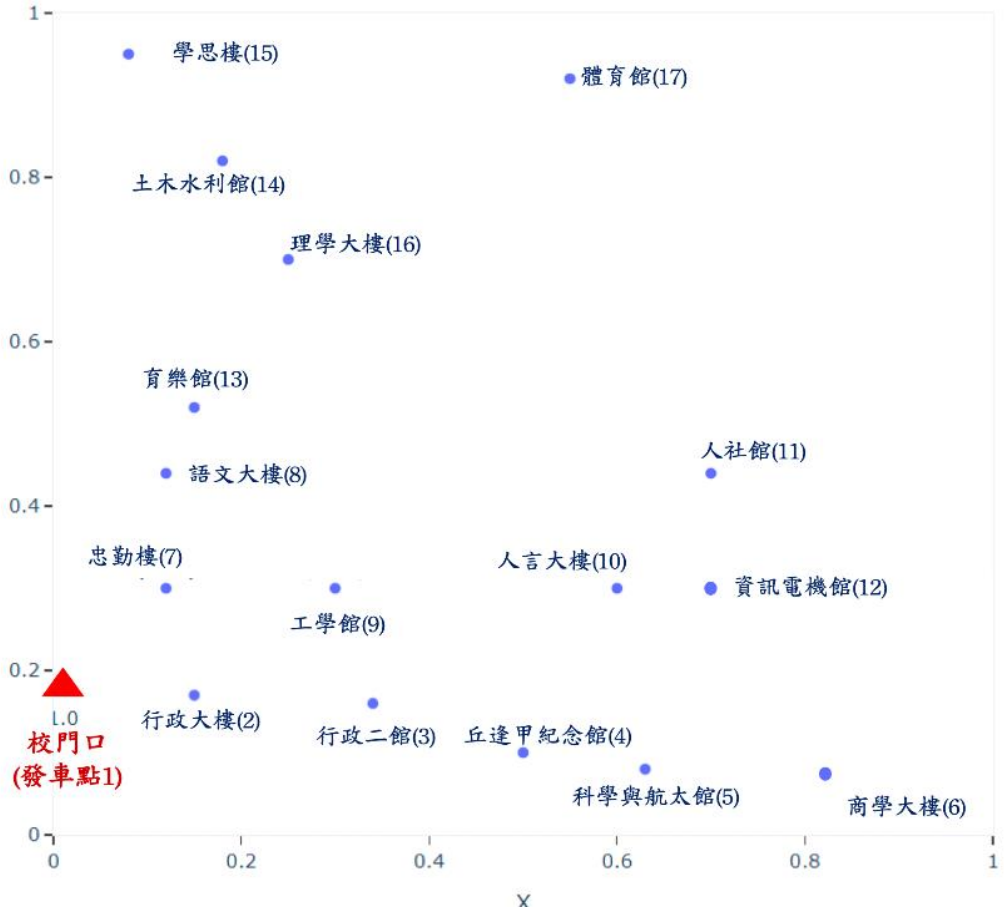


圖 18 模擬需求點位置圖

表 4 逢甲大學各大樓間距離矩陣

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	43.85	107.85	187.82	231.19	290.17	50.69	123.81	159.7	268.84	334.21	286.01	155.9	248.14	316.86	309.31	457.88
2	43.85	0	64.03	134	183.52	247.26	43.64	138.94	127.36	235.51	301.74	255.42	183.88	273.62	337.4	290.35	415.66
3	107.85	64.03	0	69.36	118.2	181.34	102.02	217.14	68.1	172.4	240.01	191.16	207.21	341.19	359.18	271.33	358.77
4	187.82	134	69.36	0	56.78	125.5	179.1	261.49	88.67	136.6	191.29	144.42	282.19	400.25	427.01	345.9	341.26
5	231.19	183.52	118.2	56.78	0	73.51	227.85	305.48	140.81	150.5	205.29	149.78	356.03	444.14	511.01	388.83	355.12
6	290.17	247.26	181.34	125.5	73.51	0	301.4	366.28	206.84	193.04	169.04	113.8	414.63	503.14	551.79	447.2	415.74
7	50.69	43.64	102.02	179.1	227.85	301.4	0	155.53	128.8	256.22	313.86	274.33	174.93	375.19	349.69	267.75	407.38
8	123.81	138.94	217.14	261.49	305.48	366.28	155.53	0	235.98	281.81	270.19	310.08	26.56	207.99	221.25	147.59	310.7
9	159.7	127.36	68.1	88.67	140.81	206.84	128.8	235.98	0	154.89	209.63	174.73	130.35	356.58	287.59	255	336.51
10	268.84	235.51	172.4	136.6	150.5	193.04	256.22	281.81	154.89	0	60.65	62.43	270.57	408.91	407.71	330.88	283.69
11	334.21	301.74	240.01	191.29	205.29	169.04	313.86	270.19	209.63	60.65	0	41.41	256.14	391.52	390.32	306.79	270.8
12	286.01	255.42	191.16	144.42	149.78	113.8	274.33	310.08	174.73	62.43	41.41	0	296.91	389.88	452.06	361.58	361.68
13	155.9	183.88	207.21	282.19	356.03	414.63	174.93	26.56	130.35	270.57	256.14	296.91	0	199.03	210.99	138.95	302.27
14	248.14	273.62	341.19	400.25	444.14	503.14	375.19	207.99	356.58	408.91	391.52	389.88	199.03	0	22.97	78.56	166.88
15	316.86	337.4	359.18	427.01	511.01	551.79	349.69	221.25	287.59	407.71	390.32	452.06	210.99	22.97	0	90.5	170.78
16	309.31	290.35	271.33	345.9	388.83	447.2	267.75	147.59	255	330.88	306.79	361.58	138.95	78.56	90.5	0	165.2
17	457.88	415.66	358.77	341.26	355.12	415.74	407.38	310.7	336.51	283.69	270.8	361.68	302.27	166.88	170.78	165.2	0

表 5 各需求點需求量表

編號	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
需求	x	2	1	21	12	10	6	5	1	13	2	4	8	7	20	3	10

4.1.2 測試結果

1. 強化學習測試結果

本研究透過強化學習模型分別測試批量大小 64、128 及 256，求解出最小化距離及測試時間，並比較其結果，以選出最佳訓練模型。表 6 為各批量間測試結果，測試時間方面，批量 64 共花費 0.173 秒，相較其他兩者而言所需時間較少;模擬距離方面，本研究考慮到障礙物問題，將模擬距離換算成實際距離進行比較，如圖 19 所示，不同路線代表不同車輛配送範圍，箭頭方向為行駛路線，各無人車最終皆回到校門口(發車點 1)。由結果可得知批量 256 所需距離較短;車輛派遣方面，三者所需車輛數相同，皆需派遣 6 臺進行配送。

綜上所述，批量 256 花費時間相較批量 64 多，且所需里程比批量 64 少，推測批量越大所花費測試時間相對較多，但行駛情形較佳，推估將批量放大至批量 512，求解值會更接近甚至優於最佳解。因鑑於資源有限，故此暫不討論以批量 512、訓練迴圈(Epoch)為 20 迴圈、訓練步數設置 10000 步作為模型測試。

表 6 各批量間之測試結果

批量大小	64	128	256
平均測試時間(s)	0.173	0.28	0.32
平均派遣距離(m)	3578.69	3534.84	3519.91
派遣車輛數	6		

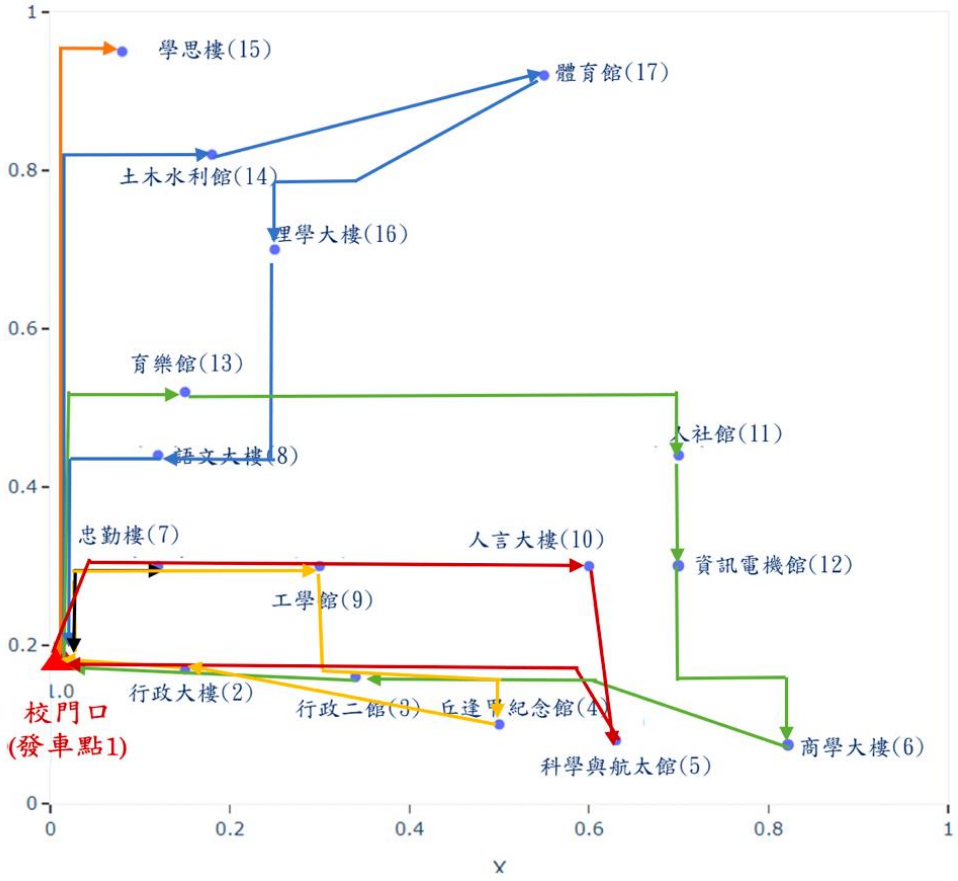


圖 19 測試結果路線圖

2. Lingo

本研究中使用商用套裝軟體 LINGO 執行傳統求解全部車輛最小化距離，依照實際測量之逢甲大學各大樓距離矩陣進行求解測試。共花費 8 分 13 秒進行求解，所得出共需 6 臺無人送餐車、總累積里程為 3463.57 公尺，所花費測試步數為 267702 步。除此之外，經過測試發現不同需求情境也會有不同測試時間。

