

國立交通大學
運輸與物流管理學系碩士班

碩士論文

使用電動車之一般共乘問題
最佳化模式建構與求解

Optimization Model and Solution Algorithm for
the General Share-a-ride Problem with Electric
Vehicles

研究生：李依庭

指導教授：盧宗成 教授

中華民國一〇八年七月

使用電動車之一般共乘問題最佳化模式建構與求解

Optimization Model and Solution Algorithm for the General
Share-a-ride Problem with Electric Vehicles

研 究 生：李依庭 Student : Yi-Ting Li

指導老師：盧宗成 Advisor : Chung-Cheng Lu

國立交通大學
運輸與物流管理學系
碩 士 論 文

A Thesis

Submitted to Department of Transportation and Logistics Management

College of Management

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Master

in

Logistics Management

July 2019

Hsinchu, Taiwan, Republic of China

中 華 民 國 一 ○ 八 年 七 月

使用電動車之一般共乘問題最佳化模式建構與求解

學生：李依庭

指導教授：盧宗成 教授

國立交通大學運輸與物流管理學系 碩士班

摘要

本研究針對一般共乘問題(General Share-a-ride Problem, G-SARP)問題進行延伸，以充電式(Plug-in)電動計程車及汽油計程車之混合車隊進行運輸與配送服務，此一延伸問題稱為使用電動車(Electric Vehicles, EVs)之一般共乘問題(G-SARP-EVs)。本研究以計程車營運業者的角度出發，於特定範圍內進行服務，考量營運時間內混合計程車隊與人貨配送需求之配對、車輛排程與電動計程車充電之規劃，以營運業者利潤最大化為目標，在固定且已知的車隊數量及車輛容量限制下，僅針對營運時間前完成預約之乘客及貨物需求進行服務。

根據 G-SARP-EVs 問題之特性，本研究建構時空網路(Time-Space Networks)描述人流、物流與車流在時空維度上面的移動情形，並將時空網路分成需求面(包含人流及物流)與供給面(包含電動車流及汽油車流)時空網路，其中供給面時空網路以每輛車輛作為分層依據，以有效進行車輛排程之描述及電動計程車之電量追蹤，並於建構之時空網路定式 G-SARP-EVs 數學規劃模式，更發展模擬退火法(Simulated Annealing)提升大規模問題之求解效率。

為驗證及測試本研究發展之使用電動車之一般共乘問題最佳化模式及啟發式演算法之應用績效，本研究以業者提供之實際營運資料，進行調整後產出不同規模之測試範例，並以最佳化軟體 Gurobi 進行 G-SARP-EVs 模式構建與求解；模擬退火法則以 Python 語言進行撰寫及求解，完成測試後輸出結果並進行比較與分析。小規模範例測試中，不論是 Gurobi 或是模擬退火法皆能於短時間內求得最佳解，並輸出車輛排程及充電規劃結果；中、大規模範例測試中，可驗證本研究發展之模擬退火法的求解效率大幅領先最佳化求解軟體 Gurobi。

關鍵詞：一般共乘問題(G-SARP)、充電式電動車、時空網路、最佳化模式、模擬退火法

Optimization Model and Solution Algorithm for the General Share-a-ride Problem with Electric Vehicles

Student: Yi-Ting Li

Advisor: Chung-Cheng Lu

Department of Transportation and Logistics Management
National Chiao Tung University

Abstract

This research introduces an extension of the general share-a-ride problem or G-SARP, called the G-SARP with electric vehicles (G-SARP-EVs). This problem considers a taxi fleet with mixed plug-in electric vehicles and gasoline vehicles can service passenger and parcel requests simultaneously. In this problem, taxis are allowed to convey more than one passenger at the same time, and there is no restriction on the maximum riding time of a passenger. In addition, the number of parcel requests that can be inserted between the pick-up and drop-off points of a passenger is limited only by vehicle capacity. This problem considers only advance requests that are given prior to the beginning of the planning horizon.

The research develops a multi-layer time-space network to effectively describe the movements of passengers, parcels and taxis in the spatial and temporal dimensions. Each taxi operates on its own layer of the network for tracking the energy consumption and load of taxis. The EVs have the priority to service the passenger and parcel requests. An optimization model is to determine the optimal schedule for the taxi fleet to service the given requests. The objective is to maximize the profit of the taxi company. Also, a meta-heuristic based on simulated annealing is proposed to efficiently solve large-scale instances of the problem.

To examine the performance of the proposed model and the heuristic, this study generates a number of instances with various sizes from the data provided by a logistics service provider in Taiwan. The model is solved by the optimization solver, Gurobi. The results from solving small-size instances show that both Gurobi and SA can obtain the optimal schedules for the taxi fleet to service all the requests in a short time, while the results from solving medium-size and larger-size instances show that SA can obtain a better solution in a shorter time than Gurobi.

Keywords: G-SARP, Plug-in Electric Vehicles, Time-space networks, Optimization, Simulated Annealing

誌謝

回想起剛推甄上研究所的那個時候，找了許多老師面談，猶豫不決不知道該如何選擇的我，如今，兩年的碩士生活即將告一段落。

首先，第一個要感謝的人就是我的指導老師－盧宗成教授，很感謝在這兩年的碩士生活中，老師總是耐心地指導我們，每個星期都花很多時間在與我們討論論文，且總是把氣氛營造地輕鬆愉悅，即便是論文進度落後或是卡關之際，老師也都會用鼓勵的方式代替責備，並始終相信我們能夠解決，很感謝老師總是給予很多正向的激勵，也很願意傾聽我們的想法，最後再給予合適的建議，謝謝老師這兩年的用心指導，才能讓這本論文順利產出。

再來，也想要感謝系上的老師們，尤其是物流組的老師，可以感覺到每位老師們都很認真地準備課程內容，上課時更是毫不吝嗇地希望把更多內容傳授給我們，也因為修習了這些與物流相關的基本課程，最後才能順利地在這些基礎之上完成這本論文。

接著，也要感謝一起讀研究所的同學們，不論是從碩一開始一起在統倉訂外送偶爾一起發瘋一起熬夜趕報告的同學們，或是接下計劃後一起做計劃一起與老師視訊開會一起同甘共苦的王盧 lab 同學們，亦或是一起來到交大一起買飯一起上課寫作業討論一起放風的成大同學們……，很感謝因為這些同學們，才讓研究所的生活更加溫馨又多采多姿，也才能一起又各自順利地完成畢業論文。

最後，要感謝我的家人們，感謝我的家人們總是很支持我的所有決定，也很相信我沒有回家就是留在新竹忙碌，給我很大的空間作我想做的事，最後才能順利產出這本論文。

謝謝所有的幫助及陪伴，才能順利產出論文，完成兩年的碩士生活。

李依庭 謹誌

108 年 7 月

目錄

中文摘要.....	i
英文摘要.....	ii
誌謝.....	iii
目錄.....	iv
圖目錄.....	vii
表目錄.....	ix
第一章 緒論.....	1
1.1 研究背景與動機.....	1
1.2 研究目的.....	3
1.3 研究範圍與對象.....	3
1.4 研究方法與流程.....	3
第二章 文獻回顧.....	5
2.1 共乘問題介紹.....	5
2.1.1 撥召問題 (Dial-a-ride Problem, DARP).....	5
2.1.2 共乘問題 (Share-a-ride Problem, SARP).....	6
2.1.3 一般共乘問題 (General Share-a-ride Problem, G-SARP).....	6
2.1.4 共乘問題演進比較.....	7
2.2 電動車應用在運輸與物流.....	9
2.3 時空網路模型應用在運輸與物流.....	9
2.4 模擬退火法應用在運輸與物流.....	11
2.5 小結.....	12
第三章 研究方法.....	13
3.1 問題描述.....	13
3.2 時空網路模型.....	15

3.2.1	需求面時空網路.....	16
3.2.2	供給面－汽油車時空網路.....	19
3.2.3	供給面－電動車時空網路.....	22
3.3	使用電動車之一般共乘問題最佳化模式.....	26
3.4	小範例測試.....	30
3.4.1	參數設定.....	30
3.4.2	測試結果.....	31
第四章	啟發式演算法.....	35
4.1	模擬退火法架構與介紹.....	35
4.2	可行解之呈現.....	36
4.3	起始解之產生.....	37
4.4	鄰近解之產生.....	43
4.4.1	需求插入法.....	44
4.4.2	排程交換法.....	48
4.4.3	需求交換法.....	50
第五章	範例測試與分析.....	54
5.1	範例設計.....	54
5.1.1	輸入資料說明.....	54
5.1.2	相關參數設定.....	56
5.1.3	模式輸出與環境設定.....	56
5.2	模擬退火法參數設定.....	57
5.3	測試與結果.....	60
5.3.1	小規模範例.....	60
5.3.2	中規模範例.....	63
5.3.3	大規模範例.....	66
5.4	參數情境分析.....	69
5.4.1	政策成本.....	69

5.4.2	耗電量	70
第六章	結論與建議	72
6.1	結論	72
6.2	未來研究建議	73
參考文獻		74

圖目錄

圖 1.1 全球零售電子商務銷售額.....	1
圖 1.2 研究流程圖.....	4
圖 3.1 共乘問題(SARP).....	13
圖 3.2 一般共乘問題(G-SARP).....	14
圖 3.3 需求面時空網路示意圖.....	18
圖 3.4 汽油車時空網路示意圖.....	21
圖 3.5 電動車時空網路示意圖.....	25
圖 3.6 小範例之車輛排程結果(EV1).....	32
圖 3.7 小範例之車輛排程結果(EV2).....	33
圖 3.8 小範例之車輛排程結果(GV).....	34
圖 4.1 模擬退火法之架構圖.....	35
圖 4.2 可行解的呈現.....	37
圖 4.3 初始化解的呈現.....	37
圖 4.4 產生起始解之流程圖.....	38
圖 4.5 完成插入順序 1 需求 2 之現解的呈現.....	39
圖 4.6 需求 4 候選插入車輛之解的呈現.....	40
圖 4.7 插入需求 2 於 EV2 之車輛容量限制判斷.....	41
圖 4.8 插入需求 4 於 EV1 之旅行時間判斷.....	41
圖 4.9 需充電之車輛排程範例.....	42
圖 4.10 插入充電站之判斷.....	42
圖 4.11 插入充電站之車輛路徑結果.....	42
圖 4.12 鄰近解產生之主要架構圖.....	44
圖 4.13 需求插入法之架構圖.....	45
圖 4.14 產生需求插入法之候選需求排程範例.....	46
圖 4.15 需求插入法之候選插入車輛路徑.....	46
圖 4.16 插入需求 1 於 EV2 之車輛容量限制判斷.....	47
圖 4.17 路徑交換法之架構圖.....	48
圖 4.18 需求交換法之架構圖.....	50
圖 4.19 產生需求交換法之候選需求列表之範例.....	51
圖 4.20 完成移除充電站編碼之車輛排程範例.....	52
圖 4.21 需求交換法之候選交換車輛排程.....	52
圖 5.1 路網規劃資料.....	55
圖 5.2 敏感度分析 - 起始溫度.....	59
圖 5.3 敏感度分析 - 終止溫度.....	59
圖 5.4 敏感度分析 - 冷卻率.....	59

圖 5.5 敏感度分析 - 單一溫度迭代次數.....	59
圖 5.6 範例 6a 之求解結果(模擬退火法).....	62

表目錄

表 2.1 共乘問題演進比較表.....	8
表 3.1 小範例測試參數設定.....	30
表 3.2 小範例起迄時間窗及乘客當量數.....	30
表 3.3 小範例問題規模.....	31
表 3.4 小範例測試結果.....	31
表 4.1 需求起迄時窗表.....	39
表 4.2 插入需求列表.....	39
表 4.3 各需求之載/卸乘客之數量列表	40
表 4.4 需求插入法之候選需求列表.....	46
表 4.5 需求交換法之候選需求列表.....	51
表 5.1 電動車規格.....	54
表 5.2 需求規劃資料範例.....	55
表 5.3 收入/成本參數設定	56
表 5.4 參數測試數值列表.....	57
表 5.5 模擬退火法參數敏感度分析.....	58
表 5.6 小規模範例規模.....	60
表 5.7 小規模範例測試結果-Gurobi	61
表 5.8 模擬退火法測試比較-小規模範例.....	62
表 5.9 小規模範例測試結果-SA	63
表 5.10 最佳化求解軟體與模擬退火法之比較-小規模範例.....	63
表 5.11 中規模範例規模.....	64
表 5.12 中規模範例測試結果 - Gurobi	64
表 5.13 模擬退火法測試比較-中規模範例.....	65
表 5.14 中規模範例測試結果 - SA	65
表 5.15 最佳化求解軟體與模擬退火法之比較-中規模範例.....	66
表 5.16 大規模範例規模.....	66
表 5.17 模擬退火法測試比較-大規模範例.....	67
表 5.18 大規模範例測試結果-SA	68
表 5.19 最佳化求解軟體與模擬退火法之比較 - 大規模範例.....	68
表 5.20 大規模範例之單一溫度迭代次數參數設定比較.....	69
表 5.21 政策成本情境測試結果.....	70
表 5.22 耗電量情境測試結果.....	70

第一章 緒論

1.1 研究背景與動機

隨著消費習慣的改變，網路購物盛行帶來龐大的商機。市場研究機構 eMarketer 以亞太及歐美等數個主要的電商平台作為研究對象，針對零售電子商務未來趨勢提出《2017 全球電子商務平台》(Global eCommerce Platforms 2017) 研究報告，報告指出全球零售電子商務銷售額將從 2017 年的 2.29 兆美元，成長到 2021 年底的 4.479 兆美元，相當於零售總額的 16.1%。

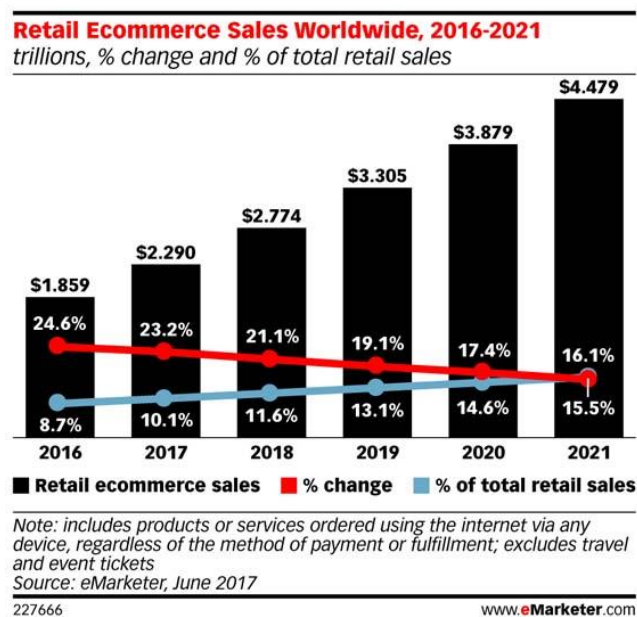


圖 1.1 全球零售電子商務銷售額

然消費者對網路購物的消費體驗亦有更高的期望，希望能更快速且準確地取得商品，2017 Yahoo 奇摩電商白皮書資料顯示，影響消費者選擇購物網站的前十大重要項目中「送貨速度」為第二名，換言之，隨著電子商務的蓬勃發展，同天點到點的物流需求(Same-day Peer to Peer Delivery)日益增長，時效性為一重要的競爭指標，其帶來的負面影響亦不容忽視，道路壅塞與效率降低、溫室氣體排放與土地擁擠、噪音與居民身體健康等生活品質，對經濟、環境及社會三大面向都造成極大的影響(Austria Tech, 2014)。

從 2010 年以來，共享經濟(Sharing Economy)的概念興起，全球網際網路市場興起一股新的商業模式。根據《經濟學人》定義共享經濟係「透過網路技術將閒置資源重新分配再利用。」物品的所有者可以透過平台將物品的使用權暫時轉讓給他人，進而提高社會資源的利用率，為平台兩端的參與者創造價值，發展至今，已有許多成功的應用。

共享經濟的概念亦被廣泛地應用在運輸領域中，造就了近年來相當蓬勃的發展—「共享移動系統(Shared Mobility Systems)」根據美國共享移動中心(Shared-Use Mobility Center, SUMC)提出，由於電子和無線技術的進步，使得共享資產變得更加容易執行，舉凡汽車製造商、租車公司、新創公司和城市贊助計劃皆如雨後春筍般地不斷湧出新方案。荷蘭於 1965 年首度提出自行車共享系統(Bike Sharing)後，共享自行車系統迅速拓展至世界各地，直到今日全球已有超過 1,000 個自行車共享系統，如台灣的 U Bike 及 O Bike，提供使用者利用佈署在整個系統的一組公共或私人自行車進行短途點對點運輸，帶來增加機動性、降低運輸成本及減少燃料使用的好處；汽車共享(Car Sharing)亦為另一發展趨勢，共享汽車分佈在大城市區域內的網路中，為會員提供短期使用汽車，通過預訂取得車輛，並按時間或距離收費，減少個人對汽車所有權的需求且無須負擔擁有私人汽車的成本，如德國 Car2Go、法國 Buzzcar 及美國 Zipcar 皆為成功的案例；此外，共乘(Ride Sharing; Ride Sourcing)的概念亦被廣泛地應用，核心概念為媒合擁有相同起迄點之乘客，充分利用既有旅次之車輛的閒置座位，提升平均車輛使用率並減少整理運輸系統路網中的車輛使用數，可以成為解決交通壅堵、排放和化石燃料依賴性問題的有效方法，如 Uber 及 Lyft 兩大享譽國際的企業。

近年來，亦有研究將共享經濟之概念應用在運輸與物流領域上，亦即針對人與貨物共同使用運輸資源(People and Freight Integrated Transportation, PFIT)此問題引入一個全新發展方向，結合乘客運輸以及貨物配送兩大路網，提出共享乘車問題(Share-a-Ride Problem, SARP)的模型(Li et al., 2014)，並將此模型應用在傳統計程車車隊的營運上，即計程車在服務乘客時亦利用車內閒置空間同時進行貨物或包裹的配送，每次最多針對一位乘客及一個貨物(包裹)作配送，且乘客有較高的優先性；此外，亦有研究進一步提出一般共享乘車問題(General Share-a-Ride Problem, G-SARP)，以共享計程車結合一般計程車，建構出一套可接受所有乘客運輸需求與貨物配送需求的最大化業者利潤模型(Yu et al., 2018)，在車輛容量限制之下，可針對多位乘客及多個貨物進行服務，且不考慮乘客的優先性。

然旨在藉由減少化石燃料消耗以達到環境、能源永續的綠色物流，亦是近年來引起關注的議題，應用在車輛途程問題 (Vehicle Routing Problem, VRP) 中演變為替代能源車輛途程問題 (Green Vehicle Routing Problem, G-VRP)，其特點是藉由實施有效率的車輛路徑及排程規劃，並考慮運輸過程中產生之燃料消耗成本或二氧化碳排放成本，此外，電動車車輛途程問題(Electric Vehicle Routing Problem, EVRP)亦為替代能源車輛途程問題的其中一個分支，提倡使用環保的燃料來源之車輛進行運輸及配送，如生物柴油、電力及乙醇等能源。加入以上考量後，使得替代能源車輛途程問題與傳統車輛途程問題相比會有不同的客戶指派、路徑選擇及車輛排程之結果(Lin et al., 2014)，而得以達到平衡環境和經濟成本的成效。

綜合以上所述，為了充分有效利用閒置之運輸資源，人跟貨物共享乘車的機制在未來有相當大的發展潛力，因此本研究針對一般共享乘車問題(G-SARP)，結合未來電動車發展與使用趨勢，探討混合計程車車隊之一般共享乘車問題(G-SARP with EVs, G-SARP-EVs)，此一混合計程車車隊包含電動車與傳統汽油車，然為求節能減碳，系統營運者將優先使用電動計程車進行服務，倘若現有電動計程車隊規模不足以服務所有顧客需求，才會使用汽油計程車進行服務。本研究以最佳化模型進行混合計程車車隊配送規劃之求解，不僅能最大化計程車業者利潤之效益，更期以能有效改善交通壅塞及環境汙染問題並解決運輸與物流產業面臨之難題，同時兼顧經濟、社會及環境三者之效益。

1.2 研究目的

將共享經濟的概念應用在運輸物流領域中，期以更有效率地利用閒置的運輸資源，並且以混合計程車車隊同時服務乘客及配送貨物，達到紓緩交通壅塞造成的社會成本及環境汙染等問題，並同時最大化計程車公司之利潤，創造雙贏的效益。本研究擬以建構時空網路模型及最佳化數學模式，進行混合計程車車隊及人貨需求之配對，並提供混合計程車車隊進行配送時的排程以及電動計程車之充電規劃，在完成所有服務需求前提下最大化計程車公司之運作效益。

1.3 研究範圍與對象

本研究選擇以混合計程車車隊作為人貨共同配送服務之運具，考量電動計程車之充耗電情況以及車輛容量限制，進行混合計程車與人貨需求之配對，並規劃每台計程車服務期間之車輛排程。

服務的對象有「乘客」及「貨物」兩大項，然於本研究中，僅針對可接受與其他乘客或貨物共乘之乘客，以及小型、常溫、不造成乘客困擾的貨物(包裹)作探討，並分別考量乘客與貨物的時窗限制，無考慮乘客與貨物間的優先性。

1.4 研究方法與流程

根據上述研究目的及研究範圍與對象，擬出本研究之研究流程，如圖 1.2 所示，針對各步驟說明如下：

1. 確立研究動機與目的

根據實務面的需求與未來發展趨勢，探討如何解決利用混合計程車隊進行人貨共同配送服務之車輛排程問題。

2. 訂定問題範圍與對象

確立本研究之動機及目的後，接著訂定本研究之問題範圍與對象，進一步將問題聚焦，並歸限出一明確的研究範圍。

3. 文獻回顧

根據本研究欲解決之問題，進行相關文獻的收集及探討，本研究針對三大方向進行文獻回顧，分別為「共乘問題」、「電動車應用在運輸與物流」以及「時空網路模型應用在運輸與物流」。

4. 建構 G-SARP-EVs 問題時空網路

根據本研究 G-SARP-EVs 問題之特性，建構人流、物流與車流之時空網路，車流進一步分成電動車時空網路及汽油車時空網路，用以有效地描述人流、物流與混合車流在時空維度上面之各種移動情形。

5. 建立數學模式

根據所建立之時空網路，定式 G-SARP-EVs 數學規劃模式以描述問題，以利進行後續之求解。

6. 建立啟發式演算法

根據所建立之時空網路，建立模擬退火法(Simulated Annealing, SA)，以利提升大規模範例之求解效率。

7. 範例測試與分析

以業者提供之實際營運資料，進行調整後產出不同規模之 G-SARP-EVs 測試範例，以最佳化軟體 Gurobi 進行 G-SARP-EVs 模式構建與求解；模擬退火法則以 Python 語言進行撰寫及求解，並針對其測試結果進行分析及探討。

8. 結論與建議

針對範例測試結果提出結論，並提出未來研究的具體建議。



圖 1.2 研究流程圖

第二章 文獻回顧

本研究係針對混合計程車車隊進行一般共乘問題之車輛排程模式，因此本章首先於 2.1 節針對一般共乘問題(G-SARP)之文獻進行回顧，並整理此問題之研究演進，其次於 2.2 節針對電動車應用在運輸與物流相關文獻進行回顧，探討電動車之發展趨勢及其優勢，接著 2.3 節針對時空網路模型應用在運輸與物流之相關文獻進行探討，作為本研究時空網路建構之參考，最後於 2.4 節作一文獻小結。

2.1 共乘問題介紹

1980 年撥召問題(Dial-a-ride Problem, DARP)被提出，其核心概念為合併考量多個乘客之運輸需求，直到 2014 年，結合乘客運輸及物流配送兩大網路之共乘問題(Share-a-ride Problem, SARP)被提出，以同一輛計程車同時進行乘客運輸及貨物配送之服務，然本研究欲解決之一般共乘問題(G-SARP)於 2018 年首度被提出，係以 SARP 為基礎並對該問題之假設一般化，使其在實務應用上可發揮更大的效益，本節接著則逐步針對 DARP、SARP 及 G-SARP 之演變過程作回顧。

2.1.1 撥召問題 (Dial-a-ride Problem, DARP)

Cordeau and Larporte (2013)針對撥召問題(DARP)之研究發展及演進進行整理，並定義本問題係滿足多個乘客的運輸需求(指定起點及目的地之接送請求)，對提供運輸服務之車輛進行其路線及排程規劃。通常，同一個乘客在同一天內會有兩個需求，從家裡到目的地(例如：醫院)的旅次需求，以及從目的地回到家裡的旅次需求，供給面則會針對單一場站、固定且已知車輛規模的單一車種進行求解，目標為規劃一組能夠滿足最多運輸需求之最低成本車輛路線，最常見之應用為針對高齡者及行動不便者提供的及門運輸(Door-to-door Transportation)。發展至今，有許多學者針對此問題作不同面向之研究，並發展出不同變化之模型，問題之目的通常會有兩大方向，一為滿足所有服務需求的前提之下最小化總營運成本，二是在車輛可及的前提之下最大化可服務之需求。

Psaraftis (1983)為撥召問題其中一個最簡單的例子，探討已知所有需求且僅使用一台車輛進行服務之情形，考量離開起點及抵達目的地之時窗限制以及車輛容量限制，目標為最小化總車輛行駛時間。

