

淡江大學運輸管理學系運輸科學碩士班碩士論文

指導教授：邱顯明 博士

撥召服務最佳化指派作業之研究

A Study on the Optimal Assignment Operation for
the Dial-and-Ride Problem

研究生：黃漢瑄 撰

中華民國 95 年 6 月

論文名稱：撥召服務最佳化指派作業之研究

頁數：91

校系(所)組別：私立淡江大學運輸管理學系運輸科學碩士班

畢業時間及提要別：94 學年度第 2 學期

碩士論文提要

研究生：黃漢瑄

指導教授：邱顯明 博士

論文摘要：

撥召服務系統具有傳統公車系統之服務容量並能提供高度之及門性與即時性，因此除了一般載客之外常被用以服務行動不便者，例如老年人、病患與身心障礙者等。其服務方式須由顧客先行預約，再透過系統將數筆預約進行路線排程。撥召服務問題即為求解此一型式之運輸系統路線排程之車輛繞徑問題。

在問題之求解上，諸多研究皆使用節點再插入法(Vertex Reinsertion)作為臨近解搜尋之方法，然而傳統逐點插入法在搜尋上之效率不佳，因此許多研究皆設計搜尋範圍之限制機制。雖然以再插入法進行搜尋之方法已為撥召服務問題之主流作法，然而在各類車輛繞徑問題中廣被應用之各類巨集啟發式演算法卻鮮少應用於撥召服務問題，

本研究的主要目的在於嘗試將基因演算法(Genetic Algorithm)與蟻群演算法(Ant Colony Optimization)用以求解多車輛撥召服務問題。除了探討此兩類演算法在最佳化撥召服務問題應用上之可能性外，在求解流程上將區分為分群與繞徑兩階段求解以及整體一階段求解兩種形式，並組合兩類演算法以找出最適之求解方式與演算法設定。透過例題之求解發現，以變異性控制適應性基因演算法(Diversity-controlling Adaptive Genetic Algorithm)其結果較佳，而蟻群演算法求解速度較快。

另外，本研究亦將撥召服務問題做一隨機處理，將路網旅行時間加入動態之觀點，並評估於動態旅行時間下，撥召服務問題適之路線接受策略。經過例題求解比較發現，以新舊路線差異為新路線長度之 20%作為接受門檻值之做法，其求解結果較為集中，是為較佳之排程策略。

關鍵字：撥召服務問題、隨機性車輛路線問題、基因演算法、蟻群演算法

A Study on the Optimal Assignment Operation for the Dial-and-Ride Problem.

Key word : Dial-and-Ride Problem, Stochastic Vehicle Routing, Ant Colony, and Genetic Algorithm.

Name of Institute : Graduate Institute of Transportation Science, Tamkang University

Graduate date : June 2006

Degree Conferred : Master

Name of student : Han-Hsuan Huang

Advisor : Dr. Hsien-Ming Chiu

黃漢瑄

邱顯明博士

Abstract :

The Dial-a-Ride system can provide door-to-door and real-time service with the capacity of the traditional bus system. It is special transportation service for the handicapped citizen. The service usually is carried by the advanced order of passengers, and routing plan of service is scheduled based on these demands. The Dial-a-Ride Problem (DARP) is a vehicle routing problem for the arrangement of the Dial-a-Ride service.

The DARP has been proved to be a NP-Hard problem, therefore, most researches addressed this problem adopted heuristic methods as the solution methods. "Vertex Reinsertion" has been adopted by previous studies. Although, it is popular solution procedure, its computation efficiency is not impressed. With tremendous interest of the Meta-heuristics on the combinatorial problem, there is few study discuss its application on the DARP. The purpose of this study is to evaluation of the application of the Genetic Algorithm (GA) and the Ant Colony Optimization (ACO) on the DARP.

Two solution procedures are proposed in this study, i.e., an integrated approach and cluster first route second approach. A series of revised GA and ACO algorithms are developed for these two approaches. A series of case studies with different characteristics such as demand density, demand size were used to test the solution capability of the proposed algorithms. Based on the results of the case studies, the Diversity-controlling Adaptive Genetic Algorithm is identified as the best algorithm in solution quality, and the Ant Colony Optimization is proved to be able to solve the problem quickly.

In addition, the DARP with stochastic travel times has been test in the final part of the analysis. Different types of the decision criteria have been used to identify good decision criteria for the dynamic DARP. Based on the test results, threshold value is 20% of the difference of sum routing durations between new and old routing plans has been proven to be better decision criteria for the dynamic DARP.