4.1.3 敏感度分析

本小節將上述基礎測試結果做進一步參數分析與探討。本研究分別針對各需求點之需求改變、需求點數變動及容量限制三種參數進行測試分析，以增減 10%以及 20%為例，並與原始測試時間、車輛數、行駛距離相互比較，並分別論述其變動情形。

1. 需求量改變

在需求部分，分別在各需求點增減 10%與 20%需求量。在+10%中，平均時間 0.54 秒，最快時間 0.399 秒，車輛數為 6 輛；在+20%中，平均時間 0.581 秒，最快時間 0.504 秒，車輛數為 7 輛；在-10%中，平均時間 0.435 秒，最快時間 0.504，車輛數為 5 輛；在-20%中，平均時間 0.441 秒，最快時間 0.372 秒，車輛數為 5 輛。由表 7 可看出，隨著需求量增加，所需配備車輛數也增加。在平均距離部分，隨著需求量增加，而距離也同時增加。且不同需求量中，求解時間隨著各需求點之需求量增加而增加，但求解時間皆在 1 秒內，如表 7 所示。

表 7 需求量改變敏感度分析

需求量改變	-20%	-10%	0	+10%	+20%
平均時間	0.441	0.435	0.48	0.54	0.581
平均距離	3208.4	3418.42	3519.91	3768.61	4070.35
車輛數	5	5	6	6	7

2. 需求點改變

在需求點改變部分，分別在增減 10%與 20%需求點數目。在+10%中，平均時間 0.6002 秒，最快時間 0.471 秒，車輛數為 6 輛；在+20%中，平均時間 0.824 秒，最快時間 0.692 秒，車輛數為 6 輛；在-10%中，平均時間 0.519 秒，最快時間 0.437，車輛數為 5 輛；在-20%中，平均時間 0.381 秒，最快時間 0.253 秒，車輛數為 5 輛。由表 8 可看出，距離部分隨著需求點增加而增加，且測試時間隨著需求點數增加而隨之提升，但仍在一秒內完成路線規劃，如表 8 所示。

表 8 需求點改變敏感度分析

需求點改變	-20%	-10%	0	+10%	+20%
平均時間	0.381	0.519	0.48	0.6002	0.824
平均距離	2530.12	3123.95	3519.91	3881.58	4557.12
車輛數	5	5	6	6	6

3. 車容量數改變

在車容量改變部分，分別在各車輛增減 10%與 20%車容量大小。在+10%中，平均時間 0.433 秒，最快時間 0.381 秒，車輛數為 5 輛；在+20%中，平均時間 0.544 秒，最快時間 0.394 秒，車輛數為 4 輛；在-10%中，平均時間 0.503 秒，最快時間 0.401，車輛數為 6 輛；在-20%中，平均時間 0.426 秒，最快時間 0.411 秒，車輛數為 7 輛。由表 9 可看出，距離部分隨著車容量增加而減少、車輛數也同樣減少，反之，車容量減少，車輛數與距離也隨之增加。測試時間皆在一秒內，如表 9 所示。

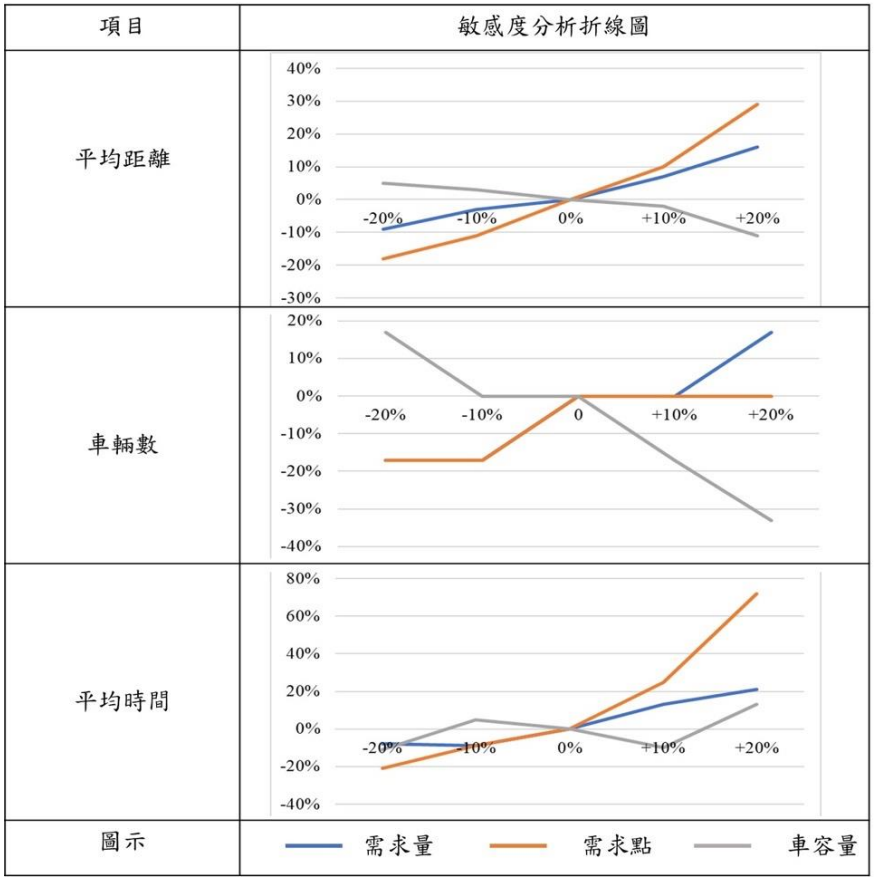
表 9 車容量改變敏感度分析

車容量改變	-20%	-10%	0	+10%	+20%
平均時間	0.426	0.503	0.48	0.433	0.544
平均距離	3692.75	3611.67	3519.91	3432.88	3127.76
車輛數	7	6	6	5	4

4. 敏感度綜合分析

根據上述幾點可知，在需求點改變、車容量改變與需求量改變這三種參數中，首先，平均距離影響部分，需求點改變>車容量改變>需求量改變；第二，在車輛數影響部分，車容量改變>需求點改變>需求量改變；最後，在平均時間部分需求點改變>車容量改變>需求量改變。綜上結果，三者中以需求量改變為較不明顯之參數，而需求點改變則對平均距離與測試時間有較大之影響，車容量改變對車輛數有較大影響，如表 10 所示。

表 10 敏感度綜合分析折線圖



4.1.4 綜合討論

本研究分別對批量、訓練步數等參數進行強化學習績效分析。得知強化學習在批量愈大，測試時間會與批量大小成正比，但並不會超過 1 秒鐘，可快速得知結果。派遣距離則與批量大小成反比，因此本研究以批量 256，與不同規模校園所需之需求點數去做進一步分析與討論對策。

根據強化學習與 Lingo 比較結果，以對於車輛路徑問題之強化學習與 Lingo 進行相關綜合探討與分析。在測試時間方面，由上述結果可知，強化學習法需要較多事先訓練時間，但實際真正求解時間與傳統求解法相比，求解速度節省數十倍以上。在實際行走距離方面，透過強化學習所得出之規劃路徑進行實際距離測量，可發現 Lingo 所規劃路徑與強化學習最大批量所行駛距離差異不大，約差 56 公尺。因此，未來可用較大批量訓練模型，方可得到最好結果。在車輛數派遣方面，強化學習與 Lingo 結果中，車輛數派遣為相等結果。綜合上述結果可知，未來運用強化學習去做餐飲派遣可在整體效益上占較大優勢。

在敏感度分析上，本研究透過對需求點、車容量與需求量相關參數分別以比例去做改變，並對平均距離、平均時間、派遣車輛數進行分析，可知變動各需求點之需求量對派遣影響較小，車容量與需求點之改變對餐飲配送影響較大。

4.2 情境分析

本研究同時針對相關參數及策略進行進一步的情境分析，探討不同規模校園內餐飲配送之車輛派遣分析。此情境分析將區分為基本參數情境設計與綜合情境設計進行訓練與分析，對此提出相關分析結果。