Jaw et al. (1986)為最早提出演算法來解決多車輛撥召問題的研究之一，需求為事先已知，並考量從家裡至目的地的出發時間與從目的地回家的抵達時間之時間窗限制，以及乘客於車內的搭乘時間限制，目標為最小化實際上下車時間與乘客預期時間之差異及實際搭乘所花費的時間，該演算法求解一實際的案例，其中包含 2617 個乘客及 28 台車輛。

Madsen et al. (1995)提出一演算法實際解決丹麥哥本哈根的高齡者及行動不便者之接送需求，考量每日動態出現之運輸需求及其相應之時窗限制，並以多種形態之車種提供服務，且車輛會有故障無法提供服務之情形發生，該演算法求解一含有 300 位乘客及 24 台車輛之實際問題。

2.1.2 共乘問題 (Share-a-ride Problem, SARP)

共乘問題(SARP)是由撥召問題(DARP)演變而來，原先撥召問題為將多個乘客運輸需求合併考量，期以找出系統中滿足所有需求之最小成本路徑，然共乘問題其核心概念與撥召問題相似，改成將人流與物流兩大網路合併考慮，期以利用車輛閒置空間同時對乘客及貨物進行服務，以達到減少交通壅塞等目標。

Li et al. (2014)提出以同一輛計程車同時進行乘客運輸及貨物配送之服務，以往針對乘客及貨物(包裹)通常分別對其進行運輸或配送服務，然此研究結合兩大系統，期藉由減少路網中的使用車輛數以達成舒緩交通壅塞之目的。此模型以計程車業者之角度出發，針對一個場站內有固定數量之有容量限制的計程車與多個乘客運輸及貨物配送需求進行求解，找出一組計程車之最佳排程規劃以最大化計程車業者之利潤，並有下列三點假設：(1)每輛計程車必須從場站出發，完成服務後必須回到場站、(2)一輛計程車一次只能服務一位乘客，且乘客相較於貨物擁有優先權、(3)每位乘客有最大搭乘時間的限制，司機亦有最長工時之上限，最後以最佳化求解軟體 Gurobi 進行小規模範例求解。

為使本問題應用至更大範圍之問題，Li et al. (2016)針對 SARP 問題，提出以適應性大規模鄰域搜尋演算法(Adaptive Large Neighborhood Search Heuristic, ALNS)進行大規模問題之求解，以貪婪插入演算法(Greedy Insertion Heuristic)建構起始解，並加以考量因配送貨物時，對乘客而言額外增加的旅行時間及收費折扣，以最大化計程車業者之利潤為目標，並在電腦硬體限制相同前提之下，針對 DARP 之基準測試例題進行求解，比對該演算法與一混合整數規劃模型搭配求解軟體之求解效果，證明此演算法於求解時間上及求解品質上皆有更佳之結果。

2.1.3 一般共乘問題 (General Share-a-ride Problem, G-SARP)

一般共乘問題(G-SARP)是由共乘問題(SARP)演變而來，其核心概念亦為結合人流與物流兩大網路，並以同一輛計程車同時進行乘客運輸與貨物配送之服務。Yu et al. (2018) 為使共乘問題更一般化，且使之在實務應用上可發揮更大的效益，提出一般共乘問題之模型，此模型亦針對一個場站內有固定數量之有容量限制的計程車與多個乘客運輸及貨物配送需求進行求解，找出一組計程車之最佳排程規劃滿足所有需求，並最大化計程車業者利潤，與 SARP 問題相同之假設為「每輛計程車必須從場站出發，完成服務後必須回到場站」不同的假設係放鬆乘客相較於貨物擁有優先權之限制，共有兩項：(1)不限制一輛計程車一次最多只能服務一位乘客，只要符合車輛的容量限制，可以同時對多位乘客或多個貨物進行服務、

(2)不考慮乘客之最大搭乘時間，僅須滿足需求之時窗限制，最後建構模擬退火法(Simulated Annealing, SA)進行求解，提升求解效率。

2.1.4 共乘問題演進比較

共乘問題(SARP)是由撥召問題(DARP)演變而來，其最大的差異係將以往分開探討之乘客運輸與貨物配送兩大路網作結合，然一般共乘問題(G-SARP)再由共乘問題(SARP)演變而來，與共乘問題不同之處為不考慮「乘客優先性」此假設，使共乘問題一般化，應用在實務上可發揮其更大的效益，然本研究欲探討的使用電動車之一般共乘問題(G-SARP-EV)係以 G-SARP 問題為基礎，結合未來電動車發展與使用趨勢，以電動計程車作為運輸與配送之工具，進行乘客運輸與貨物配送之服務，表 2.1 為本研究整理共乘問題演進比較表。

表 2.1 共乘問題演進比較表

	代表文獻	服務對象	需求	場站	車種	時間窗	車輛 容量	乘客最大 搭乘時間	目標式	建模及求解方法
撥召問題 (Dial-a-ride Problem)	Psaraftis (1983), Jaw et al. (1986), Madsen et al. (1995), Cordeau&Larporte (2013)	多個乘客	固定已知/ 動態出現	單一/ 多個	單一/多種	載客/卸客	有	有/無	最小化總營運成本/ 最小化車輛行駛距離	動態規劃法/插 入啟發式演算法
共乘問題 (Share-a-ride Problem)	Li et al. (2014), Li et al. (2016)	一個乘客及 一個貨物 (乘客有優 先權)	固定已知/ 動態出現 (不須滿足 所有需求)	單一	單一	載客(貨)及 卸客(貨)	有	有	最大化業者利潤	混合整數規劃/ 適應性大規模鄰 域搜尋演算法
一般共乘問題 (General Share-a-ride Problem)	Yu et al. (2018)	多個乘客及 多個貨物 (乘客無優 先權)	固定已知 (須滿足所 有需求)	單一	單一	載客(貨)及 卸客(貨)	有	無	最大化業者利潤	混合整數規劃及 模擬退火法
使用電動車之一般共乘問題 (General Share-a-ride Problem with Electric Vehicles)	本研究	多個乘客及 多個貨物 (乘客無優 先權)	固定已知 (須滿足所 有需求)	單一	混合 (充電式電 動車及傳統 汽油車)	載客(貨)及 卸客(貨)	有	無	最大化業者利潤	混合整數規劃及 模擬退火法

2.2 電動車應用在運輸與物流

綠色物流成為近年來供應鏈管理中的熱門議題，除以往運輸與配送領域重視之最有效率運輸及配送規劃的傳統目標外，更重視如何減少化石燃料消耗以達到環境及能源永續的車輛途程問題 (Vehicle Routing Problem, VRP)，替代能源車輛途程問題 (Green Vehicle Routing Problem, G-VRP) 開始引起大量關注及探討，其特點是藉由實施有效率的車輛路徑及排程規劃，並考慮運輸過程中產生之燃料消耗成本或二氧化碳排放成本，以達到平衡環境和經濟成本的成效。

然近年來，電動汽車(Electric Vehicles, EV)對全球的環境及金融亦有相當程度的影響，其優勢包括零氣體排放、較低的燃料成本及低噪音等，引起各界的關注及重視，應用於運輸與物流領域中係電動車輛途程問題(Electric Vehicle Routing Problem, EVRP)，亦為替代能源車輛途程問題的其中一個分支，提倡使用環保的燃料來源之車輛進行運輸及配送，如生物柴油、電力及乙醇等能源，然發展中的電動汽車亦有很多在應用上需優化之處，如電動車充電規劃、充電站設置等相關議題，因此，有不少研究於此進行探討。

Schneider et al. (2014)提出含時窗限制之電動車輛途程問題(Electric Vehicle Routing Problem with Time Windows, E-VRPTW)，考慮顧客之時窗限制、電動車充電時間、車輛剩餘電量以及車輛載運容量之限制，目標為最小化執行車輛數和車輛總旅行距離。研究結合變動鄰域搜尋法(Variable Neighborhood Search Algorithm, VNS)和禁忌搜尋法(Tabu Search, TS)發展一啟發式演算法 VNS/TS 求解問題。最後，以自行設計 E-VRPTW 問題之範例測試 VNS/TS 求解績效，結果顯示，VNS/TS 能於有效時間內求解小型範例之最佳解。

Barco et al. (2017)建立電動車輛途程問題(EVRP)探討電動車之路徑選擇及充電排程規劃，考量電動車輛之電池充電狀態、電池耗損、電池生命週期等因素，目標係最小化能源消耗成本，成本包括充電成本及車輛行駛所造成的電池損耗成本，並針對時間窗、車輛容量及剩餘電量作限制，並利用差分進化演算法(Differential Evolution)求解，再以基因演算法(GA)為基準以比較兩種演算法之求解績效。最後以哥倫比亞某機場至附近景點之接駁巴士為實際範例，探討在各種情況下之車輛充電排程結果。

2.3 時空網路模型應用在運輸與物流

時空網路模型(Time-space Network Model)係以路網流量模型建構欲解決的問題，傳統的時空網路模型為二維，將時間及空間以節點作表示，每個節點表示指定空間於指定時間下之狀態，並以節線將兩節點作連接，表示於兩個時間狀態下之空間節點之間移動的情形，與連結型網路流量模型(Connection-Based Flow Model)相比，其優勢在於可避免產生兩節點間的全部可能節線，僅針對問題可行

解範圍下之節線作呈現，減少欲解決之變數數量，可使時空網路模型應用至更大型的問題；時空網路模型亦可有多層的設計，應用在運輸與物流領域中，可用多層時空網路模型來表示多場站或多車輛的問題。

時空網路模型於 Hane et al. (1995) 首度被應用在航空排程(Airline Scheduling) 問題中，研究針對機隊指派問題(Fleet Assignment Problem)進行探討。機隊指派為航空公司面臨的主要問題之一，其主要目的為求解每個航段在考慮不同機型的不同性質與成本後，每個航段要指派何種機型的飛機，通常建立整數規劃模式進行求解，然由於大量的整數變數，使得此類問題之求解困難且耗時。本研究建構一時空網路模型以描述問題，再建立一整數規劃數學模式進行求解，計算結果顯示可得到與最佳解小於 0.02% 差距之解，且求解效率比原先使用之分枝界限法(Branch and Bound)快兩個數量級。

Kliwer et al. (2006) 為首篇將時空網路模型應用在公車排程問題之文獻，探討多場站多車種之公車車輛排程問題(Multiple-Depot Multiple-Vehicle-type Bus Scheduling Problem)，以多層時空網路模型建構車輛排程模式，每層時空網路代表一個場站，並透過 Latest-First-Matches 方法化簡大量的空渡節線，以車輛於同一場站之等待節線替代可能產生之空渡節線，並建構一混合整數規劃(Mixed Integer Programming, MIP)並使用套裝軟體(CPLEX 8.0)求得最佳解，接著再進一步針對車輛等待節線之變數，採用班次先進先出(FIFO)與班次後進先出(LIFO)之分解策略對各場站之最佳解進行勤務分解，重新求解多場站多車種車輛排程問題。

Mahmoudi & Zhou (2016) 探討車輛共乘服務系統最佳化問題，屬於考慮時窗限制與取貨送貨之車輛途程問題(VRPPDTW)，建構一三維時空網路模型描述時間、空間及車輛內乘客搭乘狀態(Passenger Carrying State)，用以表示乘客於不同時間、不同空間下是否位於車輛上，並以虛擬上、下車節點表達乘客之上車及下車時窗，目標為最小化車輛總旅行成本。本文獻利用拉式鬆弛法(Lagrangian relaxation approach)之演算法將多車輛途程問題分解成多個單一車輛途程問題，並以時間相依之動態規劃演算法求解放鬆後之單一車輛時間相依最小成本路徑問題。最後，以一中型規模及一大型規模之實際範例進行演算法之求解效率及求解品質驗證。

Yan & Chen (2011) 探討多對多起迄之小客車共乘配對問題，並考慮時窗限制、乘客類型、車輛類型、乘客要求、車輛容量限制等實際營運之相關限制，並以時空網路模型建構模式，目標為最小化共乘系統總成本，成本包含車輛旅行成本、乘客時間成本以及未被服務到之乘客的懲罰成本。為了能求解大型實際問題，發展一拉式鬆弛法(Lagrangian relaxation)及次梯度法(subgradient method)之演算法，以在合理時間內求解最佳解，透過電腦隨機產生不同規模之範例測試演算法之求解績效。

Yan et al. (2014)探討發生重大災害後補給車輛運送物料至各搶修地點之排程規劃，文獻利用時空網路流動方式表示車輛及補給物料之流動情況，並考量實際旅行時間之隨機性與其他營運限制因素，以最小化總營運成本為目標，建構隨機旅行時間下物料後勤補給排程模式。文獻以問題分解技術為基礎發展一啟發式演算法，其方法為將原問題根據時窗分解出一小型子問題，利用最佳化求解軟體先求解子問題，後固定第一個子問題之解，再繼續分解出下一子問題進行求解，直到滿足全部搶修地點之補給需求。最後，以台灣 1999 年集集大地震之災害數據進行測試，結果顯示考慮隨機性旅行時間之物料後勤補給排程模式在實際應用上有其實質的應用價值。

Lu et al. (2018)提出混合計程車隊排程問題(mixed taxi fleet scheduling problem, MTFSP)，計程車隊由充電式電動車輛與傳統汽油車輛組成，考量電動車之剩餘電量及其他限制條件，在滿足所有事先預約之確定性需求下，以混合車隊總營運成本最小化為目標。為了在合理時間範圍內求解大型範例，發展兩種啟發式解法，最後，依據台灣計程車隊之資料產生測試範例，測驗啟發式解法之求解績效，測試結果顯示，提出之兩種演算法時間分割法(network decomposition by time)與車輛分解法(network decomposition by vehicles)皆有良好之求解效率。

2.4 模擬退火法應用在運輸與物流

模擬退火法(Simulated Annealing, SA)係由 Kirkpatrick 等人於 1983 年提出，其想法源自於 Metropolis 等人於 1953 年提出之蒙地卡羅(Monte Carlo)演算法，至今已被廣泛應用在解決有一定難度之組合最佳化問題(combinatorial optimization problems)上，並獲得良好求解效果及效率。

Kim & Moon (2003)為解決船席排程問題(berth scheduling problem)，提出一混合整數線性規劃數學模式(Mixed integer linear programming)進行問題建構，並利用最佳化求解軟體 Lindo 進行問題求解，然受限於問題之難度及最佳化求解軟體之求解效能，當求解問題之船隻逐漸增加時，求解時間大幅上升，故研究亦根據問題特性提出模擬退火法進行求解，並將兩方法之求解結果進行比較，結果顯示模擬退火法可於短時間內求得與最佳解接近之結果。

Kuo (2010)提出一考慮時間相依之車輛途程問題(time-dependent vehicle routing problem, TDVRP)，考量隨著時間改變，不同路段車輛行駛速度及行駛時間之變化，然為了達到環保的目標，將目標式設為最小化車輛燃油消耗，而非傳統之最小化旅行時間或旅行距離，其中燃油消耗考慮車輛行駛距離、車輛行駛速度及車內載重。研究提出模擬退火法進行求解，結果顯示與傳統之最小化旅行時間目標相比，模擬退火法可成功求得減少 24.61% 燃油消耗的結果；若與最小化旅行距離目標相比，模擬退火法亦可成功求得減少 22.69% 燃油消耗的結果。

Lin et al. (2011)針對考量時窗限制之卡車與拖車途程問題(truck and trailer routing problem with time windows, TTRPTW)進行研究，並研擬模擬退火法進行求解，研究為驗證模擬退火法之求解效果進行兩組範例測試，範例根據前人文獻提供之標準範例，結果顯示模擬退火法可於合理時間內求得效果優良之結果。

Yu et al. (2015)提出開放式區位途程問題(open location routing problem, OLRP)，此問題為第三方物流(third-party logistic)公司興起後所衍生之新問題，最大的特性為完成服務後車輛不需回到場站，目標係最小化營運成本，包含設施營運成本、車輛固定成本及車輛旅行成本。研究利用最佳化求解軟體 CPLEX 及模擬退火法進行求解，並以標準範例進行測試，將兩方法之求解結果進行比較，結果顯示於小規模範例中，模擬退火法皆能求得比 CPLEX 成本更低之解，且求解時間更短；於大規模範例中，CPLEX 已無法求得可行解，然模擬退火法仍可在合理時間內求得一可行解。

Yu et al. (2018)針對一般共乘問題(general share-a-ride problem, G-SARP)進行研究，除建構數學模式進行問題描述及小規模範例求解外，亦提出三種演算法進行求解，包含基本模擬退火法(basic SA)、進階模擬退火法(proposed SA)及禁忌搜尋法(tabu search)，其中兩種模擬退火法之主要差別為前者於搜尋解的過程中允許不可行解之存在，然後者則於方法中設計一機制將不可行解修正為可行解，結果顯示進階模擬退火法(proposed SA)於大規模範例中表現最佳，兼顧求解效果及效率。

2.5 小結

根據文獻回顧，本研究認為 G-SARP 為一極具潛力之發展方向，將共享經濟的概念應用在運輸物流領域中，期以減少路網中使用車輛數之方法來達到舒緩交通壅塞之目的，更有效率地利用閒置的運輸資源，達到舒緩社會外部成本及環境汙染等問題，再結合電動車之未來發展趨勢，發展一 G-SARP-EVs 問題，於實務上確有其應用價值。研究方法則以時空網路模型清楚地呈現貨物、乘客與兩種車種在時空維度上面之流動情形，並提出數學模式有效地建構並求解問題，為使本研究可應用至大範圍之實際案例，更可建構模擬退火法進行問題求解，係為一值得深入探討之研究。

第三章 研究方法

本章針對混合計程車隊進行人貨共同配送服務之車輛排程問題作詳細說明，並建構時空網路及定式 G-SARP-EVs 數學規劃模式，以利後續之求解。本章共有四個小節，首先 3.1 節為問題描述，針對業者營運方式及研究假設作說明；3.2 節根據 G-SARP-EVs 問題之特性，建構人流、物流與車流之時空網路，用以有效地描述人流、物流與車流在時空維度上面之各種移動情形；3.3 節根據所建構之時空網路，提出一混合整數線性規劃數學模式進行 G-SARP-EVs 問題之描述；最後 3.4 節進行小範例測試，驗證模式之正確性。

3.1 問題描述

共享乘車問題(Share-a-Ride Problem, SARP)的模型(Li et al., 2014)結合了乘客運輸以及貨物配送兩大路網，應用在傳統計程車車隊的營運上，使計程車在服務乘客時得以利用車內閒置空間，同時進行貨物或包裹的配送，每次最多針對一位乘客與一個貨物作配送，並考量乘客之優先性(亦即會先完成乘客的運輸，將乘客送抵其迄點後，再將貨物送抵其迄點)。以圖 3.1 為例，服務車輛 1(服務車輛 2)會先抵達 A 乘客(C 乘客)的起點進行載客服務，再前往 B 貨物(D 貨物)的起點進行載貨服務，接著前往 A 乘客(C 乘客)的迄點進行卸客服務，最後前往 B 貨物(D 貨物)的迄點進行卸貨，即算是完成一趟完整的服務。

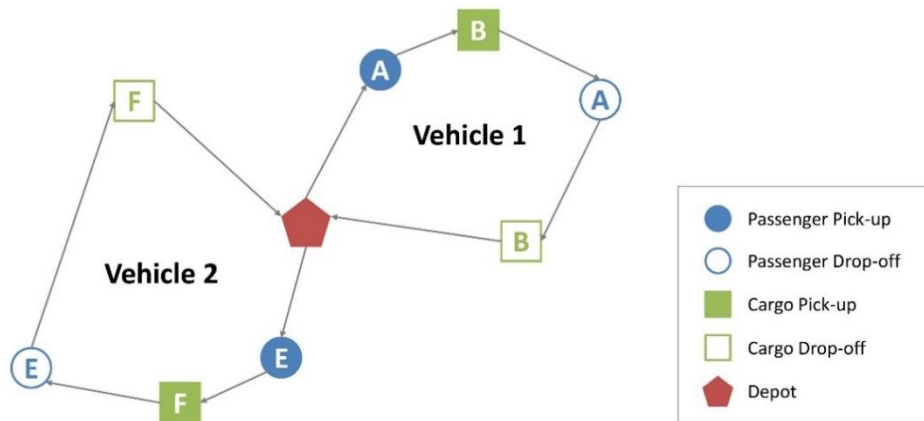


圖 3.1 共乘問題(SARP)

一般共享乘車問題(General Share-a-Ride Problem, G-SARP)，則是建立在 SARP 問題的基礎上，建構出一套可接受所有乘客運輸需求與貨物配送需求的最大化業者利潤模型(Yu et al., 2018)，在滿足車輛容量限制及需求時窗限制之下，可同時進行多位乘客與多個貨物的運輸及配送服務，且不考慮乘客之優先性。以圖 3.2 為例，服務車輛 1 先抵達 A 乘客的起點進行載客服務，再前往 B 貨物的

起點進行載貨服務，接著前往 A 乘客的迄點進行卸客服務，最後前往 B 貨物的迄點進行卸貨；而服務車輛 2，先抵達 E 乘客的起點進行載客服務，再前往 F 貨物的起點進行載貨服務，接著前往 E 乘客的迄點進行卸客服務，再前往 G 貨物的起點進行載貨服務，前往 F 貨物迄點進行卸貨服務後，最後前往 G 貨物的迄點進行卸貨。

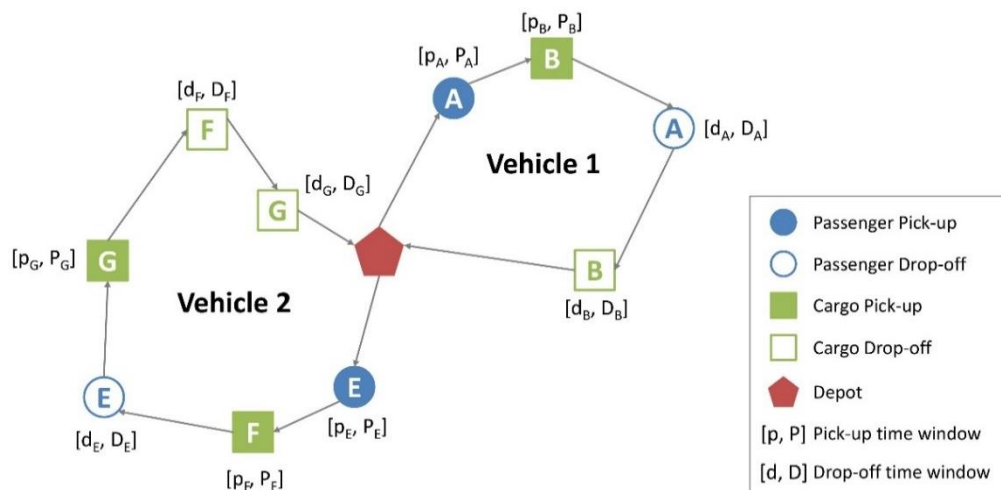


圖 3.2 一般共乘問題(G-SARP)

本研究針對 G-SARP 問題進行延伸，以充電式(Plug-in)電動計程車及汽油計程車之混合車隊進行運輸與配送服務，並以計程車營運業者的角度出發，於特定範圍內進行服務，考量當日營運時間內混合計程車隊與人貨配送需求之配對以及車輛排程與電動計程車充電之規劃，以營運業者利潤最大化為目標，在固定且已知的車隊數量及車輛容量限制下，僅針對當日營運時間前完成預約之乘客及貨物需求進行服務，建構出一確定性混合計程車隊排程模式，針對此問題，本研究分成供給及需求兩大構面，提出以下六點假設及其說明：

1. 供給面

(1) 單一場站、混合車種及固定數量車輛

針對單一場站之混合電動車與汽油車兩類車種之計程車業者進行探討，每台計程車進行當日第一個服務前須從場站出發，結束當日最後一個服務後須回到場站；使用電動車種部份，僅針對充電式電動車(Plug-in EVs)及一般汽油車進行探討，且假設業者為方便管理只購買單一電動車種，亦即每台車輛的充電與耗電特性為一致，此外，亦可將本研究之模式作適度修改，應用至使用交換式電動車(Battery Swapping EVs)的 G-SARP 問題，本文以下提到之電動車或電動計程車皆為充電式電動車；最後，本研究假設業者以固定的車隊規模進行服務，亦即系統已知可提供服務之電動計程車輛的數量，並搭配汽油車服務電動車輛無法滿足之需求。

(2)車輛容量及電量、耗油限制

探討計程車輛之容量限制，在其車輛容量限制之下，可以針對多位乘客及多個貨物提供服務；考慮充電式電動車輛之充耗電狀況，在每日營運前，每輛充電式電動計程車皆為飽電狀態，營運期間若是電量過低而無法進行下一個運輸及配送服務，則該充電式電動計程車需進行充電而暫時停止提供運輸及配送服務，待充飽電後才可繼續接下來的服務；在每日營運前，每輛汽油計程車皆為加滿油之狀態，並假設汽油車無加油之需求。

(3)旅行時間及成本已知且固定

車輛行經各站點之旅行時間及成本為已知且固定。

2. 需求面

(1)需求固定且已知

假設系統於規劃時間軸中每一時段之服務需求已知且固定，包含乘客與貨物的起迄點及時窗限制(含最早可以接受服務的時間區間及最晚必須完成服務的時間區間)。實務上，亦即僅針對當日營運時間前完成預約之乘客需求與貨物需求提供服務，假設乘客皆能接受與其他乘客及貨物共乘才進行預約，且亦接受乘客攜帶事先告知且起迄點相同之貨物；貨物則是針對小型、常溫、不造成其他乘客困擾的貨物(包裹)作配送，且不針對即時且隨機出現的需求提供服務。