誌謝

很想來個謝天，只因要感謝的人實在太多了。

首先要感謝恩師 邱顯明老師三年來對我的教導與忍耐。老實講我並不是一個好學生，常常因為外務而沒有將心神一直維持在論文上，甚至因此而延畢了一年，即便如此，邱老師還是在每次 meeting 時不厭其煩的指導我應該前進的方向。這篇論文最後得以完成，真的非常感謝邱老師的教導。同時必須感謝中央大學 顏上堯教授與元智大學 丁慶榮教授於口試時巨細靡遺的建議與指正，並提出許多在撰寫時未曾注意的疏漏，使本論文更加完整。

這三年來系上老師與助教在各方面的教導與協助亦是本論文的推手。論文題目起源於陶老師所主持的計畫案，參與該項計劃讓我有接觸這方面文獻的機會；胡老師則是我在大學時代的班級導師，在所裡這三年間亦受到老師許多照顧；小張老師在我參與計劃時的許多建議與指導，亦對我在論文構思上有不少幫助；還要感謝羅老師的認真、董老師的嚴謹、劉老師的細心、范老師的慈祥、石老師的博學以及陳老師的教誨。另外還要感謝孔助教與張助教在各項事務上的協這，讓我的求學生涯得以順利完成。

這三年中學長姐、同學與學弟妹也給我不少協助，第一年時有惠鈺學姊、忠榮學長還有復爺的指導，前兩年有書嫻、聖偉、鈺雯、啟函、士傑、禹辰、瑋琦、正宏、孟甫、宏銘、老賴的陪伴，最後一年則受到阿吉、淑芳與伊婷的協助，非常感謝。學校外的朋友無論在生活或精神上亦給我不少協助與鼓勵，Blacky 在我發生車禍的時候幫忙處理後續事宜，讓我隔天能夠準時參加考試；V 老在電腦軟硬體上的協助讓我的電腦始終服服貼貼；而 Muse、Linder、DT 與阿伯等人則是在我心灰意冷時給我鼓勵，因為有這些朋友的協助，這本論文才得以順利完成。

最後必須感謝家人的支持，老爸、老媽跟老哥不只在經濟上給我支援，亦在我心情不好發牢騷的時候予以包容，讓我能夠順利完成求學階段。

謹以此本論獻給來不及見到就過世的阿公，以及外公與外婆。

漢瑄 謹誌於淡水

2006、7、15

目錄

頁次

中文摘要	
英文摘要	
誌謝	
目錄.....	I
圖目錄.....	V
表目錄.....	IV
第一章 序論.....	1
1.1 研究動機.....	1
1.2 研究目的.....	2
1.3 研究範圍.....	3
1.4 研究架構及流程.....	4
1.5 章節配置.....	6
第二章 文獻回顧.....	7
2.1 撥召服務問題.....	7
2.1.1 撥召服務.....	7
2.1.2 撥召服務問題.....	8
2.2 基因演算法.....	12
2.2.1 基因演算法概述.....	12
2.2.2 基因演算法相關研究應用.....	13
2.3 蟻群演算法.....	14
2.3.1 蟻群演算法概述.....	14
2.3.2 蟻群演算法相關研究應用.....	15
2.4 小結.....	16
第三章 模式探討.....	17
3.1 撥召服務問題描述.....	17
3.2 基本假設.....	18

3.3 模式構建.....	19
3.3.1 符號介紹.....	20
3.3.2 數學模式.....	21
3.4 小結.....	24
第四章 求解策略.....	25
4.1 繞徑策略.....	25
4.2 求解流程.....	28
4.2.1 完整一階段流程.....	28
4.2.2 分群繞徑二階段流程.....	30
4.3 演算法設計.....	32
4.3.1 適應性基因演算法.....	32
4.3.2 族群競爭式基因演算法.....	37
4.3.3 蟻群演算法.....	39
4.4 小結.....	43
第五章 案例測試.....	44
5.1 求解能力.....	44
5.2 參數設定.....	46
5.2.1 適應性基因演算法.....	57
5.2.2 族群競爭基因演算法.....	50
5.2.3 蟻群演算法.....	54
5.2.4 參數一覽.....	58
5.3 模式參數.....	58
5.3.1 路線最大長度.....	59
5.3.2 最大乘車時間.....	60
5.4 結果比較.....	61
5.4.1 小型問題 pr1 結果比較.....	62
5.4.2 大型問題 pr5 結果比較.....	63
5.4.3 文獻結果比較.....	64
5.5 小結.....	68
第六章 隨機旅行時間問題分析.....	67
6.1 動態旅行時間.....	68

6.1.1 旅行時間之計算.....	68
6.1.2 旅行時間統計分配假設.....	69
6.1.3 亂數產生之方法.....	70
6.2 撥召服務路線排程策略.....	70
6.2.1 路線之重新排程.....	70
6.2.2 排程接受策略.....	71
6.3 策略比較.....	73
6.3.1 測試結果分析.....	73
6.3.2 測試結果小結.....	75
第七章 結論與建議.....	76
7.1 結論.....	76
7.2 建議.....	77
參考文獻.....	79