4.2.1 情境參數說明

1. 需求點

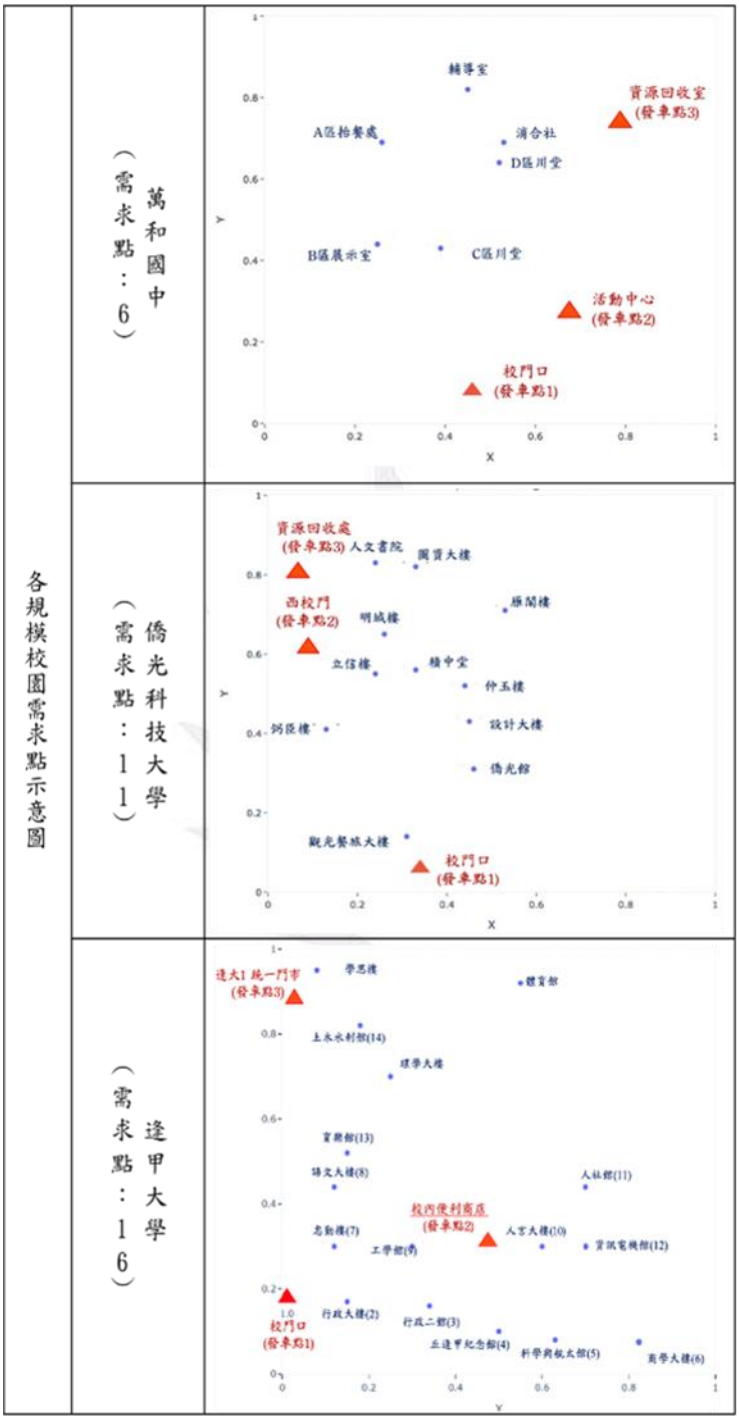
此情境參數探討不同需求點數及位置之路徑規劃問題，透過訓練不同規模大小進行模式求解與分析。本研究依據實際校園建築物棟數－台中市南屯區萬和國中、台中市西屯區僑光科技大學及台中市西屯區逢甲大學，分別對應將測試規模分為需求點 6、11、16 三種需求點數進行訓練。

2. 發車點

此情境參數分別在 6、11、16 不同規模校園內設置 1~3 個發車點，藉以分析不同規模中應設置多少發車點數為佳，並將車輛平均在各發車點進行發車。在萬和國中校園共設置 3 個發車點，分別位於校門口(發車點 1)、活動中心(發車點 2)及資源回收室(發車點 3)。在僑光科技大學亦設置 3 個發車點，分別為校門口(發車點 1)、西校門(發車點 2)及資源回收處(發車點 3)。在逢甲大學同樣設置 3 個發車點，分別為校門口(發車點 1)、校內便利商店(發車點 2)及逢大 1 統一門市(發車點 3)。

將 1、2 點參數設計之虛擬環境示意圖彙整於表 11。

表 11　各規模校園需求點示意圖



3. 車容量

本研究設計車載容量為 25 公斤與 50 公斤兩種類型去做派遣分析，並設置冷食與熱食重量皆為一份 550 公克。無人車載容量 25 公斤，即最多可裝載 45 份。無人車載容量 50 公斤，則可裝載 90 份。透過不同車容量進行不同規模校園之無人車車容量大小分析依據。

4. 需求量

表 12 擬定無人車於需求量不同環境下之情境問題，擬定三種情境進行分

析。首先為特殊假期與情形，例如寒暑假與疫情期間，此時較少學生會進入校園，情境假設每需求點平均需求將訂為 5 份、15 份兩種情況，分別對應疫情與寒暑假期間。其次為多需求點需求量高，少部分需求點需求量低模擬情形，此情形發生於校園內同時有一般教學大樓與非相關作業人員無法進入之辦公類型大樓，並設置 3/4 棟屬於一般教學大樓，為需求量較高之需求點，其每需求點需求量高達 35 份以上；剩餘 1/4 屬於辦公類型大樓，每需求點需求量設置為 10 份以下。需求量情境設置最後為有營隊活動、學生皆位於同一需求點之情境，屬於單一需求點需求量特別高，剩餘需求點需求量低，設置單一需求點所需需求量超過一台無人車可裝載之限制，其餘需求點之需求量設置為平均且低需求之情形，以凸顯需求點差距大情境。表 12 為上述三者需求量參數假設說明表，其中需求量為冷食與熱食加總需求量數。

表 12 各情境之需求點配置與需求量之參數假設

情境	需求點配置	需求量
特殊假期、情形	每需求點平均需求	5 份、15 份
一般教學大樓與辦公類型大樓	多需求點需求高(一般教學大樓)與部分需求點需求低(辦公類型大樓)	1/4 需求量低為 10 份以下、 3/4 需求量高為 35 份以上
營隊活動	單一需求點需求量高、 其餘需求點需求量低	一個需求點超過 45 份， 其餘需求點 5 份～10 份

5. 無人車類型

此情境參數探討不同營運模式餐飲無人車派遣情境，透過單一溫層無人車與多溫共配無人車二者設置，以了解各環境內不同無人車類型派遣概況。無人車容量設置為車載容量 25 公斤，單一溫層車種將物流箱溫度控制於鮮食品(恆溫 18 度)與冷藏品(0 度~7 度)分開進行配送，分別對應餐飲與飲料需求；多溫共配車種物流箱同時設有兩種溫層格，分別為 18 度與 0 度~7 度，可同時配送餐飲與飲料需求。

6. 派遣車輛數

此參數為該校所能派遣之車輛數，藉以分析不同校園規模，應配置多少台車輛數為佳。

4.2.2 分析測試

1. 基本情境 1 與基本情境 2 測試說明

圖 20 為基本情境 1 設計流程圖，使用不同校園規模、車容量與需求量三種情境參數所設計之八種情境分別進行規劃與分析，且基礎設置餐飲克數 1 份 550 公克為例。在基本情境 1 中，以三種場域需求點做為基底，分別對無人車載 25 公斤與 50 公斤進行派遣分析。在需求量部分沿用上述四種情形：車載 25 公斤中，各需求點份數總量為平均 5 份、15 份、多需求點需求高與單一需求點需求高此四種參數情境做基礎模擬分析。車載 50 公斤中，各需求點份數總量為平均 10 份、30 份、多需求點需求高與單一需求點需求高此四種參數情境做基礎模擬分析。

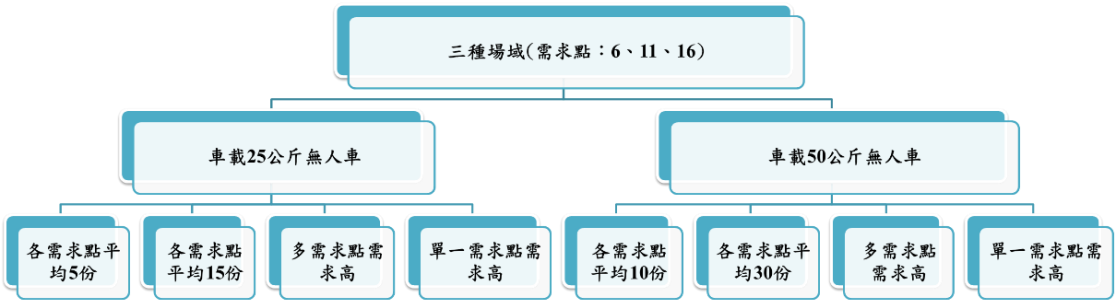


圖 20 基本情境 1 說明

圖 21 為基本情境 2 情境參數設置，參數包含不同校園規模、無人車類型與發車點。在基礎設置部分，各需求點之需求量設置冷食與熱食總和為 15 份、車輛數配置 6 台進行情境設置。除此之外，分別於各校園中配置 1、2、3 個發車點，將無人車平均設置於各發車點以進行派遣分析。