(2)無乘客及貨物需求優先權

本研究針對一般化共享乘車問題，不探討乘客及貨物間的優先順序，只要滿足乘客及貨物的時窗限制，皆可進行服務。

(3)所有需求皆須被滿足

本研究假設所有需求皆須被滿足，以電動計程車隊搭配汽油計程車隊進行模式的建構。以實務上來說，為符合未來電動車發展之趨勢及滿足當日營運時間前完成預約之需求服務，優先使用電動計程車滿足當日需求，若現有規模無法滿足所有需求，則改用汽油計程車提供服務，同時兼顧環保及業者服務可靠度。

3.2 時空網路模型

本研究利用時空網路描述人流、物流與車流在時空維度中之移動情形，時空網路橫軸為空間維度，代表乘客與貨物上下車(貨)之站點或電動車充電站；縱軸為時間維度，代表系統在不同時間點之狀態，未來使用此模式時，可依不同需求及規劃時程考量，於本時空網路作相應的修改，彈性應用至不同情境的問題。

本研究將時空網路分成需求面時空網路及供給面時空網路，需求面時空網路僅一層，用以描述乘客與貨物的流動情形；供給面時空網路包含汽油車時空網路及電動車時空網路，分別描述汽油車及電動車之流動情形，兩者皆為多層時空網路之設計，以車輛作為分層依據以方便進行車輛排程之描述及電動計程車電量之追蹤，供給面時空網路的總層數即業者所擁有的車隊規模。本節以下依序針對需求面時空網路、汽油車時空網路、電動車時空網路進行詳細介紹。

3.2.1 需求面時空網路

本研究建構之需求面時空網路可參見圖 3.3，其中有三種節點，分別為「運輸節點」、「虛擬起點節點」、「虛擬迄點節點」，以下分項說明：

1. 運輸節點 (Location Node)

為乘客運輸及貨物配送之起迄站點，針對每個需求皆有一個起點運輸節點及一個迄點運輸節點，然在每個時點下，皆有一相應之時空節點 (Location-Time Node)，用以代表不同時間狀態下之運輸節點的狀態。

2. 虛擬起點節點 (Dummy Pickup Node)

針對系統中每個已知需求，皆有其相應的虛擬起點節點，對應至該需求的起點時間窗，亦即可接受載客(貨)服務的時間區間，如圖 3.3 中的乘客(Passenger1)，該乘客可接受載客之服務時間為上午 9:15 至 9:30 間，則該乘客會有一個相應的虛擬起點節點(圖中藍色圓點)，並以載客(貨)節線與相應之時空節點作連結，作為系統中進行載客(貨)服務的表示方式。

3. 虛擬迄點節點 (Dummy Delivery Node)

針對系統中每個已知需求，亦有相應的虛擬迄點節點，對應至該需求的迄點時間窗，亦即可接受卸客(貨)服務的時間區間，如圖 3.3 中的乘客(Passenger1)，該乘客可接受卸客之時間為上午 10:15 至 10:45 間，則該乘客即有兩個相應的虛擬迄點節點(圖中藍色中空圓點)，並以卸客(貨)節線與相應之時空節點(Location-Time Node)作連結，作為系統中進行卸客(貨)服務需求的表示方式。虛擬迄點節點與虛擬起點節點為相同設計邏輯，此設計使時空網路模型得以應用在 G-SARP 的問題上，並讓此問題在時空網路模型上進行求解。

需求面時空網路節線共兩種，係「載客(貨)節線」與「卸客(貨)節線」以下分項說明：

1. 載客(貨)節線 (Pick-up Arc)

載客(貨)節線為連結運輸節點(上游點)與虛擬起點節點(下游點)的節

線，表示進行該相應需求之載客(貨)服務。載客(貨)節線之流量上限為 1，表示該相應需求有進行載客(貨)服務；流量下限為 0，表示該相應需求沒有進行載客(貨)服務；該節線有營收，亦即業者進行該需求之運輸或配送服務時可收取之營收。

2. 卸客(貨)節線 (Drop-off Arc)

卸客(貨)需求節線為連結運輸節點(上游點)與虛擬迄點節點(下游點)的節線，表示該相應需求之卸客(貨)服務。卸客(貨)節線之流量上限為 1，表示該相應需求有進行卸客(貨)服務；流量下限為 0，表示該相應需求沒有進行卸客(貨)服務。

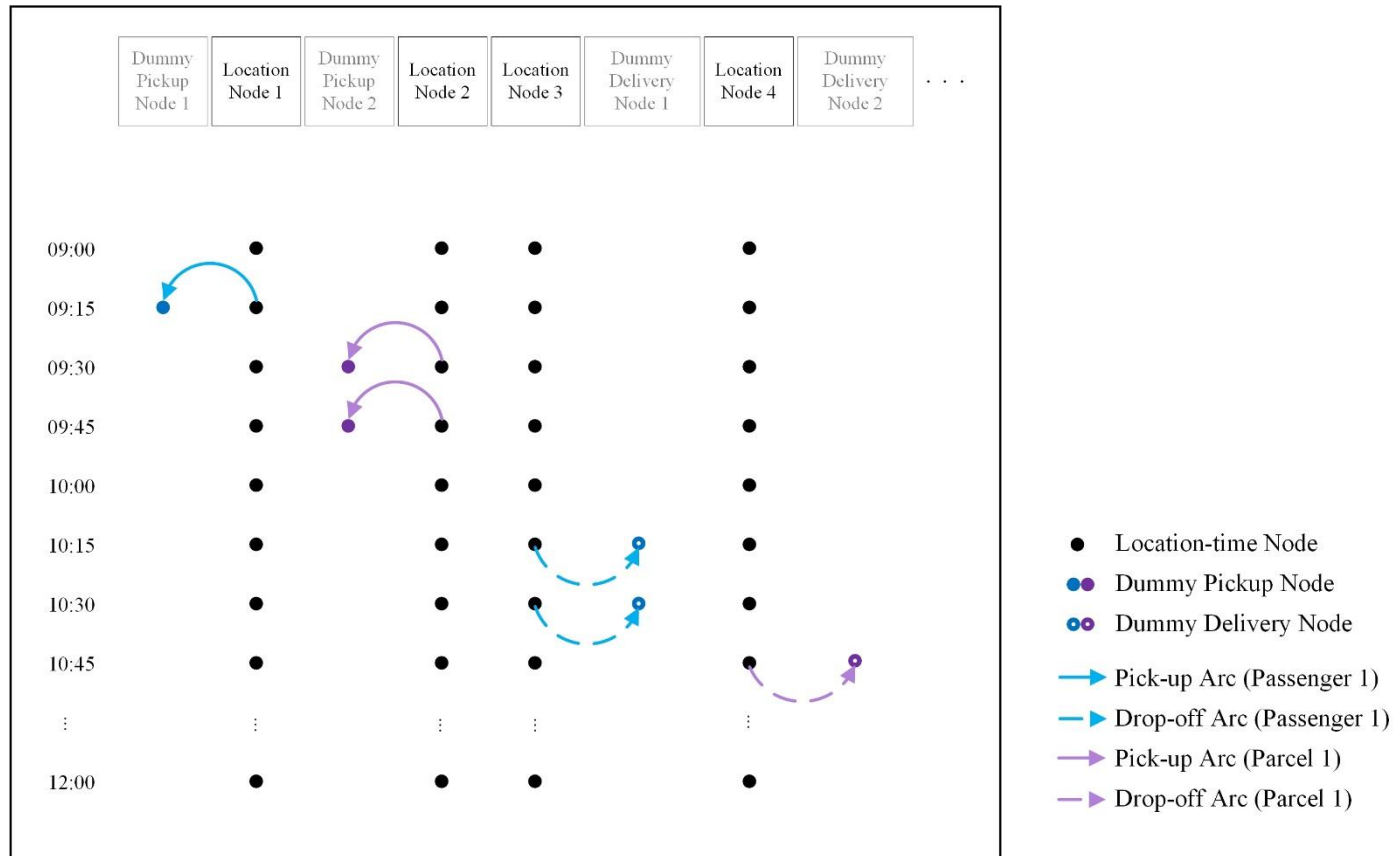


圖 3.3 需求面時空網路示意圖

3.2.2 供給面 – 汽油車時空網路

本研究建構之汽油車時空網路可參見圖 3.4，有五種節點設計，依序為「出發節點」、「匯集節點」、「運輸節點」、「虛擬起點節點」及「虛擬迄點節點」，其中運輸節點、虛擬起點節點及虛擬迄點節點與需求面時空網路設計相同不多加贅述外，以下分項說明：

1. 出發節點 (Source Node)

每層汽油車時空網路中會有一個出發節點，作為該層時空網路之汽油計程車執行服務前的虛擬出發場站，出發節點與各需求之起點運輸節點作連結，連結之節線即供給節線，代表本問題不限制計程車一定要在規劃時軸的第一個營運時間就離開場站，只須於進行服務前前往欲載客(貨)的站點。

2. 匯集節點 (Sink Node)

每層汽油車時空網路中亦會有一個匯集節點，該層時空網路之汽油計程車執行服務後必須回到此虛擬場站，各需求之迄點運輸節點亦與匯集節點作連結，連結之節線即匯集節線，代表本問題不限制計程車一定要在規劃時軸的最後一個營運時間才回到場站，只要完成所有指派之任務後即可駛回場站。

汽油車時空網路的節線設計共八種，分成「供給節線」、「匯集節線」、「旅行節線」、「停等節線」、「未用車節線」、「載客(貨)節線」、「卸客(貨)節線」及「虛擬節線」，除「載客(貨)節線」與「卸客(貨)節線」與需求面時空網路相同而不多加贅述外，其他以下分項說明：

1. 供給節線 (Supply Arc)

供給節線為連結出發節點與各需求起點運輸節點之節線，表示該層計程車於場站出發至欲進行服務之實際運輸節點的移動，本研究之計程車只須於進行服務前前往欲載客(貨)的站點，不限制計程車一定要在規劃時軸的第一個時點就離開場站。每層汽油車時空網路的供給節線之流量上限為 1，流量下限為 0；為優先使用電動計程車，汽油車之供給成本除使用汽油計程車之耗油成本外，亦須加上政策成本，使得選擇使用汽油車之成本大於選擇使用電動車。

2. 匯集節線 (Merge Arc)

匯集節線為連結各需求迄點運輸節點與匯集節點之節線，表示該層計程車完成所有指派之任務後回到場站的移動，本研究不限制計程車一定要在規劃時軸的最後一個時點才回到場站，只要完成當日服務之電動計程

車皆可隨時駛回場站。每層時空網路的匯集節線之流量上限為 1，流量下限為 0；節線成本為汽油車耗油成本。

3. 旅行節線 (Travel Arc)

旅行節線係連結不同時間狀態下兩運輸節點的節線，表示該層時空網路之汽油計程車在各實際運輸節點間的移動情形。該層汽油車時空網路的旅行節線之流量若為 1，表示該時段兩節點間有計程車經過，流量為 0 則表示該時段兩節點間無計程車經過，節線成本為汽油計程車的耗油成本加上政策成本，耗油成本與實際站點間之旅行距離成正比，政策成本則為使選擇汽油車之成本大於電動車。

4. 停等節線 (Waiting Arc)

停等節線為連結不同時間狀態下同一運輸節點的節線，表示汽油計程車留在同一場站進行停等的情形，若該層時空網路之停等節線流量為 1，表示該車輛於該運輸節點有停等之行為，節線成本為一定值，與停等時間成正比。

5. 未用車節線 (Unused-taxi Arc)

未用車節線為連結出發節點及匯集節點之節線，表示汽油計程車留等在場站的情形。該層時空網路的未用車節線之流量若為 1 表示計程車無移動行為，沒有提供運輸或配送之服務；流量為 0 則表示計程車有進行運輸或配送之服務，並非停留在場站；節線流量成本為 0。

6. 虛擬節線 (Virtual Arc)

虛擬節線有兩種，第一種是「載客(貨)虛擬節線(Pick-up Virtual Arc)」連結虛擬起點節點(上游點)與運輸節點(下游點)；第二種是「卸客(貨)虛擬節線(Drop-off Virtual Arc)」為連結虛擬迄點節點(上游點)與運輸節點(下游點)，兩種虛擬節線為滿足網路節點流量守恆之設計，並用以表示車輛於時空網路模型中進行載客(貨)服務或卸客(貨)之後可以繼續進行下一服務之狀態。虛擬節線之流量上限為 1，流量下限為 0，若其相應之載客(貨)節線或卸客(貨)節線流量為 1，則該虛擬節線的流量也必為 1，反之則必為 0；由於此虛擬節線並非表示實際的車輛流動情形，故其節線流量成本為 0。

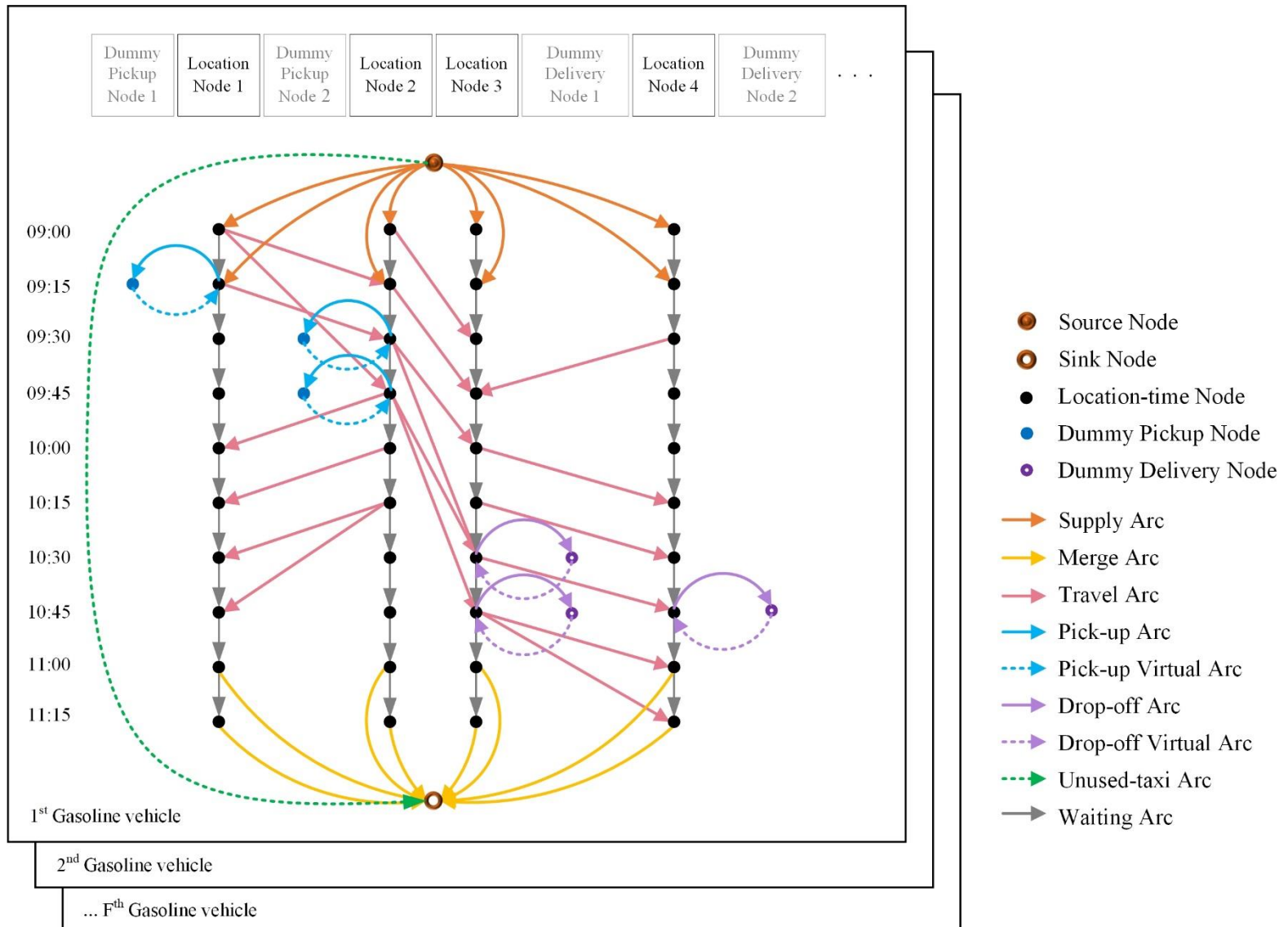


圖 3.4 汽油車時空網路示意圖

3.2.3 供給面－電動車時空網路

本研究建構之電動車時空網路可參見圖 3.5，有六種節點設計，依序為「出發節點」、「匯集節點」、「運輸節點」、「虛擬起點節點」、「虛擬迄點節點」及「充電節點」，其中運輸節點、虛擬起點節點及虛擬迄點節點與需求面時空網路設計相同不多加贅述外，以下分項說明：

1. 出發節點 (Source Node)

每層電動車時空網路中會有一個出發節點，作為該層時空網路之電動計程車執行服務前的虛擬出發場站，出發節點與各需求之起點運輸節點作連結，連結之節線即供給節線，代表本問題不限制計程車一定要在規劃時軸的第一個營運時間就離開場站，只須於進行服務前前往欲載客(貨)的站點。

2. 匯集節點 (Sink Node)

每層電動車時空網路中亦會有一個匯集節點，該層時空網路之電動計程車執行服務後必須回到此虛擬場站，各需求之迄點運輸節點亦與匯集節點作連結，連結之節線即匯集節線，代表本問題不限制計程車一定要在規劃時軸的最後一個營運時間才回到場站，只要完成所有指派之任務後即可駛回場站。

3. 充電節點 (Charging Node)

作為營運範圍內充電式電動計程車可進行充電之場站，倘若充電式電動計程車之電量低至無法完成下一個運輸或配送服務時，則須進入充電場站進行充電，待充飽電後才可繼續接下來之運輸及配送服務。

電動車時空網路中節線共九種，分成「供給節線」、「匯集節線」、「旅行節線」、「停等節線」、「未用車節線」、「載客(貨)節線」、「卸客(貨)節線」、「虛擬節線」及「充電節線」，除「載客(貨)節線」與「卸客(貨)節線」與需求面時空網路相同而不多加贅述外，以下分項說明：

1. 供給節線 (Supply Arc)

供給節線為連結出發節點與各需求起點運輸節點之節線，表示電動計程車於場站出發至欲進行服務之運輸節點的移動，本研究之電動計程車僅須於進行服務前前往欲載客(貨)的站點，不限制電動計程車一定要在規劃時軸的第一個時點就離開場站。每層時空網路的供給節線之流量上限為 1，流量下限為 0；節線成本為電動計程車的折舊與維修成本，耗電量與旅行距離成正比。

2. 匯集節線 (Merge Arc)

匯集節線為連結各需求迄點運輸節點與匯集節點之節線，表示電動計程車進行規劃時軸內所有乘客運輸或貨物配送服務後回到場站的移動，本研究不限制電動計程車一定要在規劃時軸的最後一個時點才回到場站，只要完成所有指派之服務後即可駛回場站。每層時空網路的匯集節線之流量上限為 1，流量下限為 0；節線成本為電動車之耗電成本，與站點間之旅行距離正成本，耗電量為正值，亦與站點間之旅行距離成正比。

3. 旅行節線 (Travel Arc)

與汽油車時空網路相比，除原先連結不同時間狀態下兩運輸節點用以表示電動計程車在各運輸節點間的移動情形之旅行節線外，仍有「充電旅行節線(Charging Travel Arc)」係連結不同時間狀態下之運輸節點與充電節點的節線，表示電動計程車從運輸節點前往充電站進行充電或是在充電站充電完畢後回到運輸節點繼續進行服務之移動情形。每層時空網路的旅行節線之流量上限為 1，表示該時段兩節點間有電動計程車經過，反之則為 0，節線成本為電動計程車的耗電成本，耗電量為正值，與站點間之旅行距離成正比。

4. 停等節線 (Waiting Arc)

停等節線為連結不同時間狀態下同一運輸節點的節線，表示電動計程車留在同一場站進行停等的情形，為使車輛不會發生無故於運輸場站進行停等之情形，故停等節線之成本及耗電量皆設為一定值，與停等時間成正比。

5. 未用車節線 (Unused-taxi Arc)

未用車節線為連結出發節點及匯集節點之節線，表示當日電動計程車留在場站的情形。該層時空網路的未用車節線之流量若為 1，表示該電動計程車於規劃時軸內無移動行為，沒有提供運輸或配送之服務；流量若為 0 則表示該電動計程車有進行運輸或配送之服務，並非停留在場站，節線成本與耗電量皆為 0。

6. 虛擬節線 (Virtual Arc)

虛擬節線分成兩種，第一種是「載客(貨)虛擬節線(Pick-up Virtual Arc)」連結虛擬起點節點(上游點)與運輸節點(下游點)；第二種是「卸客(貨)虛擬節線(Drop-off Virtual Arc)」連結虛擬迄點節點(上游點)與運輸節點(下游點)的虛擬節線，兩種虛擬節線用以表示車輛於時空網路模型中進行載客(貨)服務或卸客(貨)之後可以繼續進行下一服務之狀態。虛擬節線之流量上限為 1，流量下限為 0，若其相應之載客(貨)節線或卸客(貨)

節線流量為 1，則該虛擬旅行節線的流量也必為 1，反之則必為 0；由於虛擬節線之設計僅為滿足網路中節點流量守恆之特性，並非表示實際的車輛流動情形，故其節線成本與耗電量皆為 0。

7. 充電節線 (Charging Arc)

充電節線為連結不同時間狀態下之充電節點，表示電動計程車進行充電之情形。每層時空網路的充電節線流量上限為 1，表示該時段電動計程車進行充電中；流量下限為 0，表示表示該時段無電動計程車進行充電；為使電動車僅於需要充電時才至充電站進行充電，而不會無故占用充電站進行停等之行為，故充電節線之成本設為一定值，並與充電時間成正比；在充電過程中，電動車有電量的增加，因此耗電量為負值，與充電時間成正本。

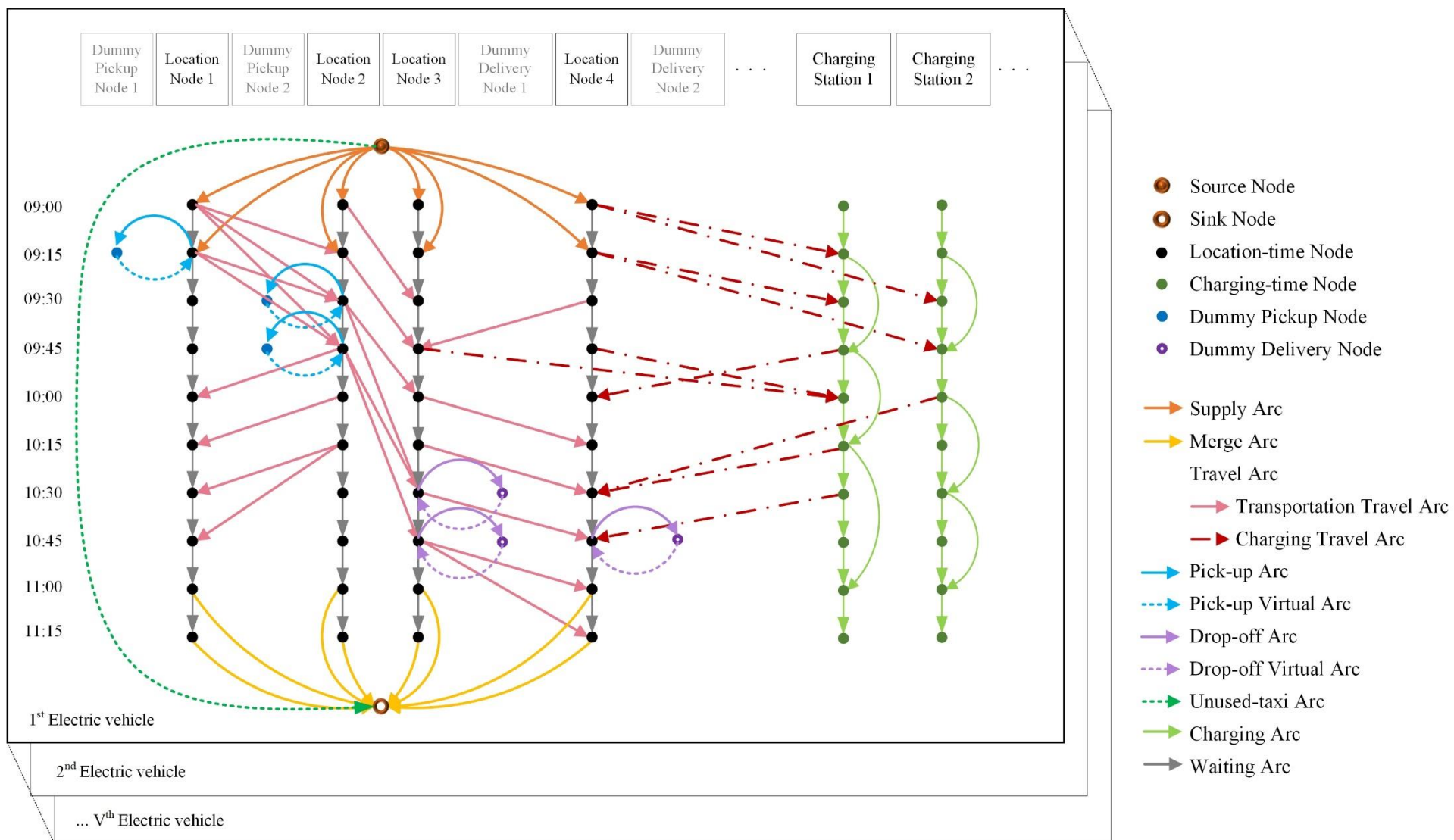


圖 3.5 電動車時空網路示意圖

3.3 使用電動車之一般共乘問題最佳化模式

本研究利用時空網路描述人流、物流與車流在時空維度上面的移動情形，並根據所建構之時空網路定式 G-SARP-EVs 數學規劃模式，求解目標為最大化計程車業者利潤，利潤計算方式為業者當日總營收減去當日總營運成本，總營收為所有載客(貨)節線的服務營收，總成本包括電動計程車供給成本(折舊與維修成本)、電動計程車的旅行成本(車輛於運輸及配送之旅行距離的耗電成本)、汽油計程車供給成本及汽油計程車的旅行成本(車輛於運輸及配送之旅行距離的耗油成本加上政策成本)，以下針對數學模式作詳細介紹。