附錄 求解結果表



圖目錄

圖 1-1 研究流程圖.....	4
圖 2-1 撥召服務路徑排程示意圖.....	8
圖 3-1 花型路線示意圖.....	18
圖 4-1 車輛於各節點之到達時間、停等時間、開始服務時間與離開時間示意圖.....	25
圖 4-2 不同到達與離開策略比較圖.....	26
圖 4-3 完整一階段求解流程圖.....	29
圖 4-4 分群繞徑二階段求解流程圖.....	30
圖 4-5 適應性基因演算法求解流程圖.....	33
圖 4-6 基因演算法起始解產生方式.....	34
圖 4-7 Uniform 交配示意圖.....	34
圖 4-8 不同求解流程基因演算法染色體編碼示意圖.....	36
圖 4-9 族群競爭基因演算法示意圖.....	37
圖 4-10 族群競爭基因演算法求解流程圖.....	38
圖 4-11 蟻群演算法螞蟻構建路線示意圖.....	39
圖 4-12 蟻群演算法求解流程圖.....	40
圖 4-13 完整一階段候選名單編排示意圖.....	41
圖 5-1 簡單問題之需求狀況與最佳繞徑圖.....	45
圖 5-2 適應性基因演算法、族群競爭式基因演算法與蟻群演算法簡單問題求解結果分佈.....	46
圖 5-3 適應性基因演算法各族群數求解結果分佈.....	48
圖 5-4 適應性基因演算法各迭代數求解結果分佈.....	49
圖 5-5 族群競爭基因演算法各迭代數求解結果分佈.....	50
圖 5-6 族群競爭基因演算法各迭代數求解結果分佈.....	51
圖 5-7 族群競爭基因演算法各迭代數求解結果分佈.....	52
圖 5-8 族群競爭基因演算法各基因樣板保留率求解結果分佈.....	53
圖 5-9 蟻群演算法螞蟻隻數求解結果分佈.....	55
圖 5-10 蟻群演算法螞蟻隻數求解結果分佈.....	56
圖 5-11 蟻群演算法費洛蒙起始濃度求解結果分佈.....	57
圖 5-12 路線長度違反成本起始值設定求解結果分佈.....	59
圖 5-13 路線長度違反成本起始值設定求解結果分佈.....	60
圖 6-1 隨機旅行時間問題決策流程.....	67
圖 6-2 動態旅行時間路線排程處理流程.....	71
圖 6-3 門檻值百分比測試結果分佈圖.....	72
圖 6-4 不同求解策略旅行時間分佈.....	75

表目錄

表 2-1 撥召公車問題相關文獻彙總表.....	11
表 4-1 移除與重劃步驟示意.....	35
表 5-1 簡單問題節點設定.....	44
表 5-2 簡單問題最佳繞徑排程.....	45
表 5-3 簡單問題各演算法求解結果.....	46
表 5-4 適應性基因演算法調整參數範圍表.....	47
表 5-5 適應性基因演算法各參數設定表.....	47
表 5-6 適應性基因演算法族群大小比較表.....	48
表 5-7 適應性基因演算法迭代數比較表.....	49
表 5-8 適應性基因演算法調整參數範圍表.....	50
表 5-9 族群競爭式基因演算法族群大小比較表.....	50
表 5-10 族群競爭基因演算法迭代數比較表.....	51
表 5-11 族群競爭基因演算法突變率比較表.....	52
表 5-12 族群競爭基因演算法基因樣板保留率比較表.....	53
表 5-13 蟻群演算法調整參數範圍表.....	54
表 5-14 蟻群演算法各參數設定表.....	54
表 5-15 蟻群演算法螞蟻隻數比較表.....	55
表 5-16 蟻群演算法迭代數比較表.....	56
表 5-17 蟻群演算法費洛蒙起始濃度比較表.....	57
表 5-18 各演算法參數比較結果.....	58
表 5-19 路線長度違反成本起始值設定比較表.....	59
表 5-20 乘車時間違反成本起始值設定比較表.....	60
表 5-21 結果分析選用例題規模.....	61
表 5-22 小型問題總路線長度各分組 t 檢定結果比較.....	62
表 5-23 小型問題總路線長度整體 t 檢定結果比較.....	62
表 5-24 大型問題總路線長度各分組 t 檢定結果比較.....	63
表 5-25 大型問題總路線長度整體 t 檢定結果比較.....	63
表 5-26 例題 pr1 結果表.....	64
表 5-27 例題 pr5 結果表.....	65
表 6-1 門檻值百分比測試結果.....	72
表 6-2 門檻值 t 檢定成對比較結果.....	73
表 6-3 不同策略解集合範圍.....	74
表 6-4 不同策略 t 檢定結果.....	74
表 6-5 不同策略 F 檢定結果.....	74

第一章 緒論

1.1 研究動機

小汽車持有率過高與大眾運輸系統使用率不足以及其所衍伸的環境污染、道路雍塞與運輸容量之浪費等問題，一直是諸多國家在交通管理上所面臨的重要課題，然而小汽車所帶來的個人滿足感與移動上具便利性與及門性卻非一般大眾運輸系統可以取代的。為此大眾運輸系統管理上常以較為低廉的票價作為誘因，利用經濟上的手段藉此設法吸引更多潛在使用者，然而此種治標不治本之作法卻常常造成營運者與管理者更大的財務問題。

近幾年來由於環境與運輸之永續概念持續發展，再加上各種智慧型運輸系統之輔助，塑造出許多新型態之大眾運輸系統與準大眾運輸系統，例如在歐美逐漸成形的各類汽車共乘(Carpool 或 Rideshare)與汽車共用(Car share)服務。車輛共乘為利用各種通訊技術將用路人集合並組成共乘群組(Carpool Group)，並由其中一人駕車於各人之起迄點接送其他組員；汽車共用則是更為先進之延伸式租車服務，使用者僅需負擔些許費用便可於各地之車場自由選用並駕駛車輛。