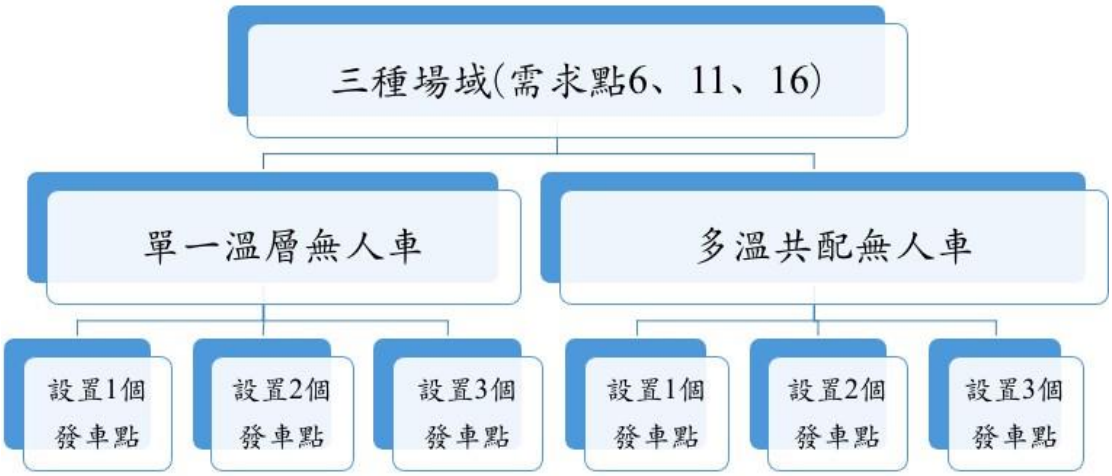


圖 21 基本情境 2 說明

1. 基本情境 1 分析結果

以下為基本情境 1 分析結果，根據不同規模校園，分為車容量 25 公斤與 50 公斤無人車，分別依照四種不同需求量去做情境比較。

（1） 車容量 25 公斤-各需求點平均 5 份

在各需求點皆為 5 份之情形下，容量可承載 25 公斤之無人車在萬和國中需要車輛數 1 台、平均測試時間為 0.223 秒、平均距離 265.02 公尺。在僑光科技大學需要車輛數 2 台、平均測試時間 0.311 秒、平均距離 1590.78 公尺。在逢甲大學需要車輛數 2 台、平均測試時間 0.462 秒、平均距離 2136.74

公尺，如表 13 所示。

表 13 車容量 25 公斤-各需求點平均 5 份

各需求點平均5份			
需求點	萬和國中(需求點：6)	僑光科技大學(需求點：11)	逢甲大學(需求點：16)
平均距離	265.02	1590.78	2136.74
平均時間	0.233	0.311	0.462
車輛數	1台	2台	2台

（2） 車容量 25 公斤-各需求點平均 15 份

在各需求點皆為 15 份之情形下，容量可承載 25 公斤之無人車在萬和國中需要車輛數 2 台、平均測試時間為 0.209 秒、平均距離 354.09 公尺。在僑光科技大學需要車輛數 4 台、平均測試時間 0.303 秒、平均距離 2596.38 公尺。在逢甲大學需要車輛數 6 台、平均測試時間 0.453 秒、平均距離 3531.03 公尺，如表 14 所示。

表 14 車容量 25 公斤-各需求點平均 15 份

各需求點平均15份			
需求點	萬和國中(需求點：6)	僑光科技大學(需求點：11)	逢甲大學(需求點：16)
平均距離	354.09	2596.38	3531.03
平均時間	0.209	0.303	0.453
車輛數	2台	4台	6台

（3） 車容量 25 公斤-多需求點需求量高

在需求點多且需求量高之情形下，容量可承載 25 公斤之無人車在萬和國中需要車輛數 5 台、平均測試時間為 0.19 秒、平均距離 836.74 公尺。在僑光科技大學需要車輛數 9 台、平均測試時間 0.32 秒、平均距離 4682.82 公尺。在逢甲大學需要車輛數 14 台、平均測試時間 0.559 秒、平均距離 6832.95 公尺，如表 15 所示。

表 15 車容量 25 公斤-多需求點需求量高

多需求點需求量高			
需求點	萬和國中(需求點：6)	僑光科技大學(需求點：11)	逢甲大學(需求點：16)
平均距離	836.74	4682.82	6832.95
平均時間	0.190	0.320	0.559
車輛數	5台	9台	14台

（4） 車容量 25 公斤-單一需求點需求量高

在單一需求點需求量高之情形下，容量可承載 25 公斤之無人車在萬和國中需要車輛數 2 台、平均測試時間為 0.216 秒、平均距離 448.18 公尺。在僑光科技大學需要車輛數 3 台、平均測試時間 0.368 秒、平均距離 1333.03 公尺。在逢甲大學需要車輛數 3 台、平均測試時間 0.652 秒、平均距離 2217.64 公尺，如表 16 所示。

表 16 車容量 25 公斤-單一需求點需求量高

單一需求點需求量高			
需求點	萬和國中(需求點：6)	僑光科技大學(需求點：11)	逢甲大學(需求點：16)
平均距離	448.18	1333.03	2217.64
平均時間	0.216	0.368	0.652
車輛數	2台	3台	3台

（5） 車容量 50 公斤-各需求點平均 10 份

在各需求點平均 10 份之情形下，容量可承載 50 公斤之無人車在萬和國中需要車輛數 1 台、平均測試時間為 0.061 秒、平均距離 277.34 公尺。在僑光科技大學需要車輛數 2 台、平均測試時間 0.093 秒、平均距離 1993.89 公尺。在逢甲大學需要車輛數 2 台、平均測試時間 0.08 秒、平均距離 2070.39 公尺，如表 17 所示。

表 17 車容量 50 公斤-各需求點平均 10 份

各需求點平均10份			
需求點	萬和國中(需求點：6)	僑光科大學(需求點：11)	逢甲大學(需求點：16)
平均距離	277.34	1993.89	2070.39
平均時間	0.061	0.093	0.08
車輛數	1台	2台	2台

（6） 車容量 50 公斤-各需求點平均 30 份

在各需求點平均 30 份之情形下，容量可承載 50 公斤之無人車在萬和國中需要車輛數 2 台、平均測試時間為 0.061 秒、平均距離 441.17 公尺。在僑光科技大學需要車輛數 4 台、平均測試時間 0.079 秒、平均距離 2708.56 公尺。在逢甲大學需要車輛數 6 台、平均測試時間 0.173 秒、平均距離 4124.98 公尺，如表 18 所示。

表 18 車容量 50 公斤-各需求點平均 30 份

各需求點平均30份			
需求點	萬和國中(需求點：6)	僑光科大(需求點：11)	逢甲大學(需求點：16)
平均距離	441.17	2708.56	4124.98
平均時間	0.061	0.079	0.173
車輛數	2台	4台	6台

（7） 車容量 50 公斤-多需求點需求量高

在需求點多且需求量高之情形下，容量可承載 50 公斤之無人車在萬和國中需要車輛數 5 台、平均測試時間為 0.085 秒、平均距離 836.74 公尺。在僑光科技大學需要車輛數 9 台、平均測試時間 0.093 秒、平均距離 4848.19 公尺。在逢甲大學需要車輛數 14 台、平均測試時間 0.126 秒、平均距離 6538.82 公尺，如表 19 所示。

表 19 車容量 50 公斤-多需求點需求量高

多需求點需求量高			
需求點	萬和國中(需求點：6)	僑光科技大學(需求點：11)	逢甲大學(需求點：16)
平均距離	836.74	4848.19	6538.82
平均時間	0.085	0.093	0.126
車輛數	5台	9台	14台

（8） 車容量 50 公斤-單一需求點需求量高

在單一需求點需求量較高之情形下，容量可承載 50 公斤之無人車在萬和國中需要車輛數 2 台、平均測試時間為 0.061 秒、平均距離 451.55 公尺。在僑光科技大學需要車輛數 3 台、平均測試時間 0.077 秒、平均距離 1803.47 公尺。在逢甲大學需要車輛數 3 台、平均測試時間 0.096 秒、平均距離 2345.38 公尺，如表 20 所示。

表 20 車容量 50 公斤-單一需求點需求量高

單一需求點需求量高			
需求點	萬和國中(需求點：6)	僑光科技大學(需求點：11)	逢甲大學(需求點：16)
平均距離	451.55	1803.47	2345.38
平均時間	0.061	0.077	0.096
車輛數	2台	3台	3台

2. 基本情境 1 綜合分析

表 21、表 22、表 23 為不同校園規模之測試總結表，分別以距離、時間與車輛數統整分析，並以車容量 25 公斤與 50 公斤無人車做區隔，可看出以下數點結果。首先，在各需求點之需求為平均需求情境下，車載 50 公斤重可配送 2 倍需求量，且總行駛距離與車載 25 公斤相比，相差不超過 1 公里。第二，車容量 50 公斤之測試時間皆小於車容量 25 公斤之測試時間，求解速率較快。第三，在多需求點需求量多之情境下，使用較大容量之無人車有較好之效果。第四，車容量 50 公斤與車容量 25 公斤之無人車，在需求為後者兩倍之情境下，無人車派遣為相同數目。第五，透過分析，規模較小之校園，使用容量較小之無人車較為合適，較能避免虧損情形發生。第六，需求量越大之情境，其所需之測試時間越長。