集合

Q	需求集合
V	電動車輛集合，車輛數量亦即電動車時空網路的層數
N^v	第 v 層電動車時空網路之所有節點集合
K^v	第 v 層電動車時空網路之充電節點集合
O_q^v	第 v 層電動車時空網路之需求 q 的起點運輸節點集合
G_q^v	第 v 層電動車時空網路之需求 q 的迄點運輸節點集合
A^v	第 v 層電動車時空網路之所有節線集合
S^v	第 v 層電動車時空網路之供給節線集合
M^v	第 v 層電動車時空網路之匯集節線集合
T^v	第 v 層電動車時空網路之旅行節線集合
W^v	第 v 層電動車時空網路之停等節線集合
U^v	第 v 層電動車時空網路之未用車節線集合
C^v	第 v 層電動車時空網路之充電節線集合
P_q^v	第 v 層電動車時空網路之需求 q 載客(貨)節線集合
D_q^v	第 v 層電動車時空網路之需求 q 卸客(貨)節線集合
H_q^v	第 v 層電動車時空網路之需求 q 載客(貨)及卸客(貨)虛擬節線集合
F	汽油車輛集合，車輛數量亦即汽油車時空網路的層數
N^f	第 f 層汽油車時空網路之所有節點集合
O_q^f	第 f 層汽油車時空網路之需求 q 的起點運輸節點集合
G_q^f	第 f 層汽油車時空網路之需求 q 的迄點運輸節點集合
A^f	第 f 層汽油車時空網路之所有節線集合
S^f	第 f 層汽油車時空網路之供給節線集合
M^f	第 f 層汽油車時空網路之匯集節線集合
T^f	第 f 層汽油車時空網路之旅行節線集合
W^f	第 f 層汽油車時空網路之停等節線集合
U^f	第 f 層汽油車時空網路之未用車節線集合
P_q^f	第 f 層汽油車時空網路之需求 q 載客(貨)節線集合

D_q^f	第 f 層汽油車時空網路之需求 q 卸客(貨)節線集合
H_q^f	第 f 層汽油車時空網路之需求 q 載客(貨)及卸客(貨)虛擬節線集合

參數

c_{ij}^v	第 v 層電動車時空網路中節線 (i, j) 之成本
r_{ij}^v	第 v 層電動車時空網路中節線 (i, j) 之營收
e_{ij}^v	第 v 層電動車時空網路中節線 (i, j) 之耗電量
u^{max}	電動車剩餘電量比例之預設上限值
cap^{max}	電動車輛之容量上限
g_{ij}^f	第 f 層汽油車時空網路中節線 (i, j) 之成本
n_{ij}^f	第 f 層汽油車時空網路中節線 (i, j) 之營收
p_i	於時空網路中 i 點進行載/卸乘客之數量，正值表示載客，負值表示卸客； 若乘客攜帶貨物，則將貨物數以乘客當量表示，並於模式中將之視為乘客
m_i	於時空網路中 i 點進行載/卸貨物之數量，將貨物數以乘客當量表示
M	正數極大值

變數

x_{ij}^v	二元變數，表示第 v 層電動車時空網路中節線 (i, j) 之流量
z_{ij}^f	二元變數，表示第 f 層汽油車時空網路中節線 (i, j) 之流量
y_i^v	二元變數，為 1 表示時空網路中 i 點之需求被第 v 層之電動車服務，反之則為 0
y_i^f	二元變數，為 1 表示時空網路中 i 點之需求被第 f 層之汽油車服務，反之則為 0
u_i^v	連續變數，表示第 v 層電動車時空網路中車輛抵達 i 點之剩餘電量佔電池總容量之比例
p_i^v	連續變數，表示第 v 層電動車時空網路中車輛抵達 i 點進行服務前之車內乘客數
m_i^v	連續變數，表示第 v 層電動車時空網路中車輛抵達 i 點進行服務前之車內貨物數
p_i^f	連續變數，表示第 f 層汽油車時空網路中車輛抵達 i 點進行服務前之車內乘客數
m_i^f	連續變數，表示第 f 層汽油車時空網路中車輛抵達 i 點進行服務前之車內貨物數

數學模式

Maximize

$$\sum_{v \in V} \sum_{q \in Q} \sum_{(i,j) \in P_q^v} r_{ij}^v x_{ij}^v + \sum_{f \in F} \sum_{q \in Q} \sum_{(i,j) \in P_q^f} n_{ij}^f z_{ij}^f - \sum_{v \in V} \sum_{(i,j) \in A^v} c_{ij}^v x_{ij}^v - \sum_{f \in F} \sum_{(i,j) \in A^f} g_{ij}^f z_{ij}^f \quad (1)$$

Subject to

$$\sum_{(i,j) \in S^v \& U^v} x_{ij}^v = 1 \quad \forall v \in V \quad (2)$$

$\sum_{(i,j) \in M^v \& U^v} x_{ij}^v = 1$	$\forall v \in V$	(3)
$\sum_{(i,j) \in S^f \& U^f} z_{ij}^f = 1$	$\forall f \in F$	(4)
$\sum_{(i,j) \in M^f \& U^f} z_{ij}^f = 1$	$\forall f \in F$	(5)
$\sum_{v \in V} \sum_{(i,j) \in P_q^v} x_{ij}^v + \sum_{f \in F} \sum_{(i,j) \in P_q^f} z_{ij}^f = 1$	$\forall q \in Q$	(6)
$\sum_{v \in V} \sum_{(i,j) \in D_q^v} x_{ij}^v + \sum_{f \in F} \sum_{(i,j) \in D_q^f} z_{ij}^f = 1$	$\forall q \in Q$	(7)
$\sum_{(i,j) \in P_q^v} x_{ij}^v \geq \sum_{(i,j) \in D_q^v} x_{ij}^v$	$\forall q \in Q, \forall v \in V$	(8)
$\sum_{(i,j) \in P_q^f} z_{ij}^f \geq \sum_{(i,j) \in D_q^f} z_{ij}^f$	$\forall q \in Q, \forall f \in F$	(9)
$\sum_{j \in N^v} x_{ij}^v - \sum_{k \in N^v} x_{ki}^v = 0$	$\forall i \in N^v, \forall v \in V$	(10)
$\sum_{j \in N^f} z_{ij}^f - \sum_{k \in N^f} z_{ki}^f = 0$	$\forall i \in N^f, \forall f \in F$	(11)
$\sum_{j \in N^v} x_{ij}^v + \sum_{k \in N^v} x_{ki}^v = 4 - 4(1 - y_i^v)$	$\forall i \in O_q^v \cup G_q^v, \forall q \in Q, \forall v \in V$	(12)
$\sum_{j \in N^f} z_{ij}^f + \sum_{k \in N^f} z_{ki}^f = 4 - 4(1 - y_i^f)$	$\forall i \in O_q^f \cup G_q^f, \forall q \in Q, \forall f \in F$	(13)
$u_i^v - u_j^v + M(1 - x_{ij}^v) \geq e_{ij}^v$	$\forall (i,j) \in A^v, \forall v \in V$	(14)
$p_j^v - p_i^v + M(1 - x_{ij}^v) \geq p_i$	$\forall (i,j) \in A^v \setminus P_q^v \cup D_q^v \cup H_q^v, \forall v \in V$	(15)
$m_j^v - m_i^v + M(1 - x_{ij}^v) \geq m_i$	$\forall (i,j) \in A^v \setminus P_q^v \cup D_q^v \cup H_q^v, \forall v \in V$	(16)
$p_j^f - p_i^f + M(1 - z_{ij}^f) \geq p_i$	$\forall (i,j) \in A^f \setminus P_q^f \cup D_q^f \cup H_q^f, \forall f \in F$	(17)
$m_j^f - m_i^f + M(1 - z_{ij}^f) \geq m_i$	$\forall (i,j) \in A^f \setminus P_q^f \cup D_q^f \cup H_q^f, \forall f \in F$	(18)
$p_i^v = 0$	$\forall i \in K^v, \forall v \in V$	(19)
$x_{ij}^v \in \mathbf{B}$	$\forall (i,j) \in A^v, \forall v \in V$	(20)
$z_{ij}^f \in \mathbf{B}$	$\forall (i,j) \in A^f, \forall f \in F$	(21)

$$0 \leq u_i^v \leq u^{max}, u_i^v \in \mathbf{R}_+ \quad \forall i \in N^v, \forall v \in V \quad (22)$$

$$0 \leq p_i^v + m_i^v \leq cap^{max}, p_i^v, m_i^v \in \mathbf{R}_+ \quad \forall i \in N^v, \forall v \in V \quad (23)$$

$$0 \leq p_i^f + m_i^f \leq cap^{max}, p_i^f, m_i^f \in \mathbf{R}_+ \quad \forall i \in N^f, \forall f \in F \quad (24)$$

目標式(1)為最大化計程車業者利潤，利潤為業者服務營收減去營運成本，第一及第二大項為服務營收，分別為電動計程車載客(貨)節線的服務營收及汽油計程車載客(貨)節線的服務營收，第三及第四大項為營運成本，包括電動計程車供給成本(購買或折舊成本)及旅行成本(車輛於運輸及配送之旅行時間的耗電成本)與汽油計程車供給成本(購買或折舊成本)及旅行成本(車輛於運輸及配送之旅行時間的耗油成本)。

限制式(2)及式(4)分別確保每層電動車時空網路及汽油車時空網路的供給節線及未用車輛節線的流量必為 1，若是該層時空網路所對應之車輛必須提供服務，則必會從場站出發；若是該層時空網路所對應之車輛無須提供服務，則未用車節線流量即為 1，表示該車輛停留於場站。式(3)及式(5)則為確保每層電動車時空網路及汽油車時空網路的匯集節線及未用車輛節線的流量必為 1，若是該層時空網路之車輛有進行運輸或配送服務，完成當日之服務後必會回到場站；若是該層時空網路之車輛無進行運輸或配送服務，則未用車節線流量必為 1，表示該車輛停留於場站。式(6)為針對每個乘客運輸或貨物配送之需求，必有一輛電動計程車或是汽油計程車對其進行載客(貨)服務(pickup service)。式(7)為針對每個乘客運輸或貨物配送之需求，必有一台電動計程車或是汽油計程車對其進行卸客(貨)服務。式(8)及式(9)為針對每層電動車及汽油車時空網路的每個乘客運輸或貨物配送之需求，有對其進行載客(貨)服務後才可對其進行卸客(貨)服務(delivery service)，亦即針對同一個(客或貨)需求，必須由同一台電動計程車或汽油計程車進行卸與載之服務。式(10)及式(11)確保每層電動車時空網路及汽油車時空網路中，每個實體節點與虛擬節點皆須符合流量守恆之限制。式(12)及式(13)分別消除電動車時空網路及汽油車時空網路中的子迴圈並定義乘客運輸或貨物配送需求係由何層時空網路車輛所服務。消除子迴圈之概念為假設某需求若被某層時空網路之車輛服務，則該層時空網路之需求時空節點必有兩進兩出的節線流量為 1，包含載客(貨)節線、卸客(貨)節線及兩旅行節線，流量加總為 4；然若該需求不被該層時空網路之車輛服務，則該層時空網路之車輛必不會行經該需求時空節點，流量加總為 0。式(14)針對電動車時空網路電動車行經各節線後之剩餘電量作定義，旅行節線之耗電量為正值，充電節線之耗電量為負值，其餘節線則無耗電量。式(15)及式(16)分別針對電動車時空網路中，電動車行經需求運輸節點後之車內乘客數及貨物數作定義，起點運輸節點之乘客或貨物數量為正值，迄點運輸節點之乘客或貨物數量為負值。式(17)及式(18)的作用與式(15)及式(16)相同，針對汽油車時空網路中汽油車行經需求運輸節點後之車內乘客數及貨物數作定義。式

(19)為確保電動車內有乘客時，不能前往充電站進行充電。式(20)及式(21)分別定義決策變數 x_{ij}^v 、 z_{ij}^f 。式(22)為限制電動車之電量上下限。式(23)及式(24)為確保電動車及汽油車內乘客總數量及貨物總數量必小於車輛容量。

3.4 小範例測試

為了驗證本研究發展之使用電動車之一般共享問題最佳化模式之正確性，本研究參考業者實際營運資料，自行設計一小範例，並以最佳化求解軟體 Gurobi 進行求解與測試。

3.4.1 參數設定

此範例之收入與成本及充/耗電參數設定如表 3.1 所示。

表 3.1 小範例測試參數設定

項目		設定
收入/成本	業者收入(乘客)	前 1.25 公里 75 元，之後 20 元/公里
	業者收入(貨物)	前 1 公里 90 元，之後 10 元/公里
	供給成本	20 元/每輛車輛
	滯留成本	1 元/15 分鐘
	耗電成本	0.42 元/車公里
	耗油成本	3.83 元/車公里
	政策成本	15 元/每段路段
充/耗電量	電池容量	33 千瓦小時
	耗電量	0.16 千瓦小時/公里
	充電量	9.9 千瓦小時/15 分鐘

此範例有 6 個需求，分別為 3 個乘客及 3 個貨物，各需求之起訖時間窗及其乘客當量數如表 3.2 所示。

表 3.2 小範例起迄時間窗及乘客當量數

需求編號	乘客當量數	起點時窗	迄點時窗
貨物 1	0.5	09:15 – 09:30	10:45 – 11:00
貨物 2	0.5	09:45 – 10:00	11:00 – 11:15
貨物 3	0.5	09:15 – 09:30	11:30 – 11:45
乘客 1	1	09:45 – 10:00	10:00 – 10:15
乘客 2	1	09:15 – 09:30	09:30 – 09:45
乘客 3	1	09:45 – 10:00	10:00 – 10:15

服務時軸訂為 3 個小時，並以 15 分鐘為一個時間單位，共有 12 個時點；共有 12 個站點，並設有 6 座充電站；規劃 2 輛電動車及 2 輛汽油車提供服務，其問題規模如表 3.3 所示。

表 3.3 小範例問題規模

項目		數量
數量	乘客數	3
	貨物數	3
汽油計程車時空網路	時空網路層數	2
	每層網路節點數	229
	每層網路節線數	3,399
電動計程車時空網路	時空網路層數	2
	每層網路節點數	301
	每層網路節線數	5,809
數學模式	限制式數量	51,400
	連續變數數量	3,620
	整數變數數量	19,864

3.4.2 測試結果

此範例之測試結果如表 3.4 所示，求解時間小於 5 秒，系統需要 2 輛電動車及 1 輛汽油車進行服務，其車輛排程及充電規劃如圖 3.6 至圖 3.8 所示。

表 3.4 小範例測試結果

項目	結果
營運收入 (元)	993
營運成本 (元)	496
營運利潤 (元)	497
求解時間 (秒)	4.08
Gap (%)	0.00
使用電動計程車數	2
使用汽油計程車數	1
未使用電動計程車數	0
未使用汽油計程車數	1

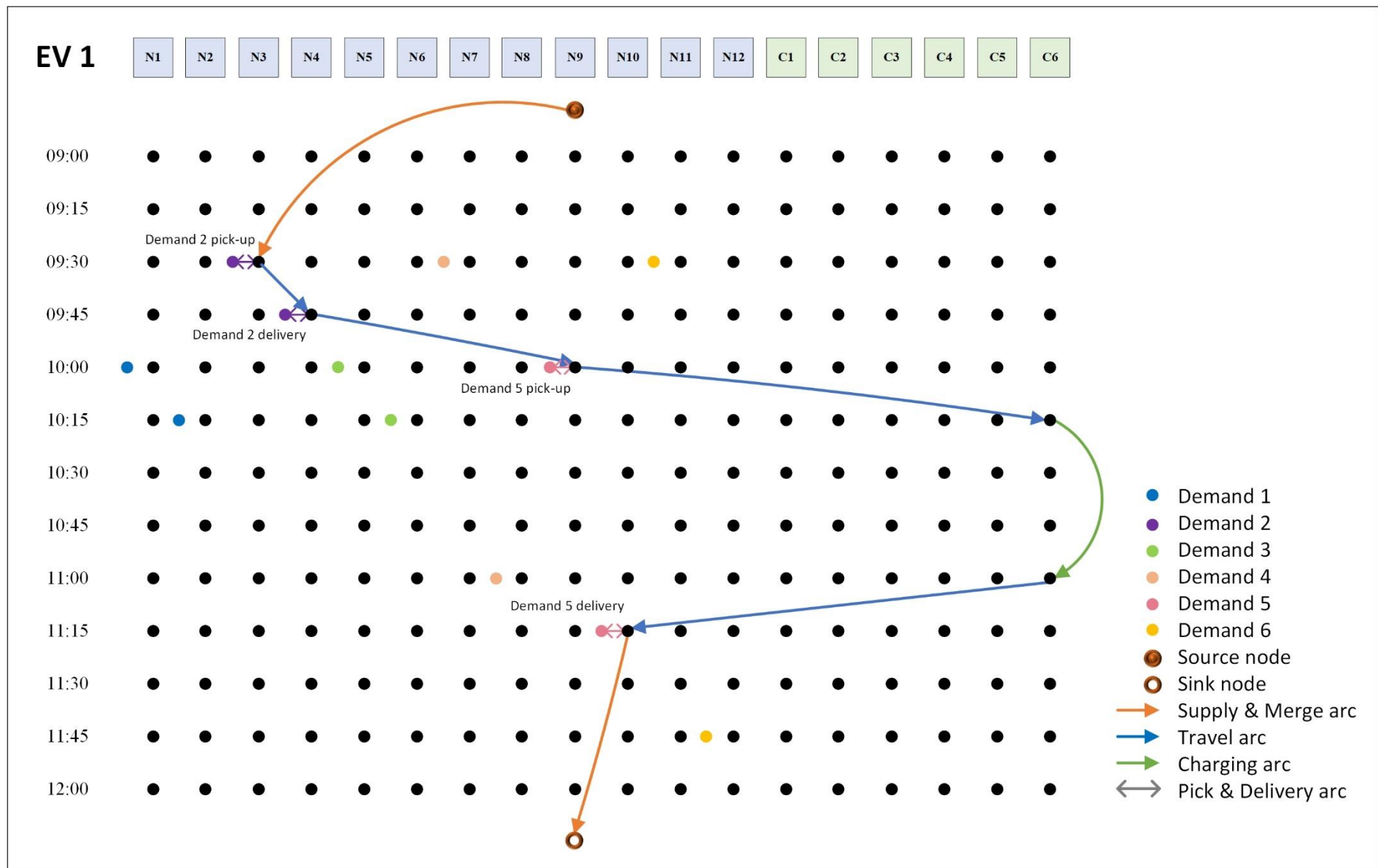


圖 3.6 小範例之車輛排程結果(EV1)

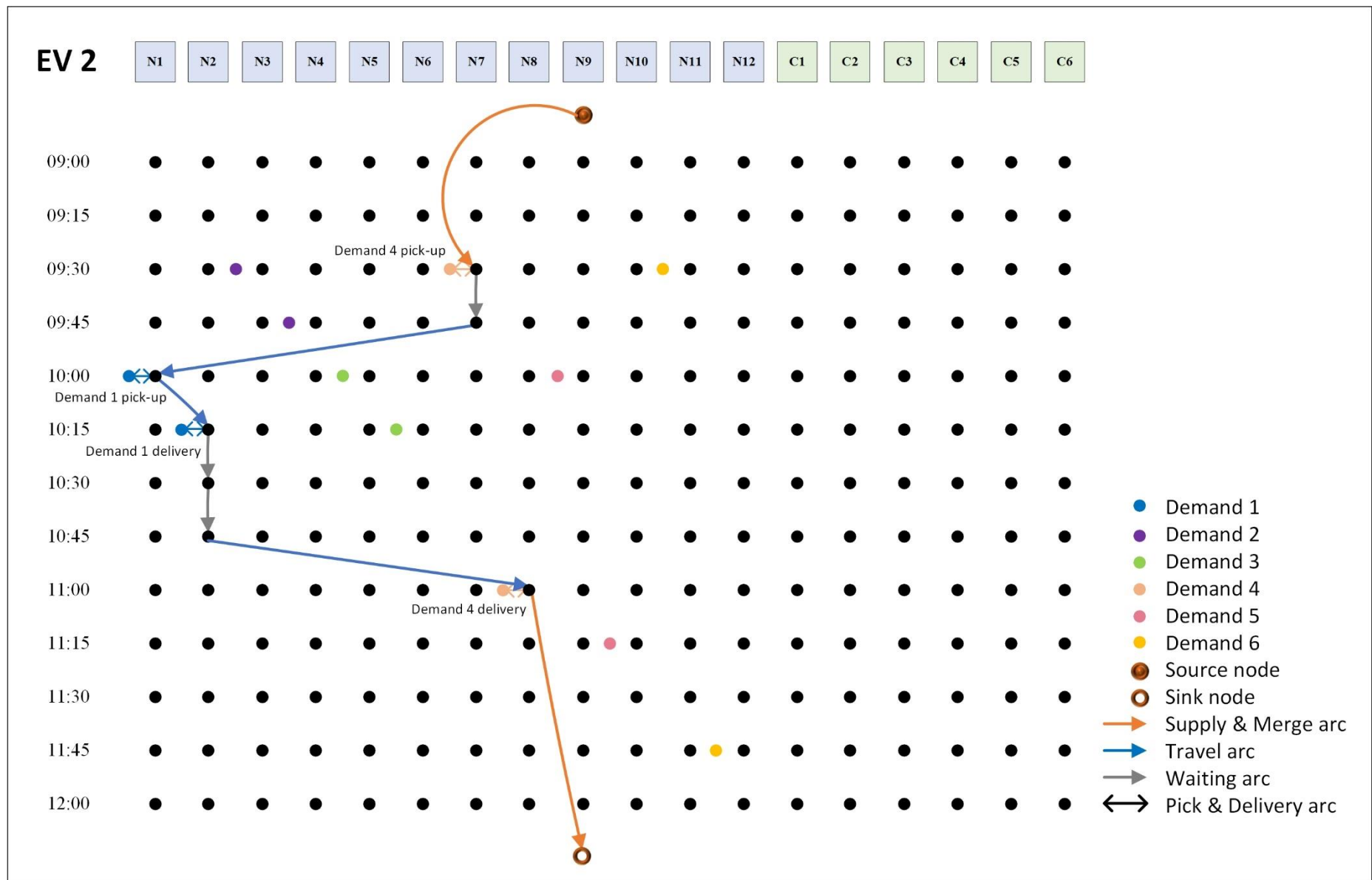


圖 3.7 小範例之車輛排程結果(EV2)

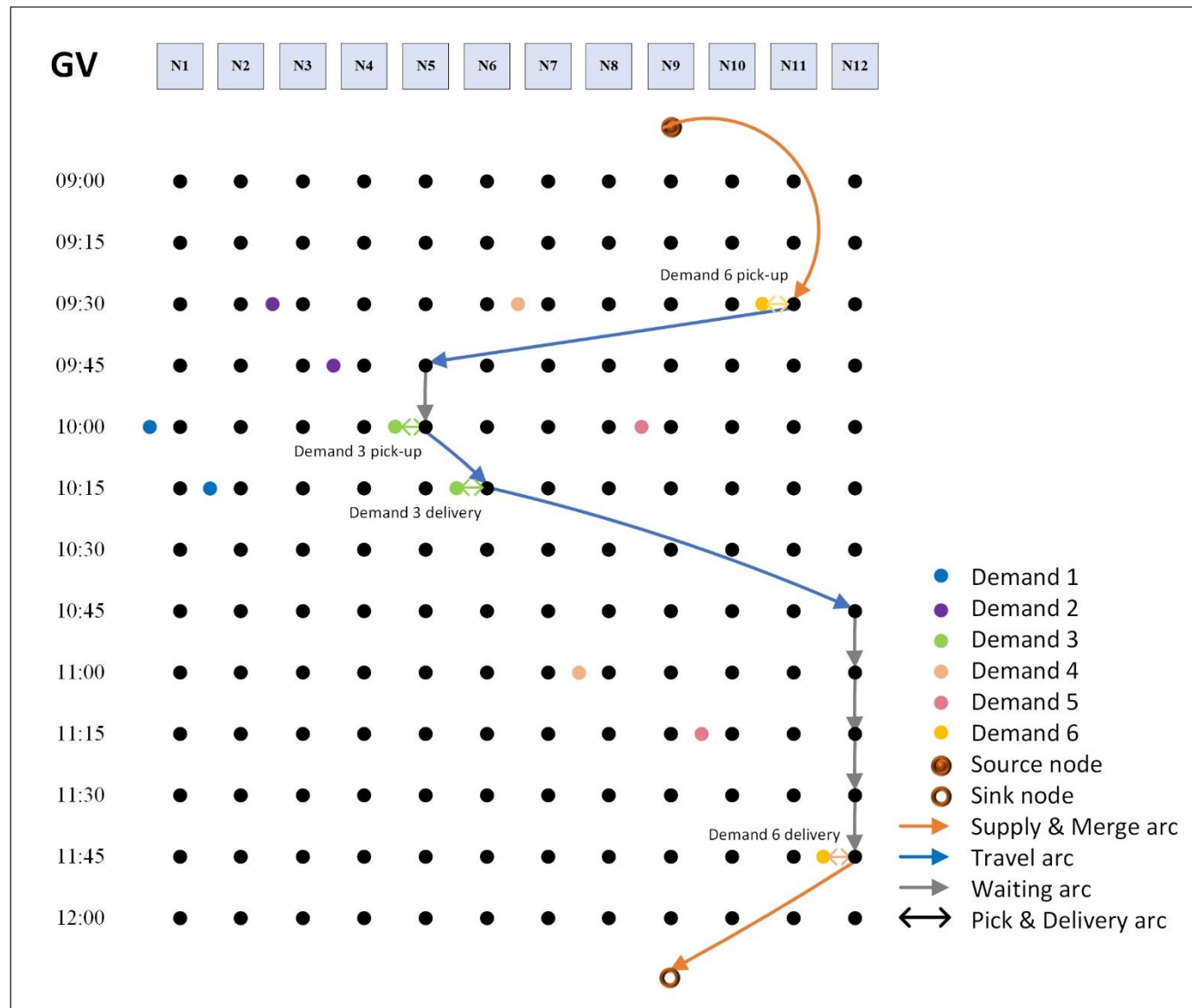


圖 3.8 小範例之車輛排程結果(GV)