1970 年代於美國發展出之撥召公車系統(Dial-a-Ride)為改善傳統公車運輸受於固定路線與班次之限制所產生之運輸系統，能夠以較為低廉的費用提供計程車之及門性與即時性，但仍保有公車系統之服務容量與高乘載之特性。使用者僅需事先預約並提供起迄點相關資訊，系統便會將其加入運行路線中進行接送。由於撥召公車具有公車之乘載量與計程車在路線上之彈性，因此如何安排一至數條成本最小之路線並滿足先接載後送達等各項限制即為其營運上之基本問題，而設計並求解此種問題之最佳化過程即稱為「撥召服務問題(Dial-a-Ride Problem, DARP)」。撥召服務問題多與起迄點時窗、車輛容量以及最長路線等各項限制結合，並依照目標式進行最佳化求解，以取得最佳之路線與乘客載送組合。此種問題在求解上與具時窗之撿取配送問題(Pickup and Delivery Problem with Time Windows, PDPTW)相當類似，然而撥召公車由於服務對象為具有感受之乘客，因此必須考慮乘客於系統中之舒適程度(Cordeau, 2003)。因此本研究將根據撥召公車服務之特性構建能反應出顧客舒適程度之接送繞徑模式，並考慮其運輸型態以設計相關之巨集啟發式(Meta-Heuristics)求解演算法，進而發展出一套完整且有效之解題程序。

1.2 研究目的

撥召公車具有傳統公車系統之服務容量並能提供高度之及門性與即時性，因此除了一般載客之外常被用以服務行動不便者，例如老年人、病患與身心障礙者等。服務方式始於顧客預約，經過系統之排程處理後將形成包含數筆繞徑路線之完整接送計畫，因此一般撥召服務問題之需求形式包含顧客之起迄點位置與兩點之時窗，故在配送形式以及求解過程上與具時窗之檢取配送問題相當類似，在模式上撥召服務問題又常被視為具時窗之一般化檢取配送問題之特化形式(Cordeau, 2003)，然由於其所乘載對象為具有感知之人，在模式上必須考慮顧客不舒適度之反應，因此撥召服務問題與一般檢取配送問題之相異點除必須於模式中考慮顧客不舒適度之外，其每筆需求之需求量皆為 1 人。

在求解流程上，撥召服務問題具有 3 大決策步驟：決定各車輛所服務之顧客、編排各路線中之顧客順序以及各起迄點之繞徑排程，其中又以繞徑排程為最重要之工作(Cordeau, 2003)，為此國內外文獻多以路線排程作業作為區分點將整體作業流程區分為分群與單路線繞徑兩階段，或是以產生起始解之後逐步改善之方式調整排程內容。有鑑於此，本研究在求解流程上亦區分為分群與繞徑兩階段求解以及整體一階段求解兩種形式，以探討求解撥召服務問題之最適作業方式。

早期撥召服務問題之文獻中，Psaraftis(1980)曾利用動態規劃求解單一車輛無時窗之問題形式以求出最佳之繞徑排程，然而雖透過最大位置偏移(Max. Position Shift)限制減少計算次數，但對於中大型問題仍耗時甚鉅，在當時僅能處理 9 人以下之需求。由於撥召服務問題為一種 NP-hard 問題，後續研究多以啟發式解法進行最佳化求解，Jaw et al.(1986)為最早提出適用於靜態多車輛撥召服務問題之啟發式解法之學者，利用搜尋可行之插入區間之方式進行成本最小化求解，並將問題規模擴展至 2617 人之大型問題，此後大部分啟發式解法皆使用節點再插入法(Vertex Reinsertion)作為臨近解搜尋之方法。另一研究重點在於傳統逐點插入法在搜尋上之效率不佳，因此諸多研究皆設計出搜尋範圍之限制機制，例如 Toth et al.(1997)利用 Tabu Thresholding 排除表現不加之移步，而 Cordeau et al.(2003)則使用禁忌過程與關鍵節點技術來減少不必要之搜尋。

雖然以再插入法進行搜尋之方法已成為求解撥召服務問題之主流作法，然而在各類車輛繞徑問題中廣被應用之各類巨集啟發式演算法卻鮮少應用於撥召服務問題，因此本研究嘗試將基因演算法(Genetic Algorithm)與蟻群演算法(Ant Colony Optimization)針對問題型態加以調整，提出改良過後之演算法以求解多車輛撥召服務問題。研究目的除了探討此兩類演算法在最佳化撥召服務問題應用上之可能性外，在求解流程上將區分為分群與繞徑兩階段求解以及整體一階段求解兩種形式，並組合兩類演算法以找出最適之求解方式與演算法設定。故本研究之目的敘述如下：

1. 探討撥召公車作業所應具有之特性。
2. 構建符合撥召公車作業之成本模式，以及足以反映顧客滿意程度之舒適度表現方式。

3. 設計應用於撥召服務問題各求解流程形式與階段之巨集啟發式解法。
4. 透過例題求解比較分析找出最佳之求解方式與演算法設定。
5. 探討於動態旅行時間下撥召服務問題最適之繞徑排程策略。