表 21 萬和國中(需求點：6)之情境 1 分析結果

項目	無人車容量/差異數	平均5份/10份	平均15份/30份	多需求點需求量高	單一需求點需求量高
平均距離	25公斤	265.02	354.09	836.74	448.18
	50公斤	277.34	441.17	836.74	451.55
	差異數	-12.32	-87.08	0	-3.37
平均時間	25公斤	0.233	0.209	0.19	0.216
	50公斤	0.061	0.061	0.085	0.061
	差異數	0.172	0.148	0.105	0.155
派遣車輛數	25公斤	1	2	5	2
	50公斤	1	2	5	2
	差異數	0	0	0	0

表 22 僑光科技大學(需求點：11)之情境 1 分析結果

項目	無人車容量/差異數	平均5份/10份	平均15份/30份	多需求點需求量高	單一需求點需求量高
平均距離	25公斤	1509.78	2596.38	4682.82	1333.03
	50公斤	1993.89	2708.56	4848.19	1803.47
	差異數	-484.11	-112.18	-165.37	-470.44
平均時間	25公斤	0.311	0.303	0.32	0.368
	50公斤	0.093	0.079	0.093	0.077
	差異數	0.218	0.224	0.227	0.291
派遣車輛數	25公斤	2	4	9	3
	50公斤	2	4	9	3
	差異數	0	0	0	0

表 23 逢甲大學(需求點：16)之情境 1 分析結果

項目	無人車容量/平均數	平均5份/10份	平均15份/30份	多需求點需求量高	單一需求點需求量高
平均距離	25公斤	2136.74	3531.03	6832.95	2217.64
	50公斤	2070.39	4124.98	6538.82	2345.38
	差異數	66.35	-593.95	294.13	-127.74
平均時間	25公斤	0.462	0.453	0.559	0.652
	50公斤	0.08	0.173	0.126	0.096
	差異數	0.382	0.28	0.433	0.556
派遣車輛數	25公斤	2	6	14	3
	50公斤	2	6	14	3
	差異數	0	0	0	0

1. 基本情境 2 分析結果

以下為基本情境二之分析結果，根據不同校園規模，分為單一溫層無人車與多溫共配無人車，依照三種不同發車點量進行情境分析比較。

（1） 單一溫層無人車-1 個發車點

在 1 個發車點之情形下，單一不同溫層之無人車之在萬和國中共需車輛數 3 台、平均測試時間為 0.028 秒、平均距離 728.89 公尺。在僑光科技大學需要車輛數 5 台、平均測試時間 0.032 秒、平均距離 4749.13 公尺。在逢甲大學需要車輛數 7 台、平均測試時間 0.078 秒、平均距離 5707.28 公尺，如表 24 所示。

表 24 單一溫層無人車-1 個發車點

單一溫層無人車-1個發車點			
需求點	萬和國中(需求點：6)	僑光科技大學(需求點：11)	逢甲大學(需求點：16)
平均距離	728.89	4749.13	5707.28
平均時間	0.028	0.032	0.078
車輛數	3台	5台	7台

（2） 單一溫層無人車-2 個發車點

在 2 個發車點之情形下，單一不同溫層之無人車之在萬和國中共需車輛數 3 台、平均測試時間為 0.025 秒、平均距離 696.01 公尺。在僑光科技大學需要車輛數 5 台、平均測試時間 0.034 秒、平均距離 3757.67 公尺。在逢甲大學需要車輛數 7 台、平均測試時間 0.042 秒、平均距離 4888.12 公尺，如表 25 所示。

表 25 單一溫層無人車-2 個發車點

單一溫層無人車-2個發車點			
需求點	萬和國中(需求點：6)	僑光科技大學(需求點：11)	逢甲大學(需求點：16)
平均距離	696.01	3757.67	4888.12
平均時間	0.025	0.034	0.042
車輛數	3台	5台	7台

（3） 單一溫層無人車-3 個發車點

在 3 個發車點之情形下，單一不同溫層之無人車之在萬和國中共需車輛數 6 台、平均測試時間為 0.028 秒、平均距離 1115.91 公尺。在僑光科技大學需要車輛數 5 台、平均測試時間 0.022 秒、平均距離 3700.13 公尺。在逢甲大學需要車輛數 7 台、平均測試時間 0.068 秒、平均距離 4297.1 公尺，如表 26 所示。

表 26 單一溫層無人車-3 個發車點

單一溫層無人車-3個發車點			
需求點	萬和國中(需求點：6)	僑光科技大學(需求點：11)	逢甲大學(需求點：16)
平均距離	1115.91	3700.13	4297.1
平均時間	0.028	0.022	0.068
車輛數	6台	5台	7台

（4） 多溫共配溫層無人車-1 個發車點

在 1 個發車點之情形下，多溫共配溫層之無人車之在萬和國中共需車輛數 2 台、平均測試時間為 0.081 秒、平均距離 398.33 公尺。在僑光科技大學需要車輛數 2 台、平均測試時間 0.079 秒、平均距離 1576.56 公尺。在逢甲大學需要車輛數 3 台、平均測試時間 0.082 秒、平均距離 2715.49 公尺，如表 27 所示。

表 27 多溫共配溫層無人車-1 個發車點

多溫共配溫層無人車-1個發車點			
需求點	萬和國中(需求點：6)	僑光科技大學(需求點：11)	逢甲大學(需求點：16)
平均距離	398.33	1576.56	2715.49
平均時間	0.081	0.079	0.082
車輛數	2台	2台	3台

（5） 多溫共配溫層無人車-2 個發車點

在 2 個發車點之情形下，多溫共配溫層之無人車之在萬和國中共需車輛數 2 台、平均測試時間為 0.033 秒、平均距離 441.35 公尺。在僑光科技大學需要車輛數 2 台、平均測試時間 0.049 秒、平均距離 1970.29 公尺。在逢甲大學需要車輛數 3 台、平均測試時間 0.068 秒、平均距離 3519.38 公尺，如表 28 所示。

表 28 多溫共配溫層無人車-2 個發車點

多溫共配溫層無人車-2個發車點			
需求點	萬和國中(需求點：6)	僑光科技大學(需求點：11)	逢甲大學(需求點：16)
平均距離	441.35	1970.29	3519.38
平均時間	0.033	0.049	0.068
車輛數	2台	2台	3台

（6） 多溫共配溫層無人車-3 個發車點

在 3 個發車點之情形下，多溫共配溫層之無人車在萬和國中共需車輛數 2 台、平均測試時間為 0.031 秒、平均距離 384.1 公尺。在僑光科技大學需要車輛數 2 台、平均測試時間 0.061 秒、平均距離 2248.78 公尺。在逢甲大

學需要車輛數 3 台、平均測試時間 0.102 秒、平均距離 2734.73 公尺，如表 29 所示。

表 29 多溫共配溫層無人車-3 個發車點

多溫共配溫層無人車-3個發車點			
需求點	萬和國中(需求點：6)	僑光科技大學(需求點：11)	逢甲大學(需求點：16)
平均距離	384.1	2248.78	2734.73
平均時間	0.031	0.061	0.102
車輛數	2台	2台	3台

2. 基本情境 2 綜合分析

表 30、表 31、表 32 為不同校園規模之發車點與車輛類型測試總結表，分別為平均距離、平均時間與派遣車輛數統整分析，並以單一溫層無人車、多溫共配無人車與兩種無人車之混合做分類，可看出以下數點結果。測試時間部分，單一溫層無人車配置相對多溫共配無人車簡易，因此其求解時間較多溫共配無人車短；派遣車輛數部分，發車點數多寡對無人車輛類型較無影響。總體而言，車輛類型為多溫共配無人車之模式其效能在平均距離與派遣車輛皆有較好之結果。不同規模之校園有不同無人車輛類型與發車點之搭配，在規模為需求點 6 之情境，單一溫層無人車適用 2 個發車點、多溫共配無人車適用 3 個發車點，且使用 3 個發車點搭配多溫共配無人車為較佳選擇；在規模為需求點 11 之情境，單一溫層無人車適用 3 個發車點、多溫共配無人車適用 1 個發車點，且使用 1 個發車點搭配多溫共配無人車為較佳選擇；在規模為需求點 16 之情境，單一溫層無人車適用 2 個發車點、多溫共配無人車適用 1 個發車點，且使用 1 個發車點搭配多溫共配無人車為較佳選擇。除此之外，本研究同時考量兩種車輛同時存在同一校園之想法，對於規模為需求點 6 之情境，設置 1 個發車點為佳、規模為需求點 11 之情境，設置 2 個發車點為佳、規模為需求點 16 之情境，設置 3 個發車點為佳。

表 30 萬和國中(需求點：6)之校園規模情境二分析結果

項目	無人車類型	1個發車點	2個發車點	3個發車點
平均距離	單一溫層無人車	728.89	696.01	1115.91
	多溫共配無人車	398.33	441.35	384.1
	兩種無人車混合	563.61	568.68	750.005
平均時間	單一溫層無人車	0.028	0.025	0.028
	多溫共配無人車	0.081	0.049	0.031
	兩種無人車混合	0.0545	0.037	0.0295
派遣車輛數	單一溫層無人車	3	3	6
	多溫共配無人車	2	2	2
	兩種無人車混合	3	3	4

表 31 僑光科技大學(需求點：11)之校園規模情境二分析結果

項目	無人車類型	1個發車點	2個發車點	3個發車點
平均距離	單一溫層無人車	4749.13	3757.67	3700.13
	多溫共配無人車	1576.56	1970.29	2248.78
	兩種無人車混合	3162.845	2863.98	2974.455
平均時間	單一溫層無人車	0.032	0.034	0.022
	多溫共配無人車	0.079	0.049	0.061
	兩種無人車混合	0.0555	0.0415	0.0415
派遣車輛數	單一溫層無人車	5	5	5
	多溫共配無人車	2	2	2
	兩種無人車混合	4	4	4