第四章 啟發式演算法

4.1 模擬退火法架構與介紹

由於本研究之問題為 NP-Hard 問題，於求解上有一定之難度，故須提出啟發式演算法提升求解效率，進行文獻回顧後發現模擬退火法(Simulated Annealing, SA)應用於一般共乘問題上有良好之求解表現，故於本節根據 G-SARP-EVs 問題特性提出模擬退火法，以利後續較大規模範例之求解。

「退火」代表一種物理過程，將固體加熱至足夠的高溫時，會從固態開始轉成液態，這時所有的分子會在液態物質中隨機地自由排列，隨著溫度緩慢下降，這些分子會逐漸到較低能量程度的結晶狀態並進行重新排列，然模擬退火法係利用「溫度」此一元素操控求解流程，圖 4.1 為本研究之模擬退火法架構圖，以下針對操作步驟進行介紹。

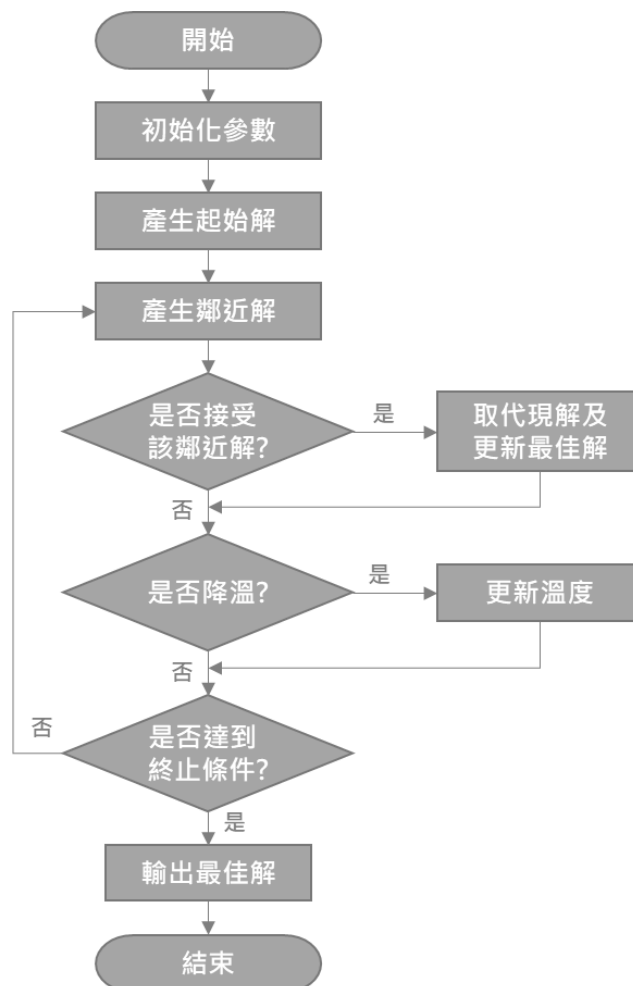


圖 4.1 模擬退火法之架構圖

1. 初始化參數

模擬退火法之參數包括起始溫度(T_0)、終止溫度(T_f)、冷卻率(α)及單一溫度迭代次數(I_{iter})。

2. 產生起始解

設定起始迭代次數 $I = 0$ ，產生起始解並將其設為現解 $X = X_0$ 及設為最佳解 $X_{best} = X_0$ ，最佳能量值 $E_{best} = E_0$ 。

3. 產生鄰近解

由現解產生一鄰近解 X' ，能量值為 E' 。

4. 是否接受該鄰近解

計算鄰近解之能量值(E')與現解之能量值(E)之差， $\Delta E = E' - E$ ，並根據下列波茲曼函數決定是否接受鄰近解取代現解：

$$P = \begin{cases} 1, & \Delta E \leq 0 \\ e^{-\Delta E/T}, & \Delta E > 0 \end{cases}$$

若接受則取代現解 $X = X'$ ；若能量值優於最佳能量值 $E < E_{best}$ ，更新最佳解 $X_{best} = X$ 。

5. 是否降溫

$I = I + 1$ ，判斷是否達到設定之單一溫度迭代次數(I_{iter})，若已達到則進行降溫並更新溫度 $T = \alpha \times T$ 。

6. 是否達到終止溫度

判斷溫度是否達到設定之終止溫度(T_f)，若否則回到步驟 3 重複執行。

7. 輸出最佳解

達到終止條件後，輸出當前最佳解 X_{best} 。

4.2 可行解之呈現

為於同一基準之下比較最佳化數學模式以及模擬退火法之求解效果與效率，本演算法之可行解呈現(solution representation)參照前述 3.2 小節時空網路模型進行設計，並以 3.4 小節之小範例為例進行設計之說明。

可行解的呈現如圖 4.2 所示，本例規劃 2 輛電動車及 2 輛汽油車提供服務，服務時軸為 3 個小時，並以 15 分鐘為一個時間單位，加上時點 0 共有 13 個時點，每一橫列表示該輛車輛的路徑，每一縱行表示該時點下每輛車輛的位置。

由於本範例有 6 個需求，於演算法內編碼從阿拉伯數字 3 開始，奇數為需求起點，偶數為需求迄點，需求 1 之起迄點編碼為(3, 4)、需求 2 之起迄點編碼為(5, 6)……以此類推，需求 6 之起迄點編碼為(13, 14)，並設有 6 座充電站，編碼從 15 至 20。以電動車輛 1(EV1)之排程結果舉例說明，EV1 於時點 2 時抵達需求 2 之起點(5)進行載客，接著前往需求 2 之迄點(6)進行卸客，再往需求 5 之起點(11)進行載貨，完成載貨後前往充電站(20)進行充電 45 分鐘，充飽電後前往需求 5 之迄點(12)進行卸貨，最後回到場站。

	0	1	2	3	4	5	6	7	8	9	10	11	12
EV1	0	0	5	6	11	20	20	20	20	12	0	0	0
EV2	0	0	9	0	3	4	0	0	10	0	0	0	0
GV1	0	0	13	0	7	8	0	0	0	0	0	14	0
GV2	0	0	0	0	0	0	0	0	0	0	0	0	0

圖 4.2 可行解的呈現

4.3 起始解之產生

本演算法之起始解產生乃是應用貪婪演算法(Greedy Algorithm)的概念，一次僅將一個需求插入於一輛車輛的路徑中，過程中會優先選擇電動車輛，若無合適之電動車輛，再選擇汽油車輛，最後挑選可行且插入成本最低之車輛進行需求插入，依序完成所有的需求插入後，產出起始解，其流程圖如圖 4.4 所示，以下將以 6 個需求之小範例，針對每個步驟的細節作介紹。

1. 初始化解的呈現

首先進行範例規模設定，包含電動車及汽油車之數量以及規劃時軸、時點長度，此範例備有 2 輛電動車、2 輛汽油車，規劃時軸為 3 小時，以 15 分鐘為一單位，加上時點 0 共有 13 個時點，設定完成後，即可進行解的呈現之初始化，亦即將此二維串列(13×4)的值皆設為 0，如圖 4.3 所示。

	0	1	2	3	4	5	6	7	8	9	10	11	12
EV1	0	0	0	0	0	0	0	0	0	0	0	0	0
EV2	0	0	0	0	0	0	0	0	0	0	0	0	0
GV1	0	0	0	0	0	0	0	0	0	0	0	0	0
GV2	0	0	0	0	0	0	0	0	0	0	0	0	0

圖 4.3 初始化解的呈現

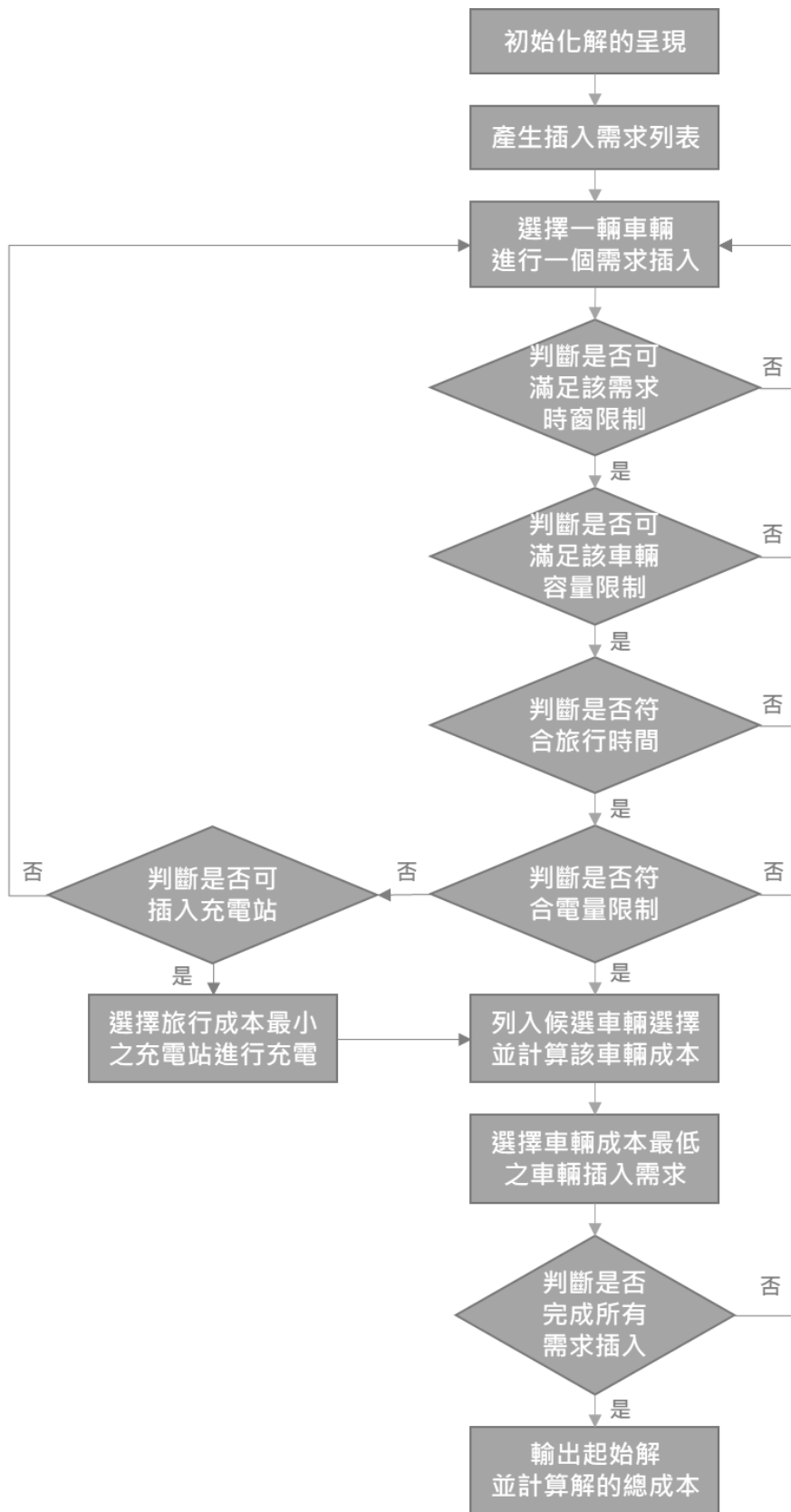


圖 4.4 產生起始解之流程圖

2. 產生插入需求列表

此步驟之目的為產出需求插入之先後順序的列表，提供後續進行需求插入之依循。本範例之各需求起迄時窗如表 4.1 所示，排序方法係將需求以起點時窗作先後排序，時窗愈早之需求，插入順序愈前面，如需求 2 之順序必在需求 1 之前，然若有起點時窗相同之情形，如需求 2、需求 4 及需求 6 之起點時窗皆為時點 2，再以迄點時窗進行先後排序，最後產出插入需求列表，如下表 4.2 所示。

表 4.1 需求起迄時窗表

需求	編碼	起點時窗(時點)	迄點時窗(時點)
1	(3, 4)	4	5
2	(5, 6)	2	3
3	(7, 8)	4	5
4	(9, 10)	2	8
5	(11, 12)	4	9
6	(13, 14)	2	11

表 4.2 插入需求列表

插入順序	1	2	3	4	5	6
需求	2	4	6	1	3	5
編碼	(5, 6)	(9, 10)	(13, 14)	(3, 4)	(7, 8)	(11, 12)

3. 選擇一輛車輛進行一個需求插入

由步驟 2 產出之插入需求列表，依序選擇需求進行插入，選定需求後，依序選擇欲插入之車輛進行演算法步驟 4、5、6、7 之判斷。假設現解已完成順序 1 之需求 2 的插入(如圖 4.5 所示)，現欲插入之需求為順序 2 之需求 4，起迄點編碼為 9、10，該需求將一次選擇一輛車輛進行插入需求，選擇車輛順序依序為 EV1、EV2、GV1 至 GV 2，直到完成所有可能插入車輛之判斷。

	0	1	2	3	4	5	6	7	8	9	10	11	12
EV1	0	0	5	6	0	0	0	0	0	0	0	0	0
EV2	0	0	0	0	0	0	0	0	0	0	0	0	0
GV1	0	0	0	0	0	0	0	0	0	0	0	0	0
GV2	0	0	0	0	0	0	0	0	0	0	0	0	0

圖 4.5 完成插入順序 1 需求 2 之現解的呈現

4. 判斷是否可滿足該需求時窗限制

承步驟 3，需求 4 之編碼為 9、10，時窗為時點 2 及時點 8，判斷是否可滿足該需求時窗限制係判斷該車輛之該時點的值是否為 0，假設現在欲插入之車輛為 EV1，該車輛之時點 2 的值不為 0，故 EV1 無法滿足該需求時窗限制，則需回到步驟 3，依序選擇下一輛車輛；下一車輛為 EV2，此車輛之時點 2 及時點 8 皆為 0，即表示此車輛滿足需求 2 之時窗限制，可列為候選插入車輛(如圖 4.6 所示)，並接續進行步驟 5。

	0	1	2	3	4	5	6	7	8	9	10	11	12
EV1	0	0	5	6	0	0	0	0	0	0	0	0	0
EV2	0	0	9	0	0	0	0	0	10	0	0	0	0
GV1	0	0	0	0	0	0	0	0	0	0	0	0	0
GV2	0	0	0	0	0	0	0	0	0	0	0	0	0

圖 4.6 需求 4 候選插入車輛之解的呈現

5. 判斷是否可滿足該車輛容量限制

每輛車輛有其車輛容量上限，每個需求亦有預先設定之載/卸乘客之數量，正值表示載客，負值表示卸客，為求單位一致，貨物的表示方式為乘客當量，此範例之需求 1、2、3 為乘客，需求 4、5、6 為貨物(包裹)，其相應之載/卸乘客之數量如表 4.3 各需求之載/卸乘客之數量列表所示。承步驟 4，判斷 EV2 可滿足需求 4 之時窗限制後，接著判斷插入需求 4 是否滿足 EV2 之容量限制，如圖 4.7 所示，若容量上限為 2 個乘客，則此範例判斷不超出該車輛容量上限，接續進行步驟 6；若判斷超過車輛容量上限，則需回到步驟 3，依序選擇下一輛車輛。

表 4.3 各需求之載/卸乘客之數量列表

需求	編碼	載/卸乘客之數量
1	(3, 4)	(1.0, -1.0)
2	(5, 6)	(1.0, -1.0)
3	(7, 8)	(1.0, -1.0)
4	(9, 10)	(0.5, -0.5)
5	(11, 12)	(0.5, -0.5)
6	(13, 14)	(0.5, -0.5)

	0	1	2	3	4	5	6	7	8	9	10	11	12
EV2	0	0	9	0	0	0	0	0	10	0	0	0	0
載/卸乘客數量	0	0	1	0	0	0	0	0	-1	0	0	0	0
車內容量	0	0	1	1	1	1	1	1	0	0	0	0	0

圖 4.7 插入需求 2 於 EV2 之車輛容量限制判斷

6. 判斷是否符合旅行時間

此步驟之目的為確保該需求之插入為該車輛在時間上可行之路徑，站點與站點間有其旅行距離，設定車輛平均速度後可計算出旅行時間(點)矩陣，以圖 4.8 為例，第一列為 EV1 車輛排程，第二列為已知的站點與站點間之旅行時點，假設現欲插入需求 5(編碼 11、12)於 EV1 中，需計算插入此需求後是否符合車輛旅行時間，亦即抵達排程中任兩站點之時點差是否大於所需的旅行時間，以(6, 11)為例，抵達編碼 6 的時點為 3，抵達編碼 11 的時點為 4，時點差為 $4 - 3 = 1$ ，已知(6, 11)之旅行時間為 1 個時點，因 $1 \geq 1$ 故該需求插入符合車輛旅行時間，完成整個路徑之判斷後接續進行步驟 7；若判斷出路徑中任兩站點不符合車輛旅行時間，表示該路徑為旅行時間不可行之解，需回到步驟 3，依序選擇下一輛車輛。

	0	1	2	3	4	5	6	7	8	9	10	11	12
EV1	0	0	5	6	11	0	0	0	0	12	0	0	0
旅行時點		1	1	1		2				1			

圖 4.8 插入需求 4 於 EV1 之旅行時間判斷

7. 判斷是否符合電量限制

站點與站點間之旅行距離亦可換算成車輛之耗電量比例，以單位距離耗電量除以電池容量求得，單位為百分比(%)。此步驟之目的為確保插入該需求後，總耗電量不會超出車輛可使用電量(100%)，係確保可產出電動車於電量上可行之排程。若判斷符合車輛電量限制，則完成所有可行解之判斷，接續進行步驟 10；然若判斷總耗電量大於 100%，表示該電動車有充電之需求，進一步計算電量瓶頸點及所需充電時間。以圖 4.9 為例，該排程之總耗電量為 114%，有充電之需求，電量瓶頸點落在累積耗電量大於或等於 100%之時點，為該例中的時點 9，表示行駛到站點 12 以前有充電的需求，且至少需要充 $114\% - 100\% = 14\%$ 的電量，由於現有充電之設計為每 15 分鐘可充 30%之電量，故本例之所需充電時間至少為 15 分鐘，計算完成後接續進行步驟 8。

	0	1	2	3	4	5	6	7	8	9	10	11	12
EV 1	0	0	5	6	11	0	0	0	0	12	0	0	0
耗電量(%)		25	20	25	1	1	1	1	30	10			
累積耗電量(%)			25	45	70	71	72	73	74	104	114		

圖 4.9 需充電之車輛排程範例

8. 判斷是否可插入充電站

由步驟 7 計算出電量瓶頸點及所需充電時間後，接著進一步判斷在該電量瓶頸點前是否有足夠時間可以進行充電，並確保無違反有乘客於車內不能進行充電之限制，如圖 4.10 所示，本範例有足夠時間進行充電，且該時點間車內無乘客，接續進行步驟 9；然若有任一條件不成立，則需回到步驟 3，依序選擇下一輛車輛。

	0	1	2	3	4	5	6	7	8	9	10	11	12
EV1	0	0	5	6	11	0	0	0	0	12	0	0	0
可充電之時點	X	X	X	X	X	V	V	V	V				
車內乘客數	0	0	1	0	0	0	0	0	0				

圖 4.10 插入充電站之判斷

9. 選擇旅行成本最小之充電站進行充電

由步驟 8 得知本範例中，時點 5、6、7、8 可進行充電，由站點 11 前往充電站，充電結束後欲前往站點 12，此步驟為選擇一個旅行時間可行且旅行成本最低之充電站，選擇完畢後將該充電站插入車輛路徑中，結果如圖 4.11 所示，並接續進行步驟 10。

	0	1	2	3	4	5	6	7	8	9	10	11	12
EV1	0	0	5	6	11	20	20	20	20	12	0	0	0

圖 4.11 插入充電站之車輛路徑結果

10. 列入候選車輛選擇並計算該車輛成本

將所有符合步驟 4 至步驟 9 的候選車輛排程結果暫存，並計算每輛候選車輛插入需求前與插入需求後之成本差異。

11. 選擇車輛成本最低之車輛插入需求

選擇車輛插入該需求前後成本差距最小之車輛，亦即插入成本最低之車輛完成該需求之插入。

12. 判斷是否完成所有需求插入

判斷是否已完成所有需求插入，若尚有需求未進行插入，則回到步驟 3，選擇尚未完成插入之需求進行插入；若已完成所有需求之插入，則接續進行步驟 13。

13. 輸出起始解並計算解的總成本

承步驟 12，完成所有需求插入後之可行解即為起始解，故可計算起始解之總成本(能量值)，包含所有車輛之供給成本、旅行成本以及滯留成本，輸出並完成起始解之產生。

4.4 鄰近解之產生

過去文獻中之鄰近解產生包含許多方法，如插入法(Insert)、交換法(Swap)、反序法(Reverse)及突變法(Mutate)等，每種方法又有多種設計方法且效果與優勢不同，本演算法設計的鄰近解產生方法以插入法(Insert)及交換法(Swap)為核心，並針對 G-SARP-EVs 問題特性進行設計。

圖 4.12 為本研究鄰近解產生的主要架構圖，可分為需求插入法、排程交換法以及需求交換法三種，產生鄰近解之步驟如下：

1. 每次產生鄰近解前先產生一個亂數機率，用以決定該次欲採用何種方法產生鄰近解。
2. 選擇三種方法之機率相同，決定採用何種方法產生鄰近解後，則進行該方法之鄰近解產生。
3. 由於並非每種方法每次皆可產出一可行鄰近解，故須判斷是否有產出鄰近解。
4. 若當次該方法無法產出鄰近解，以目前解作為鄰近解；若有產出鄰近解，則以該解作為鄰近解。

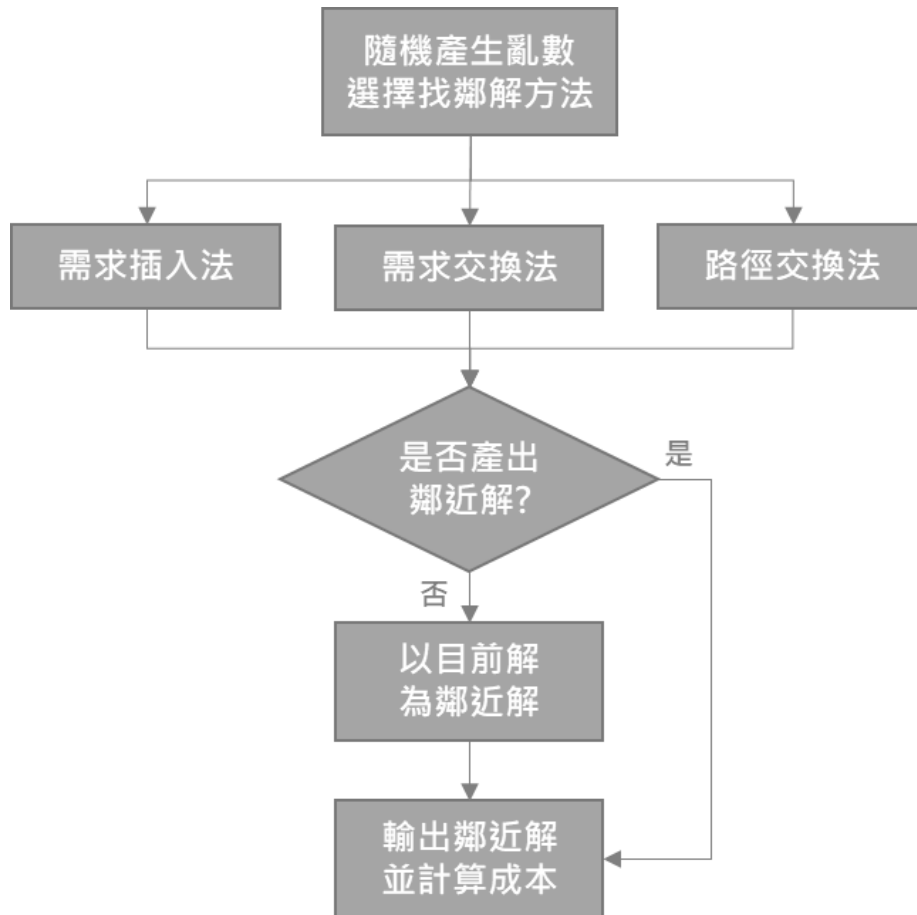


圖 4.12 鄰近解產生之主要架構圖

演算法之鄰域搜尋設計必須兼顧廣度搜尋及深度搜尋，三種產生鄰近解之方法兼顧此核心價值，其中需求插入法之設計概念為「減少車輛使用數」，相較於其他兩種方法更能保證可找到成本更低之車輛排程組合，係深度搜尋；然排程交換法之設計概念為「增加車輛排程組合之可能」，並不保證會求得成本更低之車輛排程結果，係廣度搜尋；最後需求交換法，則各有一半之機率求得成本較低及成本較高之車輛排程結果，可為廣度搜尋或為深度搜尋。