1.3 研究範圍

撥召服務問題雖與撿取配送問題皆屬於一般化撿取配送問題(General Pickup and Delivery Problem)之特例(Savelsbergh, 1995)，同為求解車輛由場站出發服務具有起迄點各一之多筆需求，完成服務之後再回到場站之最適化車輛繞徑問題，但由於乘載對象不同，在需求特性與模式設計上兩者具有顯著的差異性。本研究所探討之主要問題為車輛自場站中出發，行至需求起點接載顧客，並將其運送至指定迄點之車輛繞徑問題。由於撥召服務問題相關因素頗多，因此本研究主要範圍與假設如下：

1. 場站

本研究僅考慮位在某一特定位置之場站，所有車輛於該場站出發，完成繞徑之後則返回。

2. 車輛

本研究僅針對單一車種，故各車之容量限制相同，每一車輛由場站出發，服務所指定之顧客之後再返回場站。

3. 車隊

車隊所擁有之車輛數為已知，由於每車次發車具固定成本，多使用一車次之成本將於模式中反應。

4. 顧客

顧客會提供系統關於起迄點等需求資訊，這些資訊包括起迄點位置、起迄點時窗以及上下車所需服務時間。所有需求資料為已知。

5. 需求量

撥召服務問題設定上每筆需求之需求量皆為 1，若對象為多人之需求則劃分為與需求量相同之數筆需求處理。

6. 需求點

每位顧客包含起點與迄點兩筆需求點資訊，各需求點僅由一部車輛服務且路線中起點之位置必須先於迄點。所有需求點必須皆有車輛服務。

7. 停等(Idle)

撥召公車為滿足各需求點之時窗限制將允許車輛進行停等之動作，本研究視停等為發生

於開始服務該節點之乘客之前，即車輛到達節點與開始進行服務之時間差。

1.4 研究架構及流程

本研究流程如圖 1-1：

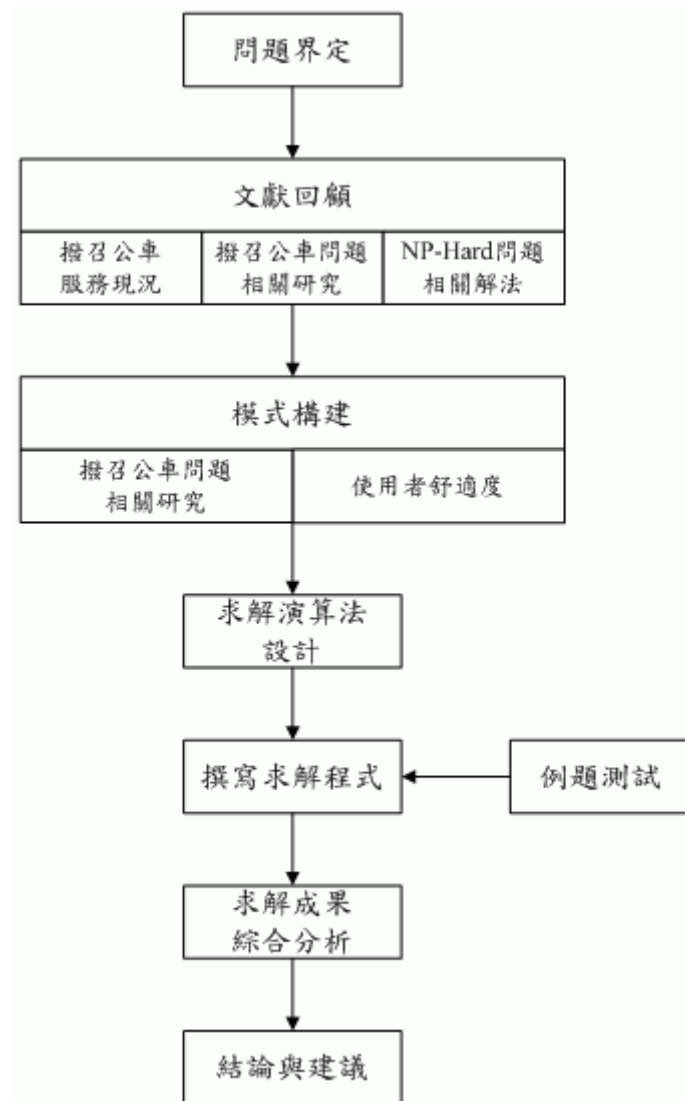


圖 1-1 研究流程圖

1. 問題界定

分析撥召公車服務下之整體操作與車輛排程作業之流程與特性，以確立研究背景、動機、界定研究範圍，並擬定完整之研究架構。

2. 文獻回顧

文獻回顧以服務現況、相關研究與求解演算法作為回顧主題。服務現況分析撥召公車服務之特性，用以構建足以反應系統之模式；相關研究則探討現階段各文獻之求解效率與品質，並歸納其成果作為求解成果分析比較之對象；最後則尋找可應用之 NP-Hard 問題求解演算法，並從中挑選較具求解潛力之作法。