表 32 逢甲大學（需求點：16）之校園規模情境二分析結果

項目	無人車類型	1個發車點	2個發車點	3個發車點
平均距離	單一溫層無人車	5707.28	3577.67	4297.1
	多溫共配無人車	2715.49	3519.38	2734.73
	兩種無人車混合	4211.385	3548.525	3515.915
平均時間	單一溫層無人車	0.078	0.034	0.068
	多溫共配無人車	0.082	0.068	0.102
	兩種無人車混合	0.08	0.051	0.085
派遣車輛數	單一溫層無人車	7	7	7
	多溫共配無人車	3	3	3
	兩種無人車混合	5	4	5

4.2.3 綜合情境分析

本研究茲為近一步分析不同校園規模之最佳配置模式，設置車容量、無人車之車輛類型、發車點數、可派遣之車輛數及隨機需求量等參數，並依照基本情境 1 及基本情境 2 之分析結果，分別對萬和國中(需求點：6)，校園面積為 26424 平方公尺，僑光科技大學(需求點：11)，校園面積為 57262 平方公尺及逢甲大學(需求點：16)，校園面積為 263143 平方公尺之三種校園規模進行分析，藉以分析此三種校園規模之最佳化派遣，並沿用基本情境之需求點與發車點位置。以下分別說明不同校園規模之參數設置及其結果。

1. 小型校園情境設置

小型校園以萬和國中作為校園情境模擬環境，在需求點需求為隨機需求量之情境下，在該環境中設置 6 個需求點與 2 個發車點，分別位於校門口(發車點 1)及活動中心(發車點 2)。以單一溫層無人車、車容量 25 公斤進行派遣任務。此外，初步規劃設置 3 台車，分別於校門口(發車點 1)設置 2 台車及活動中心(發車點 2)設置 1 台車。

表 33 為萬和國中情境分析結果，在上述情境參數設置下，萬和國中總共需派遣 3 台單一溫層無人車；在平均時間為 0.112 秒下，其測試最短時間為 0.053 秒；3 台單一溫層無人車所行駛總平均距離為 650.81 公尺。圖 22 為其情境模擬路線圖。

表 33　萬和國中情境分析表

萬和國中(需求點：6)			
最短時間	平均時間	平均距離	車輛數
0.053	0.112	650.81	3台

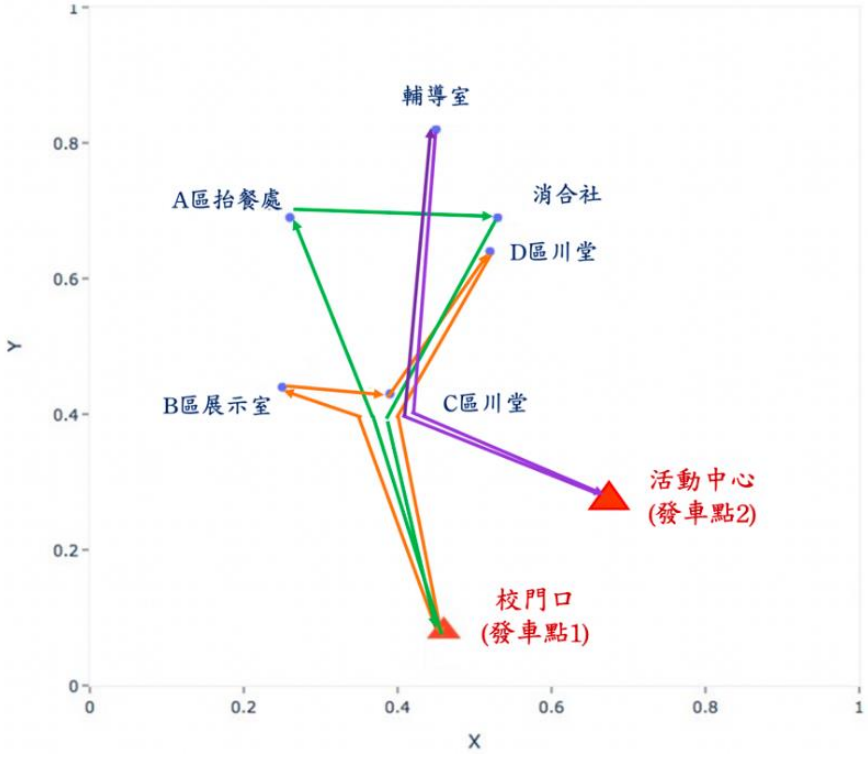


圖 22　萬和國中情境模擬路線圖

2. 中型校園情境設置

中型校園以僑光科技大學作為校園情境模擬環境，並在其校園中設置 11 個需求點。透過基本情境分析結果，以車容量 50 公斤，並使用多溫共配無人車進行派遣任務。除此之外，並在校園中設置 1 個發車點，位於校門口(發車點 1)。在需求點需求為隨機量之情況下，本研究初步於校門口(發車點 1)規劃設置 5 台車。

表 34 為僑光科技大學情境分析結果，在上述情境參數設置下，僑光科技大

學總共需派遣 5 台多溫共配無人車；在平均時間為 0.136 秒下，其測試最短時間為 0.1 秒；5 台多溫共配無人車所行駛總平均距離為 3177.44 公尺。圖 23 為其情境模擬路線圖。

表 34　僑光科技大學情境分析表

僑光科技大學(需求點：11)			
最短時間	平均時間	平均距離	車輛數
0.1	0.136	3177.44	5台

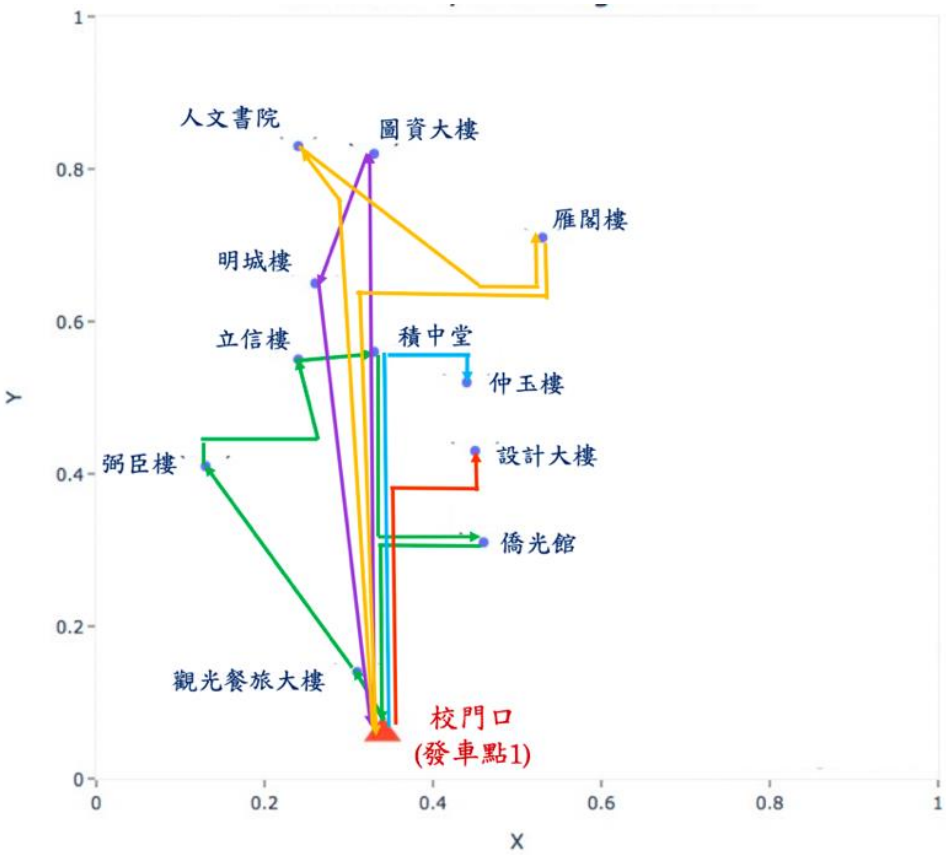


圖 23　僑光科技大學情境模擬路線圖

3. 大型校園情境設置

大型校園以逢甲大學作為校園情境模擬環境，並在其校園中設置 16 個需求點。透過基本情境分析結果，以車容量 25 及 50 公斤，並同時使用多溫共配無人車及單一溫層無人車進行派遣任務。除此之外，並在校園中設置 3 個發車點，分別位於校門口(發車點 1)、校內便利商店(發車點 2)及逢大 1 統一門市(發車點 3)。在需求點需求為隨機量之情況下，本研究初步規劃設置 16 台車，分別於校門口(發車點 1)設置 3 台多溫共配無人車、3 台單一溫層無人車；於校內便利商店(發車點 2)設置 3 台多溫共配無人車、2 台單一溫層無人車；於逢大 1 統一門市(發車點 3)設置 2 台多溫共配無人車、3 台單一溫層無人車。