以下三小節將依序針對「需求插入法」、「排程交換法」及「需求交換法」三種方法進行其詳細操作步驟介紹。

4.4.1 需求插入法

需求插入法之核心設計概念為「減少車輛使用數」，隨機選擇一需求，將之從原先服務之車輛排程中移除，並插入另一已進行服務之車輛排程中，而原先被汽油車服務之需求被選擇進行需求插入法的機率為原先被電動車服務之需求的兩倍，期以達到優先減少使用汽油車的效果，需求插入法之架構圖如圖 4.13 所示，以下介紹其操作設計。

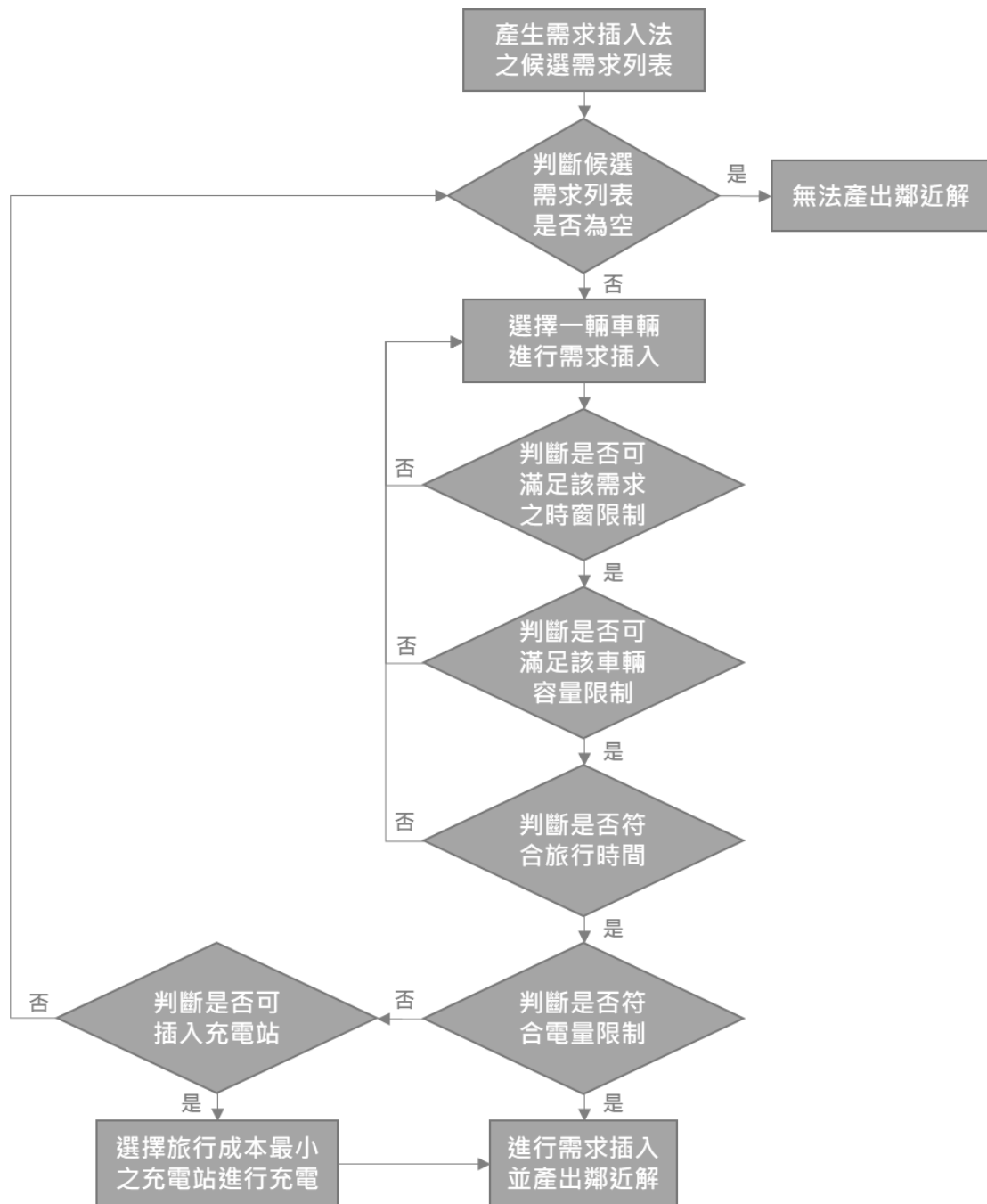


圖 4.13 需求插入法之架構圖

1. 產生需求插入法之候選需求列表

此步驟之目的為產生需求插入法之候選需求列表，基本上所有的需求皆會在此列表中，然為了達到優先減少汽油車輛的目的，會將原先被汽油車服務之需求插入列表兩次，原先被電動車服務之需求僅插入列表一次，使原先被汽油車服務之需求相較於原先被電動車服務之需求有較高的機率被優先選擇，最後產出候選需求列表。以圖 4.14 為例，產出之候選需求列表如表 4.4。

	0	1	2	3	4	5	6	7	8	9	10	11	12
EV1	0	0	5	6	11	20	20	20	20	12	0	0	0
EV2	0	0	9	0	0	0	0	0	10	0	0	0	0
GV1	0	0	13	0	7	8	0	0	0	0	0	14	0
GV2	0	0	0	0	3	4	0	0	0	0	0	0	0

圖 4.14 產生需求插入法之候選需求排程範例

表 4.4 需求插入法之候選需求列表

需求	2	5	4	6	6	3	3	1	1
編碼	(5,6)	(11,12)	(9,10)	(13,14)	(13,14)	(7,8)	(7,8)	(3,4)	(3,4)

2. 判斷候選需求列表是否為空

若候選需求列表為空，則無法以插入需求法產出鄰近解；若不為空，則隨機選擇需求並接續進行步驟 3。

3. 選擇一輛車輛進行需求插入

將步驟 2 指定之需求依序插入已進行服務且不為原先服務該需求之車輛中，並進行演算法步驟 4 至 9 之判斷。以圖 4.14 為例，假設現在欲進行需求 1 編碼(3,4)之需求插入，將依序考慮 EV1、EV2 及 GV1 之可行性判斷。

4. 判斷是否可滿足該需求之時窗限制

承步驟 3，需求 1 之時窗為時點 4 及時點 5，判斷是否可滿足該需求時窗限制係判斷該車輛之該時點的值是否為 0，假設現在欲插入之車輛為 EV1，該車輛之時點 4 及時點 5 的值皆不為 0，故 EV1 無法滿足該需求時窗限制，則需回到步驟 3，依序選擇下一輛車輛；下一車輛為 EV2，此車輛之時點 4 及時點 5 皆為 0，即表示此車輛滿足需求 1 之時窗限制，可列為候選插入車輛(如圖 4.15 所示)，並接續進行步驟 5。

	0	1	2	3	4	5	6	7	8	9	10	11	12
EV2	0	0	9	0	0	0	0	0	10	0	0	0	0
GV2	0	0	0	0	3	4	0	0	0	0	0	0	0

↓

EV2	0	0	9	0	3	4	0	0	10	0	0	0	0
GV2	0	0	0	0	0	0	0	0	0	0	0	0	0

圖 4.15 需求插入法之候選插入車輛路徑

5. 判斷是否可滿足該車輛容量限制

本範例相應之載/卸乘客之數量如表 4.3 所示。承步驟 4，判斷插入需求 1 是否滿足 EV2 之容量限制，如圖 4.16 所示，若容量上限為 2 個乘客，則此範例判斷不超出該車輛容量上限，接續進行步驟 6；若判斷超過車輛容量上限，則需回到步驟 3，依序選擇下一輛車輛。

	0	1	2	3	4	5	6	7	8	9	10	11	12
EV2	0	0	9	0	3	4	0	0	10	0	0	0	0
載/卸乘客數量	0	0	0.5	0	1.0	-1.0	0	0	-0.5	0	0	0	0
車內容量	0	0	0.5	0.5	1.5	0.5	0.5	0.5	0	0	0	0	0

圖 4.16 插入需求 1 於 EV2 之車輛容量限制判斷

6. 判斷是否符合旅行時間

此步驟與產生起始解方法之步驟 6 操作概念及設計相同，可參照 4.3 小節。承步驟 5，需求 1 之插入後 EV2 符合旅行時間，故接續進行步驟 7；然若判斷不符合旅行時間，則需回到步驟 3，依序選擇下一輛車輛。

7. 判斷是否符合電量限制

此步驟與產生起始解方法之步驟 7 操作概念及設計相同，可參照 4.3 小節之範例說明。承步驟 6，本範例判斷符合車輛電量限制，則完成所有可行解之判斷，接續進行步驟 10；然若判斷總耗電量大於 100%，表示該電動車有充電之需求，進一步計算電量瓶頸點及所需充電時間後接續進行步驟 8。

8. 判斷是否可插入充電站

此步驟與產生起始解方法之步驟 8 操作概念及設計相同，可參照 4.3 小節之範例說明。

9. 選擇旅行成本最小之充電站進行充電

此步驟與產生起始解方法之步驟 9 操作概念及設計相同，可參照 4.3 小節之範例說明。

10. 進行需求插入並產出鄰近解

若有任一車輛完成步驟 4 至步驟 9 之判斷，則將該需求插入該輛符合所有限制之車輛，最後產出該解為鄰近解。

4.4.2 排程交換法

排程交換法之設計概念為將任一輛滿足交換條件之電動車排程與任一輛滿足交換條件之汽油車排程進行交換，排程交換法之架構圖如圖 4.17 所示，本方法可分為電動車(左)以及汽油車(右)兩大部分，以下將介紹其操作步驟。

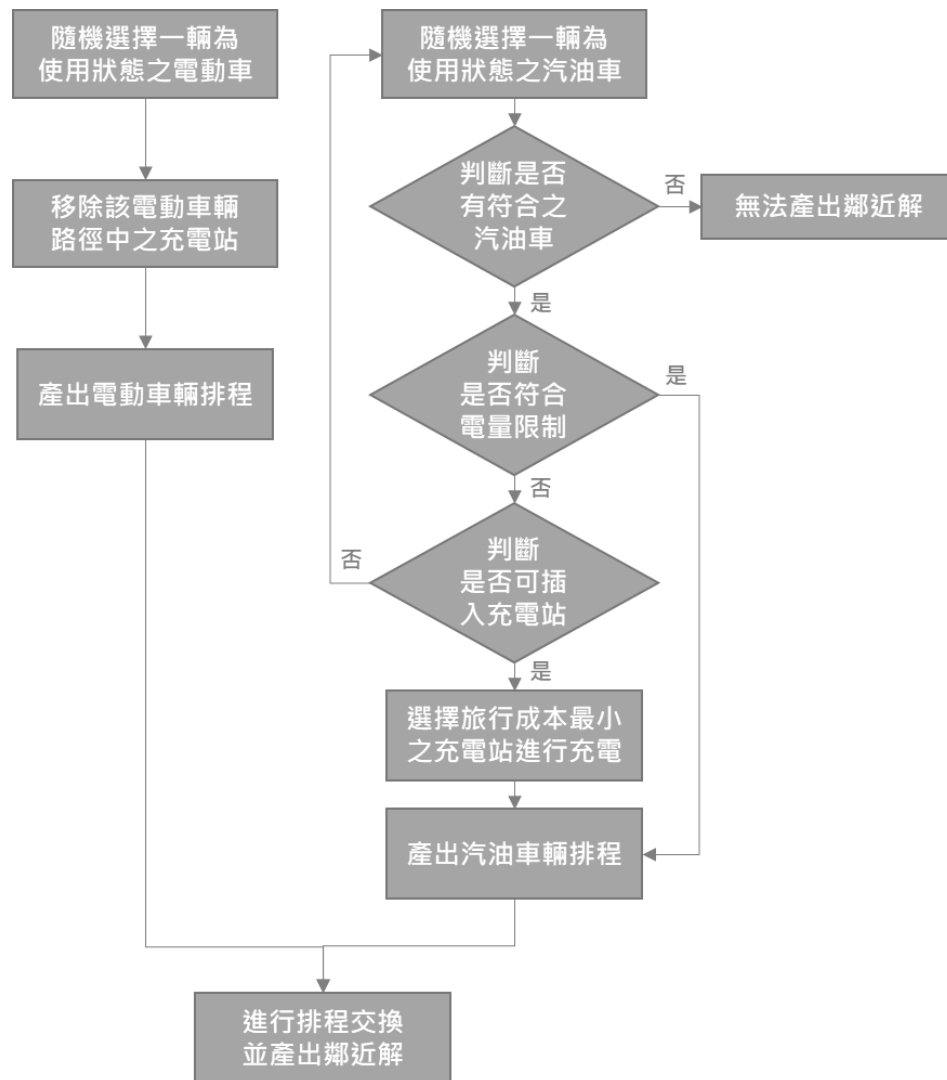


圖 4.17 路徑交換法之架構圖

1-1. 隨機選擇一輛為使用狀態之電動車

此步驟係選擇一輛欲進行排程交換之電動車輛，為不任意增加車輛使用數量，故選擇已有進行服務之車輛進行排程交換。

1-2. 移除該電動車輛排程中之充電站

因排程交換法之操作為將電動車與汽油車之排程進行交換，故當原先由電動車服務之路徑換至汽油車服務時，則不須考量耗電及充電之限制，故於此步驟中先將電動車路徑中之充電站編碼移除。

1-3. 產出電動車輛排程

完成選定及移除充電站後，則完成電動車排程進行交換前之修正，產出該排程等待進行排程交換。

2-1. 隨機選擇一輛為使用狀態之汽油車

此步驟與步驟 1.1 之概念相同，係選擇一輛欲進行排程交換之汽油車輛，為不任意增加車輛使用數量，故選擇已有進行服務之車輛進行路徑交換。

2-2. 判斷是否有符合之汽油車

為秉持不任意增加車輛使用數量之原則，若現解中無已使用之汽油車，則無法進行後續交換，故無法以此方法產出鄰近解；然若有已進行服務之汽油車，則隨機選擇一輛汽油車後進行步驟 2-3。

2-3. 判斷是否符合電量限制

因排程交換法之操作為將汽油車與電動車之排程進行交換，故當原先由汽油車服務之路徑換至電動車服務時，則須加以考量耗電及充電之限制，故於此步驟中判斷該汽油車路徑是否符合電量限制，操作設計與產生起始解方法之步驟 7 相同，可參照 4.3 小節之範例說明，若判斷符合，則跳至步驟 2-6；若判斷不符合電量限制，則接續進行步驟 2-4。

2-4. 判斷是否可插入充電站

若該汽油車路徑判斷耗電量大於 100%，則需插入充電站，判斷是否可插入充電站之操作設計亦與產生起始解方法之步驟 8 相同，可參照 4.3 小節之說明。若判斷可進行插入，則接續進行步驟 2-5；若判斷無法進行充電，則回步驟 2-1 重新選擇汽油車。

2-5. 選擇旅行成本最小之充電站進行充電

若該排程判斷可插入充電站，故選擇旅行成本最低之充電站進行充電，並修正其車輛排程，詳細操作設計可參考產生起始解方法之步驟 9。

2-6. 產出汽油車輛排程

完成選定及進行條件判斷後，完成汽油車排程交換前之修正，最後產出該修正後之排程等待進行排程交換。

3. 進行排程交換並產出鄰近解

將步驟 1-3 及步驟 2-6 產出之兩排程進行交換，並產出排程交換後之解為鄰近解。

4.4.3 需求交換法

需求交換法之設計概念為將有條件地將兩個由不同車輛服務之需求進行交換，需求交換法之架構圖如圖 4.18 所示，以下將依序介紹其操作步驟。

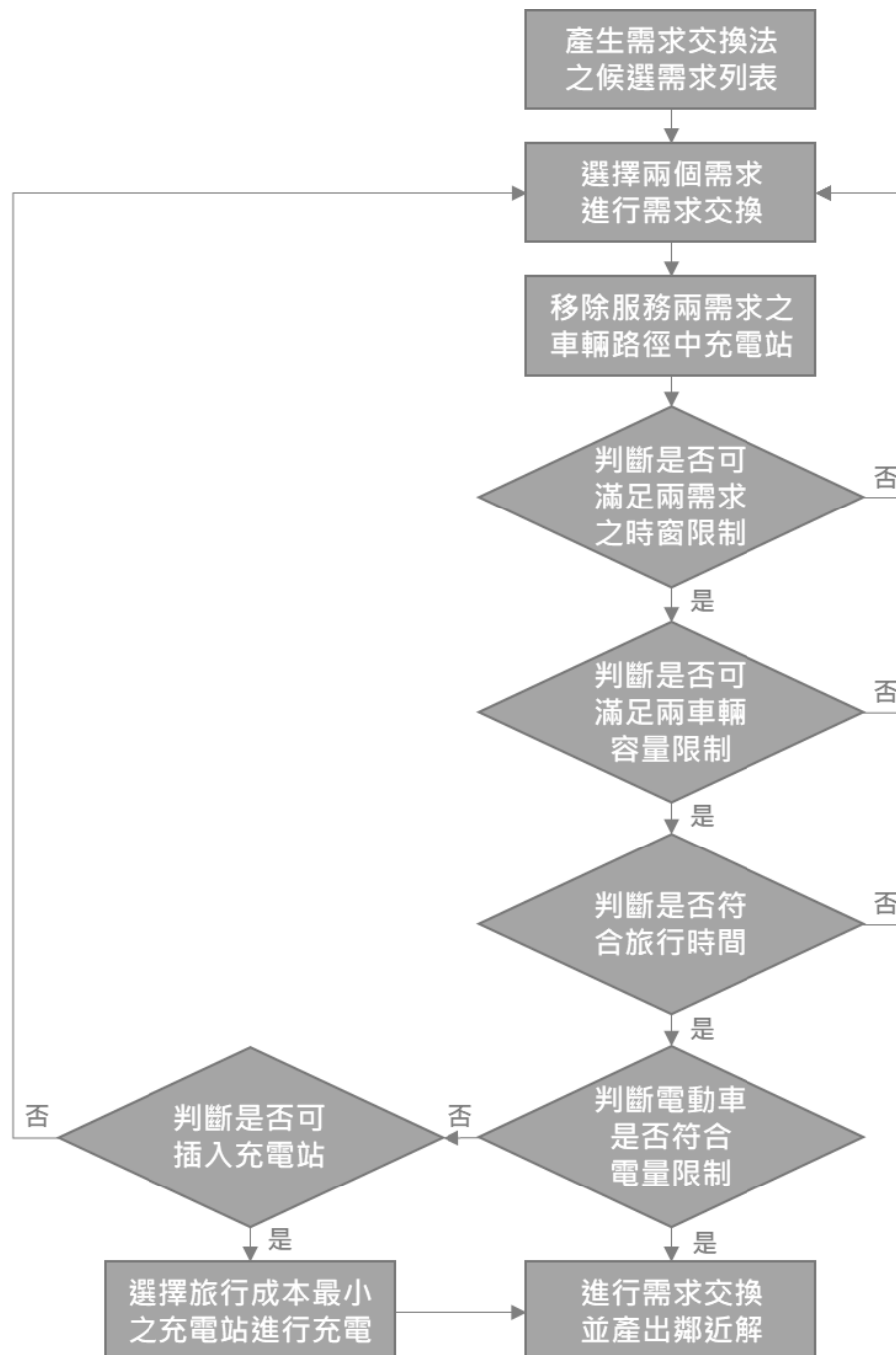


圖 4.18 需求交換法之架構圖

1. 產生需求交換法之候選需求列表

此步驟係為產生需求交換法之候選需求列表，起點時窗或迄點時窗相同之任何需求即可被列入需求交換法之候選需求列表中，僅以該需求之起點編碼進行紀錄，以圖 4.19 為例，編碼(5, 6)、(9, 10)及(13, 14)之需求起點時窗皆為時點 2，編碼(11, 12)、(3, 4)及(7, 8)之需求起點時窗皆為 4，故將其兩兩需求對列入需求交換法之候選需求列表中，產出之候選需求列表如表 4.5。

	0	1	2	3	4	5	6	7	8	9	10	11	12
EV1	0	0	5	6	11	20	20	20	20	12	0	0	0
EV2	0	0	9	0	3	4	0	0	10	0	0	0	0
GV	0	0	13	0	7	8	0	0	0	0	0	14	0

圖 4.19 產生需求交換法之候選需求列表之範例

表 4.5 需求交換法之候選需求列表

#	可進行交換之需求對 (以起點編碼為代表)
1	(5, 9)
2	(5, 13)
3	(9, 13)
4	(11, 3)
5	(11, 7)
6	(3, 7)

2. 選擇兩個需求進行需求交換

此步驟係從需求交換之候選需求列表中隨機選擇一需求對進行需求交換，選定後接續進行後續步驟之操作及判斷。

3. 移除服務兩需求之車輛排程中充電站

承步驟 2，選定欲交換之需求對後，需先檢視服務該需求對之兩輛車輛是否有至充電站進行充電，若車輛判斷有進行充電，則先將充電站之編碼從該排程中移除，以此增加需求交換法之可行性；若無進行充電，則略過此步驟之操作。以圖 4.19 為例，假定現選定交換之需求對為(5, 9)，由於 EV1 有進行至編碼 20 之充電站進行充電，故需先將所有充電站進行移除，操作完成後如圖 4.20。

	0	1	2	3	4	5	6	7	8	9	10	11	12
EV1	0	0	5	6	11	0	0	0	0	12	0	0	0
EV2	0	0	9	0	3	4	0	0	10	0	0	0	0

圖 4.20 完成移除充電站編碼之車輛排程範例

4. 判斷是否可滿足該需求之時窗限制

承步驟 3，需求編碼(5, 6)之時窗為時點 2 及時點 3，需求編碼(9, 10)之時窗為時點 2 及時點 8，分別判斷欲交換之車輛是否可滿足需求時窗限制，亦時點的值是否為 0。如圖 4.21 所示，兩輛車輛皆滿足需求時窗限制，故可接續進行步驟 5；然若有任一車輛無法滿足需求之時窗限制，則需回到步驟 2，隨機選擇另一需求對。

	0	1	2	3	4	5	6	7	8	9	10	11	12
EV1	0	0	5	6	11	0	0	0	0	12	0	0	0
EV2	0	0	9	0	3	4	0	0	10	0	0	0	0

↓

EV1	0	0	9	0	11	0	0	0	10	12	0	0	0
EV2	0	0	5	6	3	4	0	0	0	0	0	0	0

圖 4.21 需求交換法之候選交換車輛排程

5. 判斷是否可滿足兩車輛容量限制

本範例相應之載/卸乘客之數量如表 4.3 所示。承步驟 4，判斷交換需求後之 EV1 與 EV2 是否滿足容量限制，判斷方式與產生起始解方法之步驟 5 大致相同，可參照 4.3 小節，僅需求交換法需確認兩輛車輛皆不違反容量限制，可將相同操作執行兩次。假定容量上限為 2 個乘客，若判斷兩車輛皆不超出車輛容量上限，接續進行步驟 6；然若判斷任一車輛超過車輛容量上限，則回到步驟 2，隨機選擇另一需求對。

6. 判斷是否符合旅行時間

此步驟與產生起始解方法之步驟 6 操作概念及設計相同，可參照 4.3 小節，惟需針對兩輛車輛進行判斷，若皆符合旅行時間可接續進行步驟 7；然若任一車輛不符合旅行時間，回步驟 2 隨機選擇另一需求對。

7. 判斷電動車是否符合電量限制

僅需針對電動車進行此步驟，操作概念及設計與產生起始解方法之步驟 7 相同，可參照 4.3 小節之範例說明。若判斷符合車輛電量限制，則完成所有

可行解之判斷，接續進行步驟 10；然若判斷任一電動車總耗電量大於 100%，表示電動車有充電之需求，進一步計算電量瓶頸點及所需充電時間後接續進行步驟 8。

8. 判斷是否可插入充電站

此步驟與產生起始解方法之步驟 8 操作概念及設計相同，可參照 4.3 小節之範例說明。

9. 選擇旅行成本最小之充電站進行充電

此步驟與產生起始解方法之步驟 9 操作概念及設計相同，可參照 4.3 小節之範例說明。

10. 進行需求交換並產出鄰近解

若兩輛車輛皆完成步驟 4 至步驟 9 之判斷及修正，則將需求對進行交換，並產出交換後之解為鄰近解。

第五章 範例測試與分析

為了測試本研究發展之使用電動車之一般共乘問題(G-SARP-EVs)最佳化模式及啟發式演算法之應用績效，本研究以業者提供之實際營運資料，進行不同需求情境之設計並加以測試。

5.1 範例設計

本小節分成三部分，輸入資料說明、相關參數設定、環境設定及模式輸出。本研究測試所需資料主要可分為車隊規劃資料、路網規劃資料及需求規劃資料，相關參數包括規劃時軸與各項成本及收入設定，以下將一一說明。