3. 撥召公車配送模式探討

此階段為針對車輛共乘配送模式進行探討，並建立各相關之限制條件式以充分反應共乘服務環境，此部份所要探討的包括模式之基本假設及基本架構。

4. 模式構建

在求解結果上本研究採用較具普遍性之總路線長度作為比較依據，另外於求解時本研究設計一違反值模式作為各解優劣程度之評判基準，此種模式在求解過程中允許鬆弛解存在，可增加搜尋之廣度，而其中各成本函數內容則依據文獻回顧之結果做進一步之擬定。在懲罰成本之設定上採用 Cordeau et al. 所使用之自我調整非負參數，此種參數將於迭代中根據所對應之成本函數值進行調整，在可行性之收斂上較具彈性。

5. 撰寫求解程式

依據模式內容與演算法步驟撰寫求解程式，以分析各演算法之求解效益。程式利用 Visual Basic 6.0 針對完整一階段與分群繞徑兩階段求解流程進行撰寫，以便於進行不同演算法之組合求解分析。

6. 求解結果綜合分析

演算法評估部分本研究採用 Cordeau 與 Laporte 系列研究所使用之例題進行求解運算與結果分析，評估部分包括各演算法於各參數設定下以及兩種不同求解流程於各種演算組合下之求解績效，最後並與 Cordeau 與 Laporte 以及採用相同例題之 Bergvinsdottir 結果進行比較。

7. 結論與建議

整理研究經驗與心得，並提出具體結論與建議後續研究可發展方向。

1.5 章節配置

本論文之後續章節安排如下：

第一章：緒論。說明本研究之動機、目的、範圍與流程。

第二章：問題描述與文獻回顧。探討撥召公車之營運現況、撥召服務問題之研究概略以及可能利用之求解方式，完整界定出本研究所探討之問題與相關特性，以利後續步驟。

第三章：模式構建與研究設計。依據第二章對問題特性之描述，構建符合現況之數學規劃模式以作為後續求解之主要依據，並選出較具解題潛力之演算法進行求解。

第四章：求解策略與方法。針對所選之各演算法發展能有效求解撥召服務問題之演算步驟，並敘述解題之策略、流程與方法以及擬定分析之項目與步驟，再於後續章節以例題測試驗證演算法之正確性。

第五章：案例測試。先以一簡單例題驗證演算法之求解能例，再透過例題測試進行各演算法參數之比較分析以找出最佳之解題設定，並比較不同懲罰成本設定對於求解過程之影響，以及進行不同演算法於二階段流程之組合比較。

第六章：動態旅行時間。將路網旅行時間以動態方式表示，並探討撥召服務問題於動態旅行時間下最佳之繞徑排程策略。

第七章：結論與建議。整理研究經驗與心得，並提出具體結論與建議後續研究可發展方向。

第二章 文獻回顧

1970 年代於美國發展之撥召公車服務為具有路線彈性之大眾運輸系統，本身不具固定路線之限制且能將乘客於指定之起迄點間進行運輸之特性，大幅提升傳統運輸系統所欠缺之即時性與及門性，而乘客亦能保有高度之自主性，故撥召公車常被設計成為用以服務老殘等行動不便者之無障礙運輸系統。由於撥召公車是以顧客需求為導向之大眾運輸方式，其營運成功之關鍵在於系統功能是否有效發揮，能否滿足使用者之服務水準要求(邱裕鈞，民 92)，故相關研究多針對系統面進行設計，或由營運面上針對其所發展出之車輛繞徑問題形式進行研究。由於本研究之目的在於利用巨集啟發式解法以探討撥召服務問題之最佳化排程方式，因此本章之重點為撥召服務問題研究文獻以及巨集啟發式解法之發展進行回顧。

2.1 撥召服務問題

2.1.1 撥召服務

撥召服務之進行方式為乘客主動聯繫控制中心告知其起迄點等需求，並由控制中心進行路線規劃，再指派車輛進行接送以完成運送行為。最常見之撥召運輸服務為招呼式計程車，由乘客打電話至車行進行預約叫車，並由車行派車前往指定之起點載送乘客至指定之迄點。此種情況車輛通常僅服務一名或起迄點位置與時間等需求狀態皆相同之一群乘客，因此乘客所具有之自主性相當高，若車行未能在其所能忍受之等候時間內派車前往接送，乘客極有可能尋求其他車行或運輸服務。因此計程車撥召服務之重點，對於車行業者而言在於最小化操作成本並服務最多之乘客，對於顧客而言決定服務水準之因素在於最小化之等待時間。

另一種撥召服務為針對特定客層所提供之運輸服務，國內常見之復康巴士即為此種類型撥召服務。有此種服務需求之乘客亦必須事前提出預約，與計程車不同的是，一般而言此類型之撥召服務所使用之車輛為小型巴士等較為大型之車輛，為因應不同類型之需求可能會設置輪椅升降裝置等特殊設施，一台車輛可能僅服務一名乘客，或是具有混合內裝以服務多名乘客，在路線排程上須考慮不同需求之乘客。