表 35 為逢甲大學情境分析結果，在上述情境參數設置下，逢甲大學總共需派遣 8 台無人車，7 台多溫共配無人車及 1 台單一溫層無人車；在平均時間為

0.11 秒下，其測試最短時間為 0.097 秒；8 台無人車所行駛總平均距離為 5927.34 公尺。圖 24 為其情境模擬路線圖。

表 35 逢甲大學情境分析表

逢甲大學(需求點：16)			
最短時間	平均時間	平均距離	車輛數
0.097	0.11	5927.34	8台

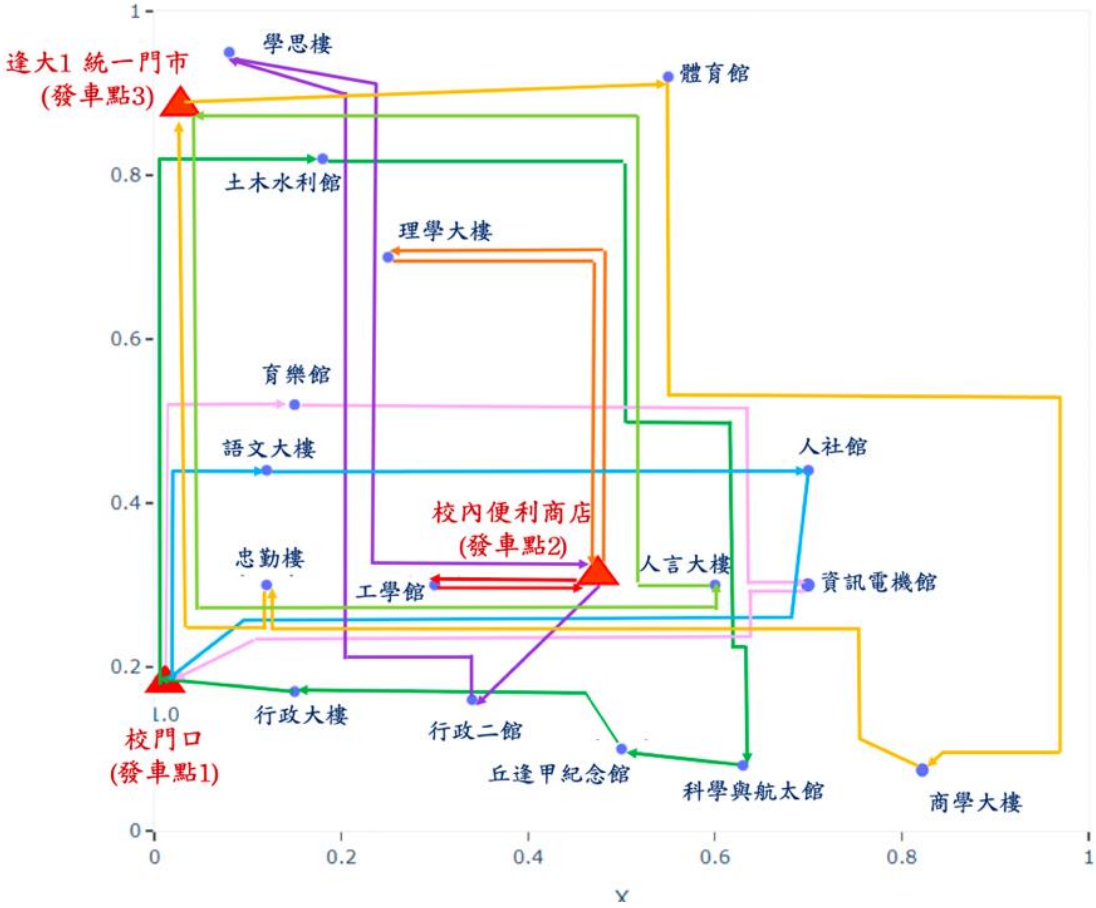


圖 24 逢甲大學情境模擬路線圖

第五章、 管理意涵

本研究根據敏感度、基本情境 1、基本情境 2 與綜合情境分析所得出之結果，針對測試時間、平均距離、派遣車輛數、發車點設置、校園無人車類型與校園規模提出相對應之管理策略，以下為本研究提出之管理規劃論點。

1. 測試時間

對測試時間而言，需求點之數目改變為最大影響因子，需求點數越多，其所需決策時間越長。除此之外，單一溫層無人車因物流模式為單一化配送模式，因此與多溫共配無人車之派遣相比，其決策時間也較短。

2. 平均距離

在平均距離部分，需求點數改變為最大影響因子，需求點數目越多，其距離越長，也會讓派遣任務複雜化。

3. 派遣車輛數

在派遣車輛數部分，車容量改變為最大影響因子，車容量越大，可派遣較少輛無人車。除此之外，多溫共配無人車派遣車輛數比單一溫層佳，因為無需區分溫度差異，可混合不同溫度之餐飲進行配送，但此派遣車輛之類型與容量配置則需要考慮購買校園無人車之成本問題。

4. 發車點設置

在發車點設置方面，透過上述分析，可知並非一味增加越多發車點越好。不同校園規模、不同需求量與不同校園無人車之類型，皆有可能影響到所需設置之發車點數量，此會間接影響到平均距離及派遣車輛數等成本問題。

5. 校園無人車類型

在校園無人車方面，不論是派遣車輛數或平均距離部分，在情境中，多溫共配無人車之模擬結果與單一溫層無人車所產生之結果相比，前者為較佳無人車之車種類型。除此之外，多溫共配無人車也會有較高之配送品質，雖然決策時間較長，但是整體模擬求解時間不會超過 1 秒鐘，並不會有過度差異產生。但若需求量較無變化、屬於每天配送數量為規律之校園，並不需要此種金額高昂之無人車。且國內冷鏈物流無人車現今才逐漸興起，其穩定性不足，且台灣於校園無人車方面尚未引進，無法知道真實餐飲配送情形。

6. 校園規模

不同校園規模有不同之最佳配置方法，隨著需求量與校園類型不同，同樣也會有不同決策準則，需要各院校依照該校之需求與編列預算去做無人車之購買與發車點設置數量。

綜合分析所提出之不同校園規模所適用之相關類型，此情境參數選擇以不考慮成本與現況發展之情況，去做最佳化餐飲配送模式。共分為小型、中型與大型校園進行分析。表 13 為各規模校園無人車參數配置規劃。

在小型規模校園內，面積於 26424 平方公尺以下，大多為國、高中院校之校園面積，派遣行為較為規律，餐飲部分大多為統一訂購。各需求點大多為平均需求，因此適用單一溫層無人車，無需使用較為複雜之派遣行為，可使求解時間與距離縮短、總體利益最佳化。在發車點設置上透過分析可知，應設置 2 個發車點。也因校園規模較小，依照模擬結果分析，可配置裝載容量為 25 公斤之單一溫層無人車。無人車之車輛數部分，根據萬和國中之校園情境分析，初步適合配

置 3 台以上之無人車，日後可依照學生人數，對無人車進行倍數增加。且根據分析資料結果顯示，設置 3 個發車點反而使車輛平均行駛距離增加，故推測因小型規模校園之用地面積較小，不宜使發車點設置密度過高。

在中型規模校園內，面積介於 26424 平方公尺至 263143 平方公尺間，大多為小型大學、專科院校之校園面積，派遣行為較多元化、需求不固定，學生餐飲選擇自由度較高。且校園內學生密度較高，因此需求方面以多需求點需求量高為較常發生之情境，可選擇使用多溫共配無人車進行派遣。在發車點設置上透過分析可知，應設置 1 個發車點。除此之外，校園規模居中，所適用之無人車為裝載容量 50 公斤之多溫共配無人車，無需額外增加無人車容量。無人車之車輛數部分，根據僑光科技大學之校園情境分析，初步適合配置 2 台以上之無人車，日後可依照學生人數，對無人車進行倍數增加。根據分析資料結果顯示，因大專院校之餐飲並無統一訂購機制、烹煮機制，餐飲方面選擇較多元化，因此不適合配置單一溫層無人車進行派遣。除此之外，因其學生人數密度比較高，若配置為單一溫層無人車或兩種無人車混合設置，只配置 1 個發車點進行車輛配送，即導致派遣距離大幅增加，因此須改設置 2 個發車點為替代策略。

在大型規模校園內，面積於 263143 平方公尺以上，大多為普通大學之校園面積，派遣行為較多元化、需求不固定，學生餐飲選擇自由度較高。且校園內學生人數眾多，在用餐時刻會有大量學生進行餐點購買動作。除此之外，需求點中也同時存在辦公大樓或研究中心等非教學型建築存在，因此需求方面為多需求點需求量高之情況，且同時存在需求較少之需求點，適用單一溫層無人車與多溫共配無人車兩種，以應付不同需求之概況。在發車點設置上透過分析可知，應設置 3 個發車點。其校園規模較大，適用之裝載容量除了維持最低容量之公斤數，更可以視需求增加容量。無人車之車輛數部分，根據逢甲大學校園進行情境分析後，初步適合配置 5 台以上之無人車，日後可依照學生人數，對無人車進行倍數增加。但此規模之校園並不適合只設置一處發車點，因其校園規模較大，需要多個發車點以應付較多需求點之情形。若無設置多發車點，會造成無人車配送路線過長之情形。導致整體運輸時間過長，無法應付校園內部距離主發車點較遠之需求點需求。

表 36 各規模校園參數配置適用表

配置規劃							
校園規模	校園面積	各需求點需求量	需求點數	車輛類型	無人車可裝載容量	基本車輛數	發車點數量
小型	26424 平方公尺以下	平均需求	<=6	單一溫層無人車	25 公斤	3 台	2
中型	26424~263143 平方公尺	多需求點需求量高	7<=11<=15	多溫共配無人車	50 公斤	2 台	1
大型	263143 平方公尺以上	多需求點需求量高	>=16	兩種無人車並存	25/50 公斤(視需求增加容量)	5 台	3