5.1.1 輸入資料說明

服務車隊包含汽油計程車輛及電動計程車輛，其中電動計程車選擇智能電動車 Luxgen S3 EV+作為電動車參數設定參考，根據其官方網站所提供的資訊，整理相關參數如表 5.1 所示。電動計程車由於能源特性，能源消耗較汽油計程車快速，本研究將追蹤電動車電量使用情形，以剩餘電量佔電池總容量之比例作為記錄方式，若是剩餘電量不足以完成後續旅次，電動計程車則須前往充電站進行充電；充電站之充電方式為直流快充，每 15 分鐘可充 9.9 度電，亦即 30%之電量，若充電 30 分鐘，則可增加 60%之電量，以此類推，最久可充電 45 分鐘，亦即增加 90%之電量，車輛可依所需電量選擇充電時間，若充電後電動計程車之剩餘電量超過 100%，則以 100% 計算。然每輛車輛亦有其容量上限，本研究將車輛之容量皆以乘客當量數表示，上限設為 2 個乘客數。

表 5.1 電動車規格

項目	數值
電池容量	33 千瓦小時
續航力	203 公里
耗電量	0.16 千瓦小時/公里
充電量(直流快充)	9.9 千瓦小時/15 分鐘

路網規劃以業者提供之實際需求資料為基礎，將各需求之起迄地點依相鄰性歸納為 18 個地點，並在規劃範圍內選定 1 個場站作為車輛之所在基地，以及 6 座充電站供電動計程車充電，如圖 5.1 所示，橘點為需求之起迄地點、綠點為充電站、藍點為計程車業者場站。各站點間之旅行距離根據 Google Maps 網站 (<https://www.google.com.tw/maps/>) 得出最短汽車行駛距離，產出一 25×25 之距離矩陣，並進行些微修正，使該距離矩陣滿足三角不等式，接著以此為根據，計

算出各站點間之旅行時間、耗電量及旅行成本。各站點間之旅行時間之計算方式係其旅行距離除以計程車平均速率，本研究將時速設為 40 公里；各站點間之耗電量則是以其旅行距離乘上單位耗電量；電動車之站點間旅行成本係為旅行距離承上單位耗電成本，然汽油車之站點間旅行成本則為旅行距離乘上耗油成本再加上政策成本，成本設定將於 5.1.2 小節說明。

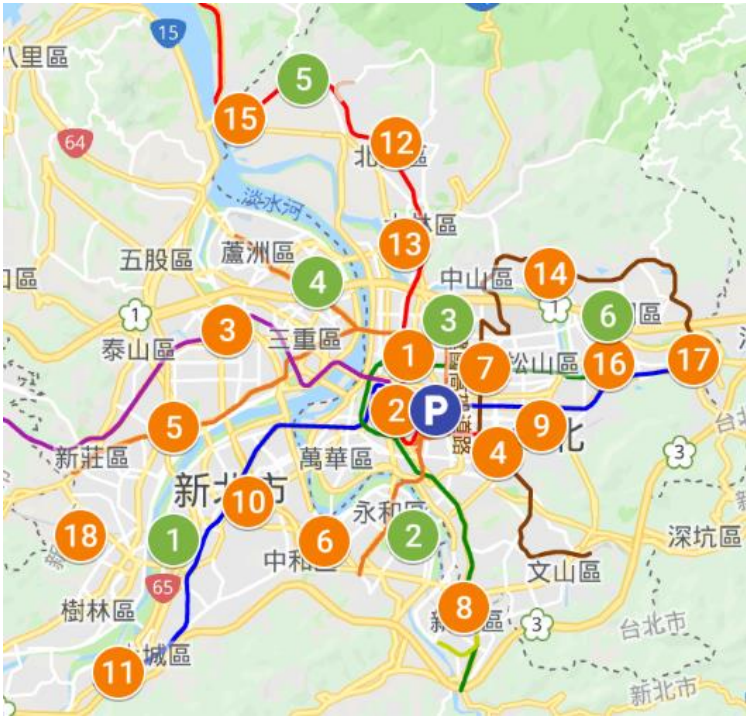


圖 5.1 路網規劃資料

需求規劃資料包含總乘客數、總貨物數、乘客起迄點及其時窗、貨物起迄點及其時窗以及各需求之乘客當量數，其中乘客起迄點、貨物起迄點及起點時窗為隨機產生，而乘客之迄點時窗為起點時窗加上起迄點間旅行時間，以不超出規劃時軸為限；貨物之迄點時窗則是隨機產生，其產生範圍由起點時窗加上起迄點旅行時間至規劃時軸間，表 5.2 為 6 個需求之需求規劃資料範例，並可依此原始需求資料將其轉換成時空網路以便進行最佳化模式求解。

表 5.2 需求規劃資料範例

	編號	起點	時窗	迄點	時窗	乘客當量數
貨物	1	4	09:15 – 09:30	2	10:45 – 11:00	0.5
	2	17	09:45 – 10:00	1	11:00 – 11:15	0.5
	3	5	09:15 – 09:30	8	11:30 – 11:45	0.5
乘客	1	9	09:45 – 10:00	7	10:00 – 10:15	1
	2	16	09:15 – 09:30	14	09:30 – 09:45	1
	3	18	09:45 – 10:00	11	10:00 – 10:15	1

5.1.2 相關參數設定

本研究之服務時軸訂為 3 個小時，並以 15 分鐘為一個時間單位，共有 12 個時點。

計算業者收入方面，參考目前台北市業者計價方式，分為乘客收入及貨物收入，乘客收入計價方式為前 1.25 公里收費 75 元，之後每公里收費 20 元；貨物收入計價方式則為前 1 公里收費 90 元，之後每公里收費 10 元。

成本參數方面，本研究考慮之成本包含供給成本、滯留成本、耗電成本、耗油成本及政策成本，供給成本包含車輛折舊與維修成本，每日每輛車輛的供給成本設為 20 元；滯留成本發生於充電節線及停等節線上，為避免車輛無故產生滯留或停等現象，設定每 15 分鐘 1 元；耗電成本計算方式為每車公里耗電量乘上單位耗電量電費，電費參考台灣電力公司公告之電價表，耗電成本設為每車公里 0.42 元；耗油成本參考新北市政府交通局公告，設為每車公里 3.83 元；政策成本加於汽油車之旅行節線上，藉由增加汽油計程車之旅行成本，使其服務成本高於電動計程車，排程時較易優先指派電動計程車，設定為每一路段 15 元，其參數設定整理於表 5.3。

表 5.3 收入/成本參數設定

項目	設定值
業者收入(乘客)	前 1.25 公里 75 元，之後 20 元/公里
業者收入(貨物)	前 1 公里 90 元，之後 10 元/公里
供給成本	20 元/每日每輛車輛
滯留成本	1 元/15 分鐘
耗電成本	0.42 元/車公里
耗油成本	3.83 元/車公里
政策成本	15 元/每段路段(汽油車)

5.1.3 模式輸出與環境設定

本研究使用之求解電腦為 Microsoft Windows 10 作業系統，求解電腦之硬體配備為 Intel(R) Core(TM) i5-8250U CPU@1.60GHz，12.0GB RAM。G-SARP-EVs 模式以最佳化軟體 Gurobi 進行構建與求解；模擬退火法以 Python 語言進行撰寫及測試，完成測試後模式輸出結果，輸出結果包含：

1. 最佳解目標值
2. 求解時間
3. 使用車輛數
4. 車輛排程結果及電動車輛充電規劃。

5.2 模擬退火法參數設定

本研究設計之模擬退火法中有四項參數，分別為起始溫度(T_0)、終止溫度(T_f)、冷卻率(α)及單一溫度迭代次數(I_{iter})，為使本研究之模擬退火法達到最佳效益，本節參考 Yu et al.(2018)所使用的參數組合並調整，進行模擬退火法參數設定敏感度分析，期以求得兼顧效率及效果的最佳參數組合，表 5.4 為本研究進行測試之參數數值列表。

表 5.4 參數測試數值列表

參數	測試數值
起始溫度(T_0)	5, 6, 7
終止溫度(T_f)	1, 0.1, 0.01
冷卻率(α)	0.96, 0.97, 0.98
單一溫度迭代次數(I_{iter})	100, 200, 300

本研究之參數敏感度分析以 24 個需求之範例進行測試，有 12 個乘客需求及 12 個貨物需求，並規劃 7 輛電動計程車及 6 輛汽油計程車進行服務。每個參數組合皆會進行 5 次求解，並計算其求解時間及目標值的平均值作為該參數組合之代表，再進行參數組合之選擇。

進行參數敏感度分析時，首先隨機選擇一組參數組合作為基礎參數組合，接著依序針對起始溫度(T_0)、終止溫度(T_f)、冷卻率(α)及單一溫度迭代次數(I_{iter})進行參數敏感度測試，每次僅針對欲測試之參數進行更動，且進行下一參數之敏感度分析時，須將上一測試順序之參數的值設定為選定之參數值，以下將詳細說明參數選擇過程，求解結果可參考表 5.5。

隨機選擇之基礎參數組合為參數組合 1， $T_0 = 5, T_f = 1, \alpha = 0.97, I_{iter} = 300$ ，順序一之測試參數為起始溫度(T_0)，可參考圖 5.2，可知起始溫度設定愈高求解時間亦愈長，然起始溫度設定至 6 以上時目標值成長幅度小，故綜合求解時間與目標值的考量下將起始溫度訂為 6。接著進行終止溫度(T_f)之測試，可參考圖 5.3，當終止溫度愈低，求解時間愈長且目標值愈低，故將終止溫度定為 1 係為目標值最高、求解時間最低之選擇。接著進行冷卻率(α)之測試，可參考圖 5.4，在冷卻率設定愈高之情形下，前半段求解時間緩慢上升，目標值卻有較明顯之增長；後半段之求解時間急遽增長，然目標值卻成長緩慢，故可知將冷卻率訂為 0.97 為最佳選擇。最後則為單一溫度迭代次數(I_{iter})之測試，可參考圖 5.5，可知當單一溫度迭代次數設定愈低時，求解時間愈短；以目標值來看，雖迭代次數高時目標值為最高，但當單一溫度迭代次數為 200 時，目標值與 Gurobi 上界解(Upper bound)之差距已在可接受之範圍內，故綜合求解時間與目標值的考量之下，選擇 200 作為單一溫度迭代次數之參數設定值。

在目標值及求解時間權衡下，選定之最佳參數組合為 $T_o = 6, T_f = 1, \alpha = 0.97, I_{iter} = 200$ ，後續將會以此參數組合進行範例測試。

表 5.5 模擬退火法參數敏感度分析

測試 參數	T_o	T_f	α	I_{iter}	目標值 (元)	與 Gurobi 上界解 之 Gap(%)	求解 時間 (分)
T_o	5	1	0.97	300	3,965	2.88	5.60
	6	1	0.97	300	3,990	2.24	6.18
	7	1	0.97	300	3,991	2.20	6.97
T_f	6	1	0.97	300	3,990	2.24	6.18
	6	0.1	0.97	300	3,977	2.56	14.15
	6	0.01	0.97	300	3,965	2.89	22.30
α	6	1	0.96	300	3,967	2.83	4.70
	6	1	0.97	300	3,990	2.24	6.18
	6	1	0.98	300	3,992	2.18	9.37
I_{iter}	6	1	0.97	100	3,879	5.16	2.08
	6	1	0.97	200	3,948	3.33	4.18
	6	1	0.97	300	3,990	2.24	6.18

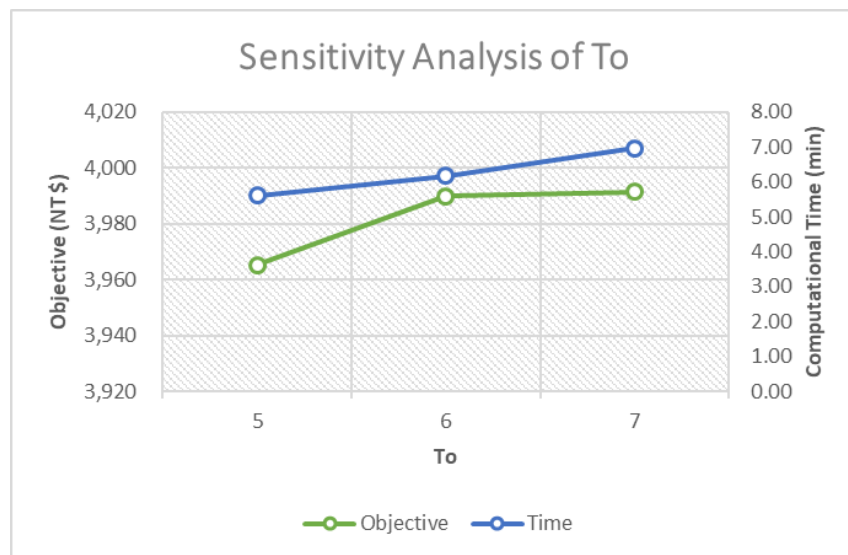


圖 5.2 敏感度分析 - 起始溫度

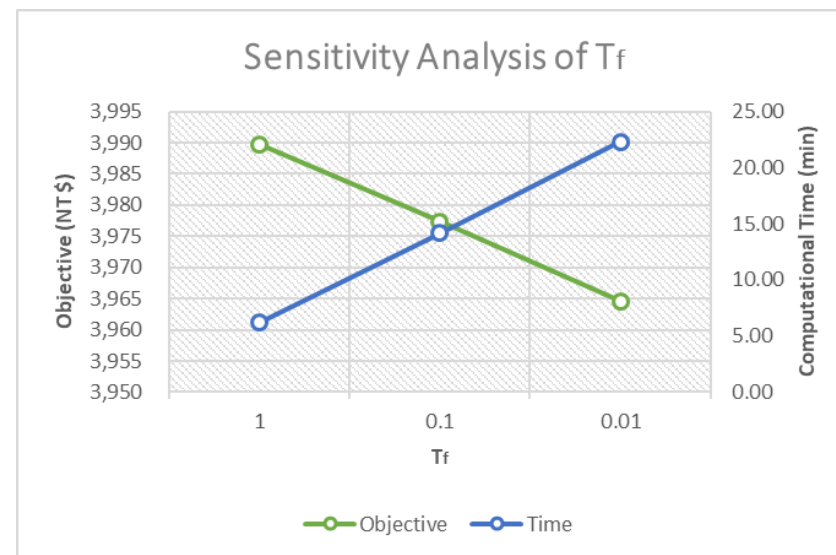


圖 5.3 敏感度分析 - 終止溫度

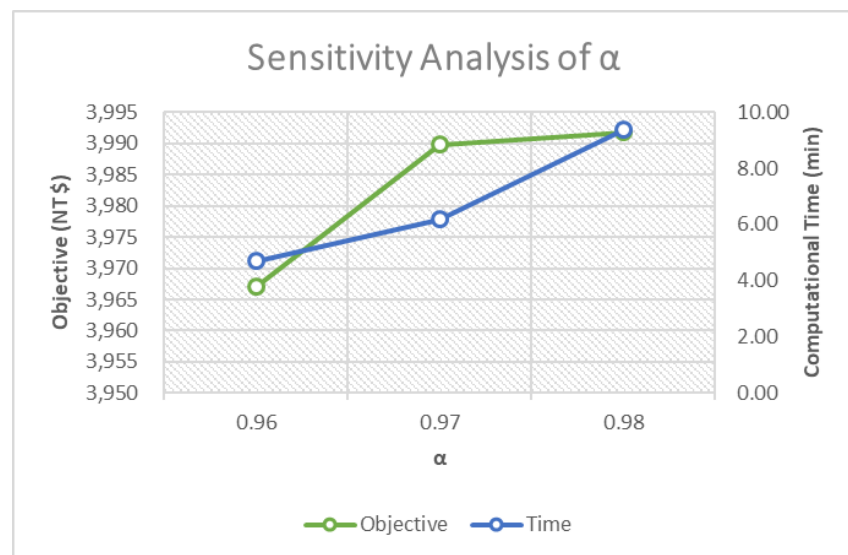


圖 5.4 敏感度分析 - 冷卻率

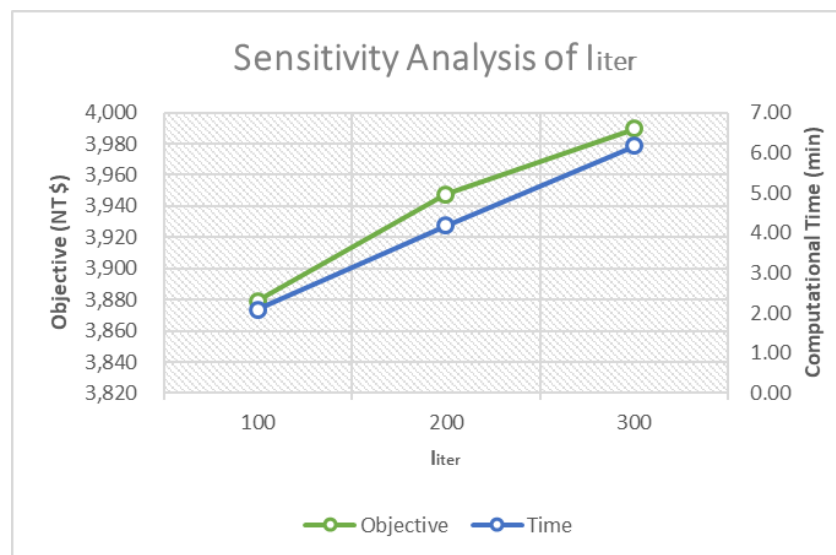


圖 5.5 敏感度分析 - 單一溫度迭代次數

5.3 測試與結果

為了測試本研究發展之 G-SARP-EVs 最佳化模式以及模擬退火法之效益，分別針對小規模範例、中規模範例、及大規模範例進行測試，小規模範例為 Gurobi 可於 24 小時內以求得最佳解(Gap = 0.00%)之範例；中規模範例係 Gurobi 可於 24 小時內求得可行解但非最佳解(Gap > 0.00%)之範例；大規模範例則為 Gurobi 於 24 小時內無法求得解之範例。範例又可依乘客數與需求數的比例分成 a 及 b 兩類，a 類別範例的乘客數與需求數相同，b 類別範例的乘客數為貨物數的兩倍。

5.3.1 小規模範例

1. 需求輸入與規模

小規模範例有 4 個測試例題，有 6 個需求及 12 個需求兩種，分別為 6a、6b、12a、12b，範例規模如表 5.6 所示，同一個需求數之兩類範例規模相同。6 個需求的範例規劃 2 輛電動車及 2 輛汽油車提供服務；12 個需求的範例則規劃 5 輛電動車、2 輛汽油車提供服務。

由於本研究之問題有載客(貨)及卸客(貨)之服務，故設計時空網路時，每個需求之起迄點於空間維度中皆會以兩個站點表示(一個起點及一個迄點)，以 6 個需求為例，即會有 12 個站點，然為減少時空網路中之節線數及節點數，進而減少變數數量，本研究進行範例測試時，若遇不同需求間起迄點相同且時間窗不同時，則可簡化僅用同一個站點表示之，進而達到減少時空網路規模及變數規模之效益，如表 5.6 所示，12a 及 12b 時空網路規模中，每層網路節點數及每層網路節線數僅略多於 6a 及 6b 時空網路，而非 6a 及 6b 時空網路規模的兩倍；然由於 12a 及 12b 之範例中，系統備有較多輛電動車，故數學模式規模約為 6a 及 6b 的兩倍，包含限制式數量、連續變數數量及整數變數數量。

表 5.6 小規模範例規模

項目		6a	6b	12a	12b
數量	乘客數	3	4	6	8
	貨物數	3	2	6	4
汽油車 時空網路	時空網路層數	2		2	
	每層網路節點數	229		241	
	每層網路節線數	3,399		3,435	
電動車 時空網路	時空網路層數	2		5	
	每層網路節點數	301		313	
	每層網路節線數	5,809		5,845	
數學 模式	限制式數量	51,400		106,264	
	連續變數數量	3,620		6,878	
	整數變數數量	19,864		38,629	

2. Gurobi 求解結果

小規模範例之求解結果如表 5.7，每個範例皆求得最佳解，亦即 Gap 為 0.00%，其中 6a 及 6b 的求解時間在 5 秒以內，兩者皆需要 2 輛電動計程車及 1 輛汽油計程車提供服務，利潤分別為 497 元及 406 元；然 12a 及 12b 的求解時間約 2 個小時，12a 需 5 輛電動計程車及 1 輛汽油計程車提供服務，12b 則需 5 輛電動計程車及 2 輛汽油計程車提供服務，利潤分別為 1,753 元及 1,832 元，以營運成本來看，因 12b 比 12a 多使用一輛汽油計程車，故其營運成本明顯高於 12a，不過 12b 之營運收入較高，故利潤仍為 12b 高於 12a。

表 5.7 小規模範例測試結果 - Gurobi

項目	6a	6b	12a	12b
營運收入 (元)	993	893	2,476	2,699
營運成本 (元)	496	487	723	867
營運利潤 (元)	497	406	1,753	1,832
求解時間 (秒)	4.08	2.40	7,020	6,320
與上界解之 Gap (%)	0.00	0.00	0.00	0.00
使用電動計程車數	2	2	5	5
使用汽油計程車數	1	1	1	2
未使用電動計程車數	0	0	0	0
未使用汽油計程車數	1	1	1	0

由小規模範例測試結果可知，本研究之數學模式可有效地求解 G-SARP-EV 問題，並求得各車輛之排程規劃及電動計程車之充電規劃，然由於本研究之問題複雜度高，當範例需求數量愈大時，求解效率逐漸下降，故仍須仰賴啟發式演算法，期以同時兼顧效率及效果。

3. SA 求解結果與比較

為測試本研究所設計之模擬退火法的效果及效率，並與 Gurobi 測試結果進行比較，故本節以 SA 測試小規模範例。

首先，以範例 6a 為例，模擬退火法之求解結果如圖 5.6 所示，其每一橫列表示該相應車輛之排程結果與充電規劃，每一縱行表示該時點下每輛車輛的位置，服務時軸為 3 個小時，並以 15 分鐘為一個時間單位，故共有 12 個時點。6a 範例有 6 個需求，共有 12 個站點，於演算法內編碼從阿拉伯數字 3 開始，奇數為需求起點，偶數為需求迄點，需求 1 之起迄點編碼為(3, 4)、需求 2 之起迄點編碼為(5, 6)……以此類推，需求 6 之起迄點編碼為(13, 14)，並設有 6 座充電站，編碼從 15 至 20，規劃 2 輛電動車及 2 輛汽油車提供服務。從圖 5.6 可知，6a 範例需要 2 輛電動計程車及 1 輛汽油計程車進行服務，而以電動車輛 1(EV1)之排程結果舉例說明，EV1 於時點 2 時抵達需求 2 之起點(5)進行載客，接著前往需

求 2 之迄點(6)進行卸客，再往需求 5 之起點(11)進行載貨，完成載貨後前往充電站(20)進行充電 45 分鐘，充飽電後前往需求 5 之迄點(12)進行卸貨，最後回到場站。將 6a 之所有車輛排程結果與 Gurobi 產出之排程結果相比，可知兩者求得之車輛排程及充電規劃結果相同，皆為最佳解。

	0	1	2	3	4	5	6	7	8	9	10	11	12
EV1	0	0	5	6	11	20	20	20	20	12	0	0	0
EV2	0	0	9	0	3	4	0	0	10	0	0	0	0
GV1	0	0	13	0	7	8	0	0	0	0	0	14	0
GV2	0	0	0	0	0	0	0	0	0	0	0	0	0

圖 5.6 範例 6a 之求解結果(模擬退火法)

由於模擬退火法有其隨機性，針對同一範例，每次求解所需花費的時間及目標值皆不相同，故每個範例皆會進行 5 次求解，表 5.8 列出小規模範例中每個範例的最佳、最差及平均目標值、求解時間及與上界解之差距，由表中結果可知，以模擬退火法進行小範例規模之求解時，以求解時間來看，每個範例每次求解所花費的時間差異不大；以目標值來看，除範例 12a 於 5 次測試中求得一次 Gap 為 0.86%的解外，其餘範例每次皆求得最佳解。

表 5.8 模擬退火法測試比較 - 小規模範例

		6a	6b	12a	12b
目標值 (元)	最佳值	497	406	1,753	1,832
	最差值	497	406	1,738	1,832
	平均值	497	406	1,750	1,832
求解時間 (秒)	最佳值	8.48	8.00	33.51	41.29
	最差值	8.82	8.51	33.93	42.82
	平均值	8.63	8.16	33.57	42.05
與上界解 之 Gap (%)	最佳值	0.00	0.00	0.00	0.00
	最差值	0.00	0.00	0.86	0.00
	平均值	0.00	0.00	0.17	0.00

表 5.9 列出利用模擬退火法進行小規模範例測試後，目標值最大之詳細測試結果，並以此與 Gurobi 求得之最佳上界進行比較，可知每個範例與最佳上界之差距皆為 0.00%，且求解時間皆小於 1 分鐘，求解效果及效率皆佳。

表 5.9 小規模範例測試結果 - SA

項目	6a	6b	12a	12b
營運收入 (元)	993	893	2,476	2,699
營運成本 (元)	496	487	723	867
營運利潤 (元)	497	406	1,753	1,832
求解時間 (秒)	8.48	8.00	32.92	41.29
與上界解之 Gap (%)	0.00	0.00	0.00	0.00
使用電動計程車數	2	2	5	5
使用汽油計程車數	1	1	1	2
未使用電動計程車數	0	0	0	0
未使用汽油計程車數	1	1	1	0