由於車輛派遣次數越多所造成之派車成本越多，為了有效減少操作成本，撥召服務必須引入車輛共乘之特性於一條路線中服務多名乘客，於乘客之起迄點路線間穿插其他乘客之起迄點以減少實際之派車次數。經過共乘路線排程之撥召服務將能有效減少所派出之車次，然而車輛在行駛時所產生之成本將會隨著路線距離與時間長度增加，所有車輛亦具有最大行駛長度限制，故此種形式之撥召服務在路線規劃時亦必須考慮最小化繞徑距離或時間長度，亦或是必須遵守車輛於此方面之限制。

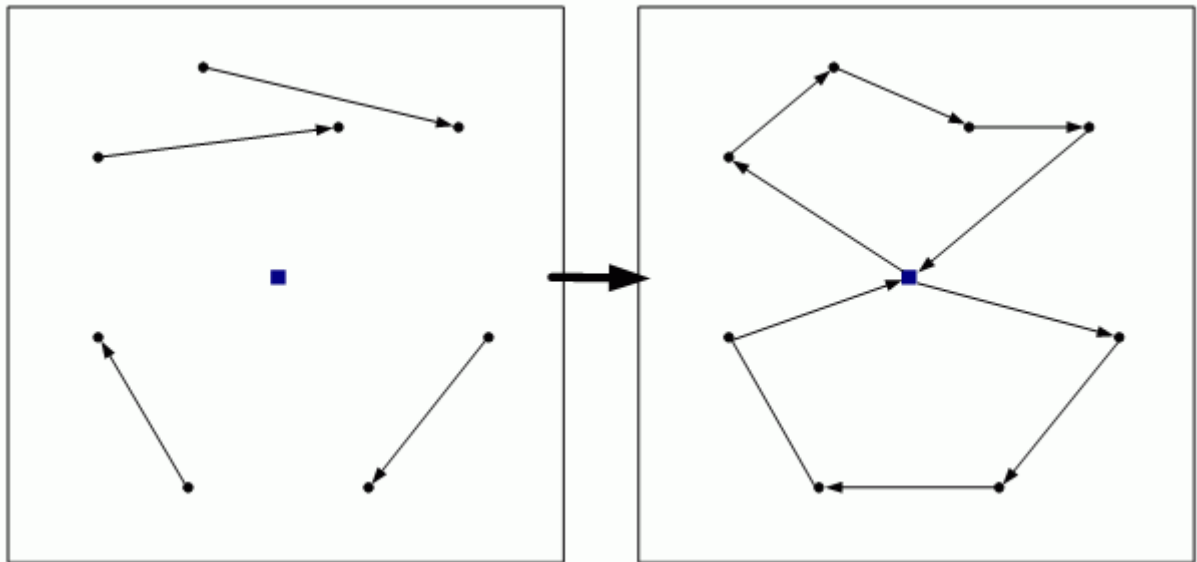


圖 2-1 撥召服務路徑排程示意圖。乘客之起迄點直接路線如左圖，經撥召服務系統進行路線排程後將構成右圖之兩條路線。

具共乘性質之撥召服務如撥召公車在服務型態上與招呼式計程車差距甚大，由於需要離開原本行進路線以進行其他乘客之接送，且此種服務因具有大眾運輸之性質使得乘客在服務方式上之限制較多，亦不再具有招呼式計程車所含有之高度服務水準，然而由於撥召服務為運送乘客之運輸服務，因此仍必須考慮服務水準之評估。在服務水準之考量上，此類撥召服務最常使用之指標為時窗遵守與否以及超出原本旅程之時間多寡，另外，路線極可能因為需要滿足其他節點之時窗限制而必須有中途停車等待之狀態，此部份即所謂停等時間(Waiting Time)。停等之發生無異會減少乘客之舒適度並降低服務水準，甚至會造成乘客無法接受之狀況，然而在最佳化繞徑排程中，為求滿足時窗限制，發生停等將無可避免，故在撥召服務問題中常以假設之方式允許車輛於繞徑中發生停等。由於影響撥召服務之服務水準的因素眾多，故在撥召服務問題操作下常針對各種顧客舒適度指標進行限制，並配合最小操作成本模式進行最佳化路線編排。

2.1.2 撥召服務問題

撥召服務問題為求解包含數條服務具有各自起迄點之多名乘客之車輛繞徑問題，車輛於路線運行中必須在遵守各點之時窗限制下將乘客從起點運送至迄點。問題之目標為最小化總運輸操作成本與乘客之不舒適度(Bergvinsdottir, 2004)，決定前者之因素為車輛容量與車隊規模，而乘客舒適度常使用時窗與乘車時間以及停等時間進行評估。Cordeau et al.(2003)曾指撥召服務問題之特性為：