第六章、 結論與建議

現今外送服務平台興起，儼然成為一種主流，然而許多校園仍然禁止校外汽、機車進入校園中，導致訂餐者須親自到校門口取餐，且國內目前於校園內並無相關餐飲無人車完善配送措施。因此，本研究以強化學習對校園無人車之車輛路徑問題進行規劃與分析，並將物流策略納入探討範圍，得到不同校園規模之無人車餐飲配送最佳化模式。

本研究以逢甲大學為基本測試模擬環境，分析批量、車容量、迴圈、訓練步數、餐飲重量與需求點等參數之設定，進行強化學習規劃與測試。將測試之結果與 Lingo 進行比較，其測試時間與批量大小成正比，派遣距離則與批量大小成反比，因此本研究以批量 256 為基礎做進一步分析與討論。且根據基礎測試分析之結果可發現強化學習訓練時間較長，且無人車實際行走距離和車輛數派遣與傳統學習相比並無太大差異，但其求解速度明顯優於傳統學習。

本研究透過基礎測試結果與敏感度分析結果，提出變動各需求點之需求量對派遣影響較小，車容量與需求點改變對餐飲配送影響較大。除此之外，透過各校園需求點數與位置、需求量、車容量、無人車類型、發車點數量與設置位置等參數進行基本情境與綜合情境探討。根據綜合情境顯示，在各需求點需求量為隨機設置基礎下，小型校園適合配置車容量 25 公斤、2 個發車點與 3 台單一溫層無人車進行派遣。中型校園適合配置車容量 50 公斤、1 個發車點與 5 台多溫共配無人車進行派遣。大型校園適合配置車容量 25 及 50 公斤、3 個發車點、7 台多溫共配無人車與 1 台單一溫層無人車進行派遣。

本研究分析校園無人車不同派遣情境，獲悉不同校園面積規模之車輛派遣策略皆不同，以小型規模校園而言，因需求變化較單一且校園面積小，適合使用基本車容量(25 公斤)與單一溫層無人車，發車點數量設置 2 個以供及時配送；中型校園規模則因需求較多元且校園面積沒有那麼寬闊，因此適合設置 1 個發車點，並且同時配置多溫共配無人車；大型校園因需求大且餐飲種類多元，校園面積也寬廣，因此適合設置 3 個以上發車點，在車種方面可以單一與多溫無人車混合使用，以滿足不同需求。

透過本研究之研究成果，提出以下建議供日後進行研究使用。首先，後續研究可透過其他強化學習模式進行校園無人車車輛派遣測試，對於各需求點之開始、每需求點之停留時間、每個發車點之使用時間及各需求點之獎勵制度進行分析與探討，供車輛派遣行為有更完善之結果存在。再者，基於環保意識高漲，後續研究可透過收送機制，讓已使用過之餐具進行消毒作業，供下次使用，來促進環保、循環再利用。

參考文獻

胡尚民、沈惠璋 (2020)，「基于強化學習的電動車路徑優化研究」，*TANET2019 台灣網際網路研討會*，頁 187-192。

莫凡 (2016)，什麼是 Policy Gradients，擷取日期：2020 年 10 月 20 日，網站：https://mofanpy.com/tutorials/machine-learning/reinforcement-learning/intro-PG/。

莫凡 (2016)，什麼是 Actor Critic，擷取日期：2020 年 10 月 20 日，網站：https://mofanpy.com/tutorials/machine-learning/reinforcement-learning/intro-AC/

葉強 (2017)，《強化學習》第三講 動態規劃尋找最優策略，擷取日期：2020 年 10 月 25 日，網站：https://zhuanlan.zhihu.com/p/28084955。

張震、高軍偉、張彩虹、趙婷婷（2015）。一種基於多智能體強化學習的運輸車路徑優化方法。取自 https://patents.google.com/patent/CN104680264A/zh

陳信宇（2010）。機器人之增強式學習：整合遺傳規劃與神經網路。取自 https://ndltd.ncl.edu.tw/cgi-bin/gs32/gsweb.cgi/ccd=8fJnzB/search#result

鍾玉峰、張文鎰、蔡惠峰、廖建明、許泰文（2019）。強化學習之發展： 以船行路徑最佳化為例。取自 https://www.airitilibrary.com/Publication/alDetailedMesh?docid=P20200109001-201912-202001090021-202001090021-187-192

Aigerim Bogyrbayevay , Sungwook Jang , Ankit Shahy , Young Jae Jang , & Kwony, C. (2020), A Reinforcement Learning Approach for Rebalancing Electric Vehicle Sharing Systems, arXiv:2010.02369 [cs.LG], available at: https://arxiv.org/abs/2010.02369 (accessed 4 December ,2020).

Arthur Delarue, Ross Anderson, & Tjandraatmadja, C. (2020), "Reinforcement Learning with Combinatorial Actions： An Application to Vehicle Routing", arXiv:2010.12001 [cs. LG], available at: https://arxiv.org/abs/2010.12001 (accessed 12 December ,2020).

Arun Kumar Kalakanti, Shivani Verma, Topon Paul, & Yoshida, T. (2019), "URL Solver Pro： Reinforcement Learning for Solving Vehicle Routing Problem", *2019 1st International Conference on Artificial Intelligence and Data Sciences (AiDAS)* ,doi: 10.1109/AiDAS47888.2019.8970890.

Bharathan Balaji, Jordan Bell-Masterson, Enes Bilgin and Andreas Damianou, Pablo Moreno Garcia, Arpit Jain and Runfei Luo, Alvaro Maggiar , . . . Ye, C. (2019), "ORL：Reinforcement Learning Benchmarks for Online Stochastic Optimization Problems", arXiv:1911.10641 [cs.LG], available at: https://arxiv.org/abs/1911.10641 (accessed 15 December ,2020).

Emmanouil Tzorakoleftherakis (2019), Getting Started with Reinforcement Learning, Retrieved June 15, 2020, website: https://www.digitalengineering247.com/article/getting-started-with-reinforcement-learning/partner-content.

Hanjun Dai, Elias B. Khalil, Yuyu Zhang, Bistra Dilkina, & Song, L. (2017), "Learning Combinatorial Optimization Algorithms over Graphs", arXiv:1704.01665 [cs.LG], available at: https://arxiv.org/abs/1704.01665 (accessed 14 December ,2020).

Irwan Bello, Hieu Pham, Quoc V. Le, Mohammad Norouzi, & Bengio, S. (2017), "NEURAL COMBINATORIAL OPTIMIZATION WITH REINFORCEMENT LEARNING", arXiv:1611.09940 [cs.AI], available at: https://arxiv.org/abs/1611.09940 (accessed 12 December ,2020).

Maxim Lapan (2019), Deep Reinforcement Learning Hands-On, United Kingdom: Packt Publishing.

Mohammadreza Nazari, Afshin Oroojlooy, Lawrence V. Snyder, & Takáč, M. (2018), "Reinforcement Learning for Solving the Vehicle Routing Problem", *32nd Conference on Neural Information Processing Systems (NeurIPS 2018)*, available at: https://papers.nips.cc/paper/2018/file/9fb4651c05b2ed70fba5afe0b039a550-Paper.pdf (accessed 5 October ,2020).

Patrick, S. D., Nycz, A., Noakes, M. W., & Gaul, K. T. (2018), "Reinforcement Learning for Generating Toolpaths in Additive Manufacturing", OSTI.GOV, available at: https://www.osti.gov/biblio/1474597 (accessed 14 December ,2020).

Palash Goyal , & Ferrara, E. (2018), "Graph embedding techniques, applications, and performance： A survey", arXiv:1705.02801 [cs.SI], available at: https://arxiv.org/abs/1705.02801 (accessed 16 December ,2020).

Sudharsan Ravichandiran (2019)，Hands-On Reinforcement Learning with Python, United Kingdom: Packt Publishing.

Sutton, Richard S. and Barto, Andrew G. (2018), Reinforcement Learning: An Introduction, USA: Bradford Book.

Vaibhav Kumar (2020)，Mathematical Analysis of Reinforcement Learning — Bellman Optimality Equation, Retrieved October 22, 2020, website: https://towardsdatascience.com/mathematical-analysis-of-reinforcement-learning-bellman-equation-ac9f0954e19f.

Wouter Kool, Herke van Hoof, & Welling, M. (2019), "Attention, Learn to Solve Routing Problems!", *ICLR 2019*, available at: https://arxiv.org/pdf/1803.08475.pdf (accessed 9 December ,2020).

Yujiao Hu, Yuan Yao, & Lee, W. S. (2020), "A reinforcement learning approach for optimizing multiple traveling salesman problems over graphs", *Knowledge-Bases Systems*, Volume 204, available at: https://www.sciencedirect.com/science/article/abs/pii/S0950705120304445?via%3Dihub (accessed 1 December ,2020).

Zheng Weijian, Wang Dali, & Fengguang, S. (2020), "OpenGraphGym： A Parallel Reinforcement Learning Framework for Graph Optimization Problems", *International Conference on Computational Science 2020*, pp. 439-452, available at:

https://link.springer.com/content/pdf/10.1007%2F978-3-030-50426-7_33.pdf (accessed 1 December ,2020).