將模擬退火法與 Gurobi 兩者的求解時間與目標值統整於表 5.10 中並進行比較，其中最後一欄之最佳上界(Best bound)係為 Gurobi 進行求解時所得到之最佳上界，用來進行與兩方法求得之目標值進行差距的計算。

以目標值來看，兩方法皆可求得最佳解，即與最佳上界之差距皆為 0.00%；以求解時間來看，雖然 SA 在求解 6a 及 6b 時效率略慢於 Gurobi，但於求解 12a 及 12b 時求解時間比起 Gurobi 大幅降低，從約 2 小時降到 1 分鐘以內，求解效率相比之下 SA 表現極佳。

表 5.10 最佳化求解軟體與模擬退火法之比較 - 小規模範例

Instance	Gurobi			SA			Best bound
	Objective	Time(s)	Gap(%)	Objective	Time(s)	Gap(%)	
6a	497	4.08	0.00	497	8.48	0.00	497
6b	406	2.40	0.00	406	8.00	0.00	406
12a	1,753	7,021	0.00	1,753	32.92	0.00	1,753
12b	1,832	6,320	0.00	1,832	41.29	0.00	1,832

5.3.2 中規模範例

1. 需求輸入與規模

中規模範例有 4 個測試例題，有 24 個需求及 48 個需求兩種，分別為 24a、24b、48a、48b，範例規模如表 5.11 所示。24 個需求範例規劃 7 輛電動計程車及 6 輛汽油計程車提供服務；48 個需求的範例則規劃 12 輛電動計程車及 12 輛汽油計程車提供服務。

表 5.11 中規模範例規模

項目		24a	24b	48a	48b
數量	乘客數	12	16	24	32
	貨物數	12	8	24	16
汽油車 時空網路	時空網路層數	6		12	
	每層網路節點數	265		314	
	每層網路節線數	3,507		3,651	
電動車 時空網路	時空網路層數	7		12	
	每層網路節點數	337		386	
	每層網路節線數	5,917		6,061	
數學 模式	限制式數量	174,780		319,727	
	連續變數數量	11,946		23,160	
	整數變數數量	67,167		125,808	

2. Gurobi 求解結果

中規模範例之求解結果如表 5.12，將求解時間上限設為 86,400 秒，亦即 24 小時，求解結果仍與最佳解有些微差距。24a 的利潤為 3,798 元，離上界解仍有 7.40%的差距，排程結果需 7 輛電動計程車及 3 輛汽油計程車；24b 的目標值與上界解之差距為 4.11%，利潤為 4,185 元，需 7 輛電動計程車及 4 輛汽油計程車；48a 的利潤為 8,285 元，與上界解差距 11.19%，需 12 輛電動計程車與 4 輛汽油計程車；48b 需 12 輛電動計程車與 3 輛汽油計程車，利潤與上界解有 8.76%的差距，利潤為 8,672 元。

表 5.12 中規模範例測試結果 - Gurobi

項目	24a	24b	48a	48b
營運收入 (元)	5,354	5,897	11,128	11,382
營運成本 (元)	1,556	1,712	2,843	2,710
營運利潤 (元)	3,798	4,185	8,285	8,672
求解時間 (秒)	86,400	86,400	86,400	86,400
與上界解之 Gap (%)	7.40	4.11	11.19	8.76
使用電動計程車數	7	7	12	12
使用汽油計程車數	3	4	4	3
未使用電動計程車數	0	0	0	0
未使用汽油計程車數	3	2	8	9

3. SA 求解結果與比較

同樣將中規模範例的四個範例重複進行求解 5 次，並將其最佳、最差及平均目標值、求解時間及與上界解之差距整理於表 5.13 中，由結果可知，以模擬退火法進行中範例規模之求解時，同一個範例的最長及最短求解時間差距皆在 1 分鐘以內，目標值亦無大幅差距，可知本研究設計之模擬退火法每次皆可穩定地求得解。

表 5.13 模擬退火法測試比較 - 中規模範例

		24a	24b	48a	48b
目標值 (元)	最佳值	3,966	4,321	8,324	8,747
	最差值	3,923	4,321	8,268	8,695
	平均值	3,948	4,321	8,287	8,718
求解時間 (秒)	最佳值	248.90	255.40	752.00	740.40
	最差值	252.50	264.80	807.90	774.60
	平均值	251.18	261.58	782.18	756.74
與上界解 之 Gap (%)	最佳值	2.85	0.83	10.67	7.83
	最差值	3.98	0.83	11.42	8.48
	平均值	3.33	0.83	11.16	8.20

表 5.14 列出中規模範例中，利用模擬退火法進行測試後之目標值最佳測試結果，包含目標值、求解時間及使用車輛數。

表 5.14 中規模範例測試結果 - SA

項目	24a	24b	48a	48b
營運收入 (元)	5,354	5,897	11,128	11,382
營運成本 (元)	1,388	1,576	2,804	2,635
營運利潤 (元)	3,966	4,321	8,324	8,747
求解時間 (秒)	252	255	782	740
與上界解之 Gap (%)	2.85	0.83	10.67	7.83
使用電動計程車數	7	7	12	12
使用汽油計程車數	3	4	6	4
未使用電動計程車數	0	0	0	0
未使用汽油計程車數	3	2	6	8

將兩者求解時間與目標值進行比較，如表 5.15 所示，以目標值來看，SA 皆求得比 Gurobi 更佳的目標值，亦即 Gap 更小；以求解時間來看，SA 能於 5 分鐘左右完成求解 24 個需求之範例，48 個需求的範例也僅須約 13 分鐘即可完成，與須花費 24 小時求解之 Gurobi 相比效率大大提升，可知在中規模範例測試中，

SA 皆能求得比 Gurobi 更好的解，且求解時間大幅減少。

表 5.15 最佳化求解軟體與模擬退火法之比較 - 中規模範例

Instance	Gurobi			SA			Best bound
	Objective	Time(s)	Gap(%)	Objective	Time(s)	Gap(%)	
24a	3,798	86,400	7.40	3,966	252	2.85	4,079
24b	4,185	86,400	4.11	4,321	255	0.83	4,357
48a	8,285	86,400	11.19	8,324	782	10.67	9,212
48b	8,672	86,400	8.76	8,747	740	7.83	9,432

5.3.3 大規模範例

1. 需求輸入與規模

大規模範例有 6 個測試例題，有 72 個需求、90 個需求及 144 個需求三種，分別為 72a、72b、90a、90b、144a 及 144b，範例規模如表 5.16 所示。72 個需求範例規劃 18 輛電動計程車及 12 輛汽油計程車提供服務；90 個需求的範例規劃 20 輛電動計程車及 12 輛汽油計程車提供服務；144 個需求範例規劃 28 輛電動車及 22 輛汽油車提供服務，然由於 144a 及 144b 兩範例已無法由 Gurobi 進行求解，故無法求得數學模式規模之資訊。

表 5.16 大規模範例規模

項目		72a	72b	90a	90b	144a	144b
數量	乘客數	36	48	45	60	72	96
	貨物數	36	24	45	30	72	48
汽油車 時空網路	時空網路層數	12		12		22	
	每層網路節點數	362		398		722	
	每層網路節線數	3,795		3,903		12,461	
電動車 時空網路	時空網路層數	18		20		28	
	每層網路節點數	434		470		794	
	每層網路節線數	6,205		6,313		16,529	
數學 模式	限制式數量	442,680		489,504		N/A	
	連續變數數量	33,852		39,480		N/A	
	整數變數數量	170,250		188,136		N/A	

2. Gurobi 求解結果

將求解時間上限設為 86,400 秒，亦即 24 小時。72a、72b、90a 及 90b 四個範例皆無法於時間上限內求得可行解，後續將以 Gurobi 內建演算法求得之上界

解與 SA 之求解結果進行比較；然 144 個需求之兩範例已超出 Gurobi 可求解之上限，故無上界解可進行 Gap 之計算，後續僅會列出 SA 之求解結果。

3. SA 求解結果與比較

將大規模範例的每個範例同樣進行 5 次模擬退火法求解，將測試結果列於表 5.17 中，由結果可知，以模擬退火法進行大範例規模之求解時，同一範例每次求得的目標值差距並不大，而以求解時間來看，最長及最短求解時間之差距除範例 90a 差距約 13.5 分鐘外，其餘範例差距皆小於或約莫 6 分鐘，本研究設計之模擬退火法每次皆可穩定地求得解。

表 5.17 模擬退火法測試比較 - 大規模範例

		72a	72b	90a	90b	144a	144b
目標值 (元)	最佳值	14,439	15,672	18,056	19,132	27,575	29,788
	最差值	14,406	15,571	18,002	19,016	27,481	29,650
	平均值	14,425	15,597	18,028	19,071	27,544	29,683
求解時間 (分)	最佳值	28.73	28.23	62.58	61.68	215.67	220.00
	最差值	30.72	29.67	76.05	63.33	221.83	226.00
	平均值	29.48	28.86	67.09	62.59	219.17	223.10
與上界解 之 Gap (%)	最佳值	8.66	6.38	9.97	8.90	N/A	N/A
	最差值	8.91	7.07	10.30	9.56	N/A	N/A
	平均值	8.76	6.90	10.14	9.25	N/A	N/A

表 5.18 統整利用模擬退火法進行大規模範例測試之最佳測試結果，72a 的利潤為 14,439 元，離 Gurobi 求得之上界仍有 8.66% 的差距，排程結果需 18 輛電動計程車及 5 輛汽油計程車；72b 的目標值與上界解之差距為 6.38%，利潤為 15,672 元，需 18 輛電動計程車及 4 輛汽油計程車；90a 的利潤為 18,056 元，與上界差距 9.97%，90b 的利潤為 19,132 元，與上界差距 8.90%，兩者皆需使用 20 輛電動計程車與 6 輛汽油計程車；144a 的利潤為 27,575 元，排程結果需 28 輛電動計程車及 19 輛汽油計程車；144b 的利潤為 29,788 元，排程結果需 28 輛電動計程車及 18 輛汽油計程車。

將 Gurobi 及 SA 之大規模範例求解結果整理於表 5.19 中，由大規模範例之求解結果可知，當需求數輛及車隊規模逐漸增大，求解時間愈來愈長，尤其是 144a 及 144b 兩範例，求解時間逼近 4 小時，雖 5.2 小節所述之 SA 參數進行求解可得到品質不錯的解，然此兩範例之求解時間明顯較長，在實務上應用有其困難，故以下探討以迴圈數較少之 SA 參數組合進行 144a 及 144b 範例求解之績效比較。

表 5.18 大規模範例測試結果 - SA

項目	72a	72b	90a	90b	144a	144b
營運收入 (元)	16,871	17,968	21,110	22,254	33,742	35,936
營運成本 (元)	2,432	2,296	3,054	3,122	6,167	6,148
營運利潤 (元)	14,439	15,672	18,056	19,132	27,575	29,788
求解時間 (分)	28.73	28.23	76.05	61.68	215.67	220.00
與上界解之 Gap (%)	8.66	6.38	9.97	8.90	N/A	N/A
使用電動計程車數	18	18	20	20	28	28
使用汽油計程車數	5	4	6	6	19	18
未使用電動計程車數	0	0	0	0	0	0
未使用汽油計程車數	7	8	6	6	3	4

表 5.19 最佳化求解軟體與模擬退火法之比較 - 大規模範例

Instance	Gurobi			SA			Best bound
	Objective	Time(m)	Gap(%)	Objective	Time(m)	Gap(%)	
72a	N/A	1,440	N/A	14,439	28.73	8.66	15,689
72b	N/A	1,440	N/A	15,672	28.23	6.38	18,047
90a	N/A	1,440	N/A	18,056	76.05	9.97	19,856
90b	N/A	1,440	N/A	19,132	61.68	8.90	20,834
144a	N/A	N/A	N/A	27,575	215.67	N/A	N/A
144b	N/A	N/A	N/A	29,788	220.00	N/A	N/A

參數設定的部分，起始溫度、終止溫度及冷卻率皆比照 5.2 小節求得之結果進行設定，分別為 6、1 及 0.97，惟進行單一溫度迭代次數參數之調整，比較 200、100 及 50 三個設定值之求解績效，由於模擬退火法有其隨機性，故將每組參數組合皆進行 5 次測試，並以目標值最高之求解結果進行比較，結果整理於表 5.20。

由表中結果可知，於此兩範例中，減少單一溫度迭代次數後，排程結果須使用較多的汽油車輛進行服務，推斷因單一溫度迭代次數減少，故搜尋鄰近解之次數跟著降低，尚未求得車隊規模更精簡之排程結果，然若以求解時間來看，進行調整後可以大幅度縮減求解時間，從原先將近 4 小時之求解時間減少至 1 小時左右，且目標值差距亦不大，故為使本方法更加妥善地應用至解決實務問題上，建議可以藉由減少單一溫度迭代次數之方法來有效縮減求解時間。

表 5.20 大規模範例之單一溫度迭代次數參數設定比較

	項目	144a			144b		
參數 設定	單一溫度 迭代次數	200	100	50	200	100	50
求解 結果	營運收入(元)	33,742	33,742	33,742	35,936	35,936	35,936
	營運成本(元)	6,167	6,276	6,385	6,148	6,286	6,366
	營運利潤(元)	27,575	27,466	27,357	29,788	29,650	29,570
	求解時間(分)	215.67	112.90	61.87	220.00	117.00	57.67
	使用電動 計程車數	28	28	28	28	28	28
	使用汽油 計程車數	19	20	20	18	19	20
	未使用電動 計程車數	0	0	0	0	0	0
	未使用汽油 計程車數	3	2	2	4	3	2

5.4 參數情境分析

本節將針對政策成本及耗電量兩參數進行情境分析，進行該參數分析時，僅針對該參數進行調整，其於參數不變，以範例 6a 作為測試對象，驗證並分析在不同參數設定下之求解結果。

5.4.1 政策成本

政策成本之設計目的為使系統於指派服務車輛時優先選擇電動計程車，以達到環保之效益，故於汽油計程車路網中的旅行節線上加入政策成本，使得以汽油計程車進行服務之營運成本較高。以下設計不同政策成本之情境，進行測試與分析，結果如表 5.21 所示。

情境一為無政策成本之情境，可發現最佳解使用較多輛汽油計程車，且電動計程車無須進行充電；情境二之政策成本設為 1 元，其車輛排程結果與情境一完全一致，目標值亦相同，可見政策成本設定太低時並無效果；情境三將政策成本設為 15 元，最佳解為使用兩輛電動計程車及一輛汽油計程車，且其中一輛電動計程車需進行充電，充電時間為 45 分鐘，雖因充電行為使得整體成本微幅上升，利潤相對下降，但卻可以有效達到優先選擇電動計程車進行服務之效果；情境四將政策成本設為 150 元，最佳解為使用兩輛電動計程車及兩輛汽油計程車，當使用同一輛汽油計程車之旅行成本大於使用兩輛汽油計程車之供給成本時，系統會求解出使用多輛汽油計程車之排程結果，進而導致整體成本提升、利潤下降的情

形，由此得知當政策成本設定過大時，亦無法達成減少使用汽油計程車之效果。

表 5.21 政策成本情境測試結果

	情境一 (基礎情境)	情境二	情境三	情境四
政策成本(元)	0	1	15	150
求解結果				
營運收入(元)	993	993	993	993
總成本(元)	366	372	416	763
政策成本(元)	0	6	45	300
修正後成本(元)	366	366	371	463
營運利潤(元)	627	627	622	530
使用電動計程車數	1	1	2	2
使用汽油計程車數	2	2	1	2
充電規劃	無	無	1 輛 / 45 分鐘	1 輛 / 45 分鐘

5.4.2 耗電量

本研究之耗電量係參考 Luxgen S3 EV+電動車作設定，該電動車之耗電量為 0.16 千瓦小時/公里，詳細規格可參考表 5.1，然本小節欲探討使用不同型號之電動車，亦即耗電量不同之情形下車輛排程及充電規劃之結果變化，情境設計與結果整理於表 5.22 中。

表 5.22 耗電量情境測試結果

	情境一	情境二 (基礎情境)	情境三
耗電量(千瓦小時/公里)	0.08	0.16	0.32
求解結果			
使用電動計程車數	2	2	2
使用汽油計程車數	1	1	2
充電規劃	無	1 輛 / 45 分鐘	1 輛 / 45 分鐘

測試分成三種情境，情境一電動車耗電量為 0.08 千瓦小時/公里，係單位耗電量最小之情境，排程結果為使用兩輛電動計程車及一輛汽油計程車，且電動計程車皆無須進行充電即可完成所有需求服務；情境二電動車耗電量為 0.16 千瓦小時/公里，結果為使用兩輛電動計程車及一輛汽油計程車，其中一輛電動計程車需要充電 45 分鐘方能完成所有需求指派；情境三電動車耗電量為 0.32 千瓦小時/公里，為三個情境中單位耗電量最高的情境，結果為使用兩輛電動計程車及兩輛汽油計程車，且其中一輛電動計程車須進行 45 分鐘的充電，可知當電動車

之單位距離耗電量愈高時，每輛電動計程車可服務之距離下降，也因電動計程車能服務之需求變少，故最佳解為多使用一輛汽油計程車進行服務，以完成所有需求運送。

由此可知，當電動計程車之電池效能提升時，可服務之需求數愈多，需使用之車輛數亦愈少。

第六章 結論與建議

6.1 結論

本研究針對一般共乘問題進行延伸，提出以充電式電動計程車及汽油計程車之混合車隊進行一般共乘問題之運輸與配送服務，係使用電動車之一般共乘問題(G-SARP-EVs)。

研究以計程車營運業者的角度出發，於特定範圍內進行服務，考量營運時間內混合計程車隊與人貨配送需求之配對以及車輛排程與電動計程車充電之規劃，在固定且已知的車隊數量及車輛容量限制下，僅針對營運時間前完成預約之乘客及貨物需求進行服務，以營運業者利潤最大化為目標，利潤計算方式為營運收入減去營運成本，營運收入包含載運乘客及配送貨物之服務營收，營運成本包括供給成本、耗電成本、耗油成本、滯留成本以及政策成本。

根據問題特性，本研究建構時空網路以描述人流、物流與車流在時空維度中的移動情形，並於建構之時空網路定式 G-SARP-EVs 數學規劃模式，亦提出模擬退火法加速求解效率。本研究以業者提供之實際營運資料，進行調整後產出不同規模之測試範例，並以最佳化軟體 Gurobi 進行 G-SARP-EVs 模式構建與求解，模擬退火法則以 Python 語言進行撰寫及求解，進行結果分析與探討。小規模範例測試中，不論是 Gurobi 或模擬退火法皆能在短時間內求得最佳解；中規模範例測試中，驗證本研究發展之模擬退火法的可求得比 Gurobi 更好的解，且求解效率大幅領先 Gurobi；大規模範例中，雖 Gurobi 已無法求得解，然模擬退火法仍可求得一可行解。以下歸納本研究四點主要貢獻：

1. 將 G-SARP 問題進行延伸，加入充電式(Plug-in)電動計程車及汽油計程車之混合車隊進行服務，提出一新的問題 G-SARP-EVs，期待未來業界能引入此想法，達到改善交通壅塞及環境汙染問題並解決運輸與物流產業面臨之難題，同時兼顧經濟、社會及環境三者之效益。
2. 本研究建構時空網路，並分成需求面時空網路及供給面時空網路，清楚地有效地詳細描述人流、物流及車流於時空維度中的流動情形，並於時空網路中加入虛擬節點及虛擬節線之設計，用以表示乘客與貨物之共乘效果，克服於時空網路上描述共乘問題之困難，並可依不同需求數量及規劃時程考量，將時空網路作相應的修改，彈性應用至不同情境的問題。
3. 於建構之時空網路上進行 G-SARP-EVs 問題之數學模式建構，有效地描述利用數學模式描述問題，並在後續順利利用最佳化求解軟體 Gurobi 進行求解，並產出一有效之車輛排程與電動車充電規劃結果。

4. 建立啟發式演算法－模擬退火法(Simulated Annealing, SA)於建構之時空網路模型上，提出適合 G-SARP-EVs 問題特性的可行解呈現(Solution Representation)，並建構產生起始解及鄰近解之方法，克服修正諸多不可行解之型態所衍生之困難，提出一可用之演算法以加速求解效率。

6.2 未來研究建議

本研究相關後續研究方向建議如下：

1. 本研究僅針對事先已知之確定性需求進行建構與求解，未來可發展動態指派演算法，針對即時需求進行載運服務之車輛排程規劃，可供不同業者型態參考與使用。
2. 啟發式演算法之可行解呈現、建立初始解及搜尋鄰近解方法非常多元，建議未來可設計不同於本研究之方法，並進行測試分析與比較，精進演算法之設計，進而提升求解效果及效率。
3. 本研究僅研擬一種啟發式演算法－模擬退火法進行求解，並將求解結果與 Gurobi 最佳化求解軟體進行比較，建議未來可建構不同啟發式演算法，或結合不同啟發式演算法之優勢，進而提出最適合 G-SARP-EVs 問題特性之方法，於求解時同時兼顧效果與效率。

參考文獻

- Barco, J., Guerra, A., Munoz, L., & Quijano, N. (2017). Optimal Routing and Scheduling of Charge for Electric Vehicles: A Case Study. *Mathematical Problems in Engineering*, Vol.2017, pp.16.
- Braysy, I., & Gendreau, M. (2005a) Vehicle routing problem with time windows, part 1: Route construction and local search algorithms. *Transportation Science*, Vol.39, pp.104-118.
- Cordeau, J., & Laporte, G. (2003). The dial-a-ride problem (DARP): Variants, modeling issues and algorithms. *Journal of Operations Research*, Vol.1, pp.89-101.
- Hane, C., Barnhart C., Johnson, E., Marsten, R., Nemhauser, G., Sigismondi, G. (1995). The fleet assignment problem: Solving a large integer program. *Mathematical Programming*, Vol.70, pp.211-232.
- Jaw, J.J., Odoni, A., Psaraftis, H., Wilson, N. (1986). A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. *Transportation Research B: Methodological*, Vol.20, pp.243-257.
- Kim, K.H., & Moon, K.C. (2003). Berth scheduling by simulated annealing. *Transportation Research Part B: Methodological*, Vol.37, pp.541-560.
- Kliwer N., Mellouli T., Suhl L. (2006). A time-space network based exact optimization model for multi-depot bus scheduling. *Europe Journal Operation Research*, Vol.175(3), pp.1616-1627.
- Kuo, Y. (2010). Using simulated annealing to minimize fuel consumption for the time-dependent vehicle routing problem. *Computers and Industrial Engineering*, Vol.59, pp.157-165.
- Li B., Krushinsky D., Reijers H., Van Woensel T. (2014). The share-a-ride problem: people and parcels sharing taxis. *Europe Journal Operation Research*, Vol.238, pp.31- 40.
- Li, B., Krunshinsky D., Van Woensel T., Reijers H. (2016). An Adaptive Large Neighborhood Search Heuristic for the Share-a-Ride Problem. *Computers & Operations Research*, Vol.66, pp.170-180.

- Lin, C., Choy, K.L., Ho, G.T., Chung, S.H., Lam, H.Y. (2014). Survey of green vehicle routing problem: past and future trends. *Expert Systems with Applications*, Vol.41(4), pp.1118-1138.
- Lin, S.W., Yu, V. F., Lu, C.C. (2011). A simulated annealing heuristic for the truck and trailer routing problem with time windows. *Expert Systems with Applications*, Vol.38(12), pp.15244–15252
- Lu, C.C., Yan, S., Huang, Y.W. (2018). Optimal scheduling of a taxi fleet with mixed electric and gasoline vehicles to service advance reservations. *Transportation Research Part C: Emerging Technologies*, Vol.93, pp.479-500.
- Madsen, O.B.G., Ravn, H.F., Rygaard, J.M. (1995). A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. *Annals of Operations Research*, Vol.60, pp.193-208.
- Mahmoudi, M., & Zhou, X. (2016). Finding optimal solutions for vehicle routing problem with pickup and delivery services with time windows: A dynamic programming approach based on state–space–time network representations. *Transportation Research Part B: Methodological*, Vol.89, pp.19-42.
- Psaraftis, H.N. (1983). An exact algorithm for the single-vehicle many-to-many dial-a-ride problem with time windows. *Transportation Science*, Vol.17, pp.351-357.
- Schneider, M., Stenger, A., Goeke, D. (2014). The Electric Vehicle-Routing Problem with Time Windows and Recharging Stations. *Transportation Science*, Vol.48(4), pp.500-520.
- Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P. (1983). Optimization by Simulated Annealing. *Science*, Vol.220(4598), pp. 671-680.
- Yu, V.F., Purwanti, S.S., Perwira Redi, A.A.N., Lu, C.C., Suprayogi, S., Jewpanya, P. (2018). Simulated annealing heuristic for the general share-a-ride problem. *Engineering Optimization*, Vol.50(2), pp.1-20.
- Yu, V. F., Redi, A. A. N. P., Hidayat, Y. A., Wibowo, O. J. (2017). A simulated annealing heuristic for the hybrid vehicle routing problem. *Applied Soft Computing*, Vol.53, 119–132.
- Yu, V.F., & Lin, S.W. (2015). A simulated annealing heuristic for the open location-routing problem. *Computers and Operations Research*, Vol.62, pp.184-196.

- Yan, S., & Chen, C.Y. (2011). An optimization model and a solution algorithm for the many-to-many car pooling problem. *Annals of Operations Research*, Vol.191(1), pp.37-71.
- Yan, S., Lin, C.K., Chen, S.Y. (2014). Logistical support scheduling under stochastic travel times given an emergency repair work schedule. *Computers & Industrial Engineering*, Vol.67, pp.20-35.