1. 所有繞徑路線起始於一場站。
2. 每位乘客之起迄點需求由相同車輛服務且起點需求必先於迄點需求。
3. 車輛容量限制、繞徑時間等限制不得違反。

4. 所有時窗限制必須滿足。
5. 允許車輛於起始地點或是任何服務之起迄點進行停等。
6. 最小化總繞徑成本。

早期撥召服務問題基本上是由旅行推銷員問題(Traveling Salesman Problem, TSP)加上起迄點關係等限制條件所形成(游進俊, 民 82), 此種問題為單車輛撥召服務問題, 近期研究則以多車輛問題為主要形式。問題中之乘客需求一般包含起迄點位置資訊以及時窗限制, 在進行繞徑排程時須將起迄點先後順序與時窗等限制一併加以考量, 在整體求解流程上與同屬於一般化撿取配送問題(General Pickup and Delivery Problem)特化之撿取配送問題(Pickup and Delivery Problem Vehicle Routing Problem, PDVRP)相當類似(Savelsbergh, 1995), 在模式上亦與撿取配送問題以及具時窗之車輛繞徑問題(Vehicle Routing Problem with Time Windows, VRPTW)雷同, 然而所有需求數量固定為 1 且評估運輸水準之乘客不舒適度指標須於模式中一併考量, 此為撥召服務問題與其他類型車輛繞徑問題最主要之差異。

相關研究中 Psaraftis(1980)曾利用動態規劃之方式進行求解單車輛無時窗撥召服務問題, 使用最大位置偏移(Max. Position Shift)以限制乘客預約順序與實際被接送順序間之差異, 目的在維持服務水準並有效減少動態規劃計算次數, 所使用之目標式為最小化總行駛時間, 並以最小化乘客之候車與乘車時間作為滿意度指標。求解使用動態規劃法以求得最佳解, 然而其所需之求解時間呈 $O(n^2 3n)$ 增加, 對於大型問題耗時甚鉅, 在當時於合理之時間內僅能求解 9 人以下之需求。其後 Psaraftis(1983)則提出另一套啟發式解法用以求解相同類型之問題, 以最小擴張樹(Minimum Spanning Tree, MST)並透過繞行順序修正以構建起始路線, 之後以節線交換法搜尋較佳路線, 在搜尋之過程中仍須遵守「先接載後送達」之次序性限制以維持解之可行性。靜態單車輛無時窗之問題形式之後由游進俊(民 82)針對起始解構建與較佳解尋優兩階段進行改良, 第一階段之起始解採用最節省插入法、最近插入法、最鄰近點法與點對插入法四種做法進行構建, 並於第二階段利用兩種節線交換之方式進行尋優改善, 其一為搜尋最佳之節線交換情況之橫向優先搜尋, 其二為變更最先到達點之縱向優先搜尋。結果顯示以最鄰近點法搭配橫向優先搜尋之作法表現較佳。

在無時窗多車輛撥召服務問題之研究上, Stein(1978)曾提出以轉運點之方式將服務範圍切分為等同為車輛數之數個區域, 於每個區域設立一轉運點並以一部車輛進行服務, 且以另一部車輛巡迴於各轉運點間以形成整體服務路網。Jaw 等人(1986)則為最早使用啟發式解法求解具時窗多車輛之靜態問題之學者之一, 其作法為一兩階段啟發式解法, 改善方式為利用插入法並依兩階段進行搜尋: 先找出所有可供插入之可行時間區間, 再於這些區間中尋找目標值最小者進行節點插入。可行區間之判別作法為須符合乘客最大乘車時間與時窗限制, 並於可行區間內先插入乘客之起點, 再於起點之後之可行區間搜尋迄點之位置。其最小化目標式為總繞徑長度, 並配以乘車時間與時窗限制反應旅客之不舒適度, 另外在操作上不允許車輛處於載客狀態下時進行待機以維持服務水準。

Toth 與 Vigo(1997)則利用插入法搜尋結合禁忌門檻(Tabu Thresholding)求解真實資料之殘障者撥召服務問題。其車隊由裝設相關設備之特殊小型巴士與車輛構成, 並由計程車補足車輛容量所無法乘載之乘客。求解過程採用二階段法, 第一階段為顧客分群並進行各車路線編

排，此部分配合禁忌門檻以排除表現過差之搜尋移步，該階段完成後則以路線內節線交換與路線間乘客交換之方式進行改善。

Cordeau 與 Laporte 於 2003 年提出求解多車輛靜態撥招公車問題之啟發式演算法為一利用節點再插入法(Vertex Reinsertion)作為鄰近解搜尋方式之禁忌演算法，並使用自我調整非負參數(Self-Adjusting Positive Parameter)作為最小成本目標式中各違反值函數之懲罰值。其演算法使用亂數編排起始解，於迭代中互換兩路線中之兩位乘客需求起迄點作為移步並進行禁忌過程，每次迭代後根據目前解之限制式違反狀況調整各違反值，此過程中違反之限制式項目懲罰值將被放大以增加其在搜尋中之重要性。在經過一定次數之迭代後將針對每條路線進行路線內改善，作法為重新於路線內插入所包含之乘客起迄點，為有效減少此步驟之搜尋次數，Cordeau 等人使用一關鍵節點(Critical Vertex)判斷技術找出乘客起迄兩點中之關鍵節點，將其優先插入路線中，再根據其位置搜尋非關鍵節點之最適位置。Cordeau 等人根據實際大眾運輸系統狀況提出一系列例題，並透過此例題之求解以驗證演算法。

在巨集啟發式解法於撥召服務問題之研究上，Bergvinsdottir 曾使用基因演算法求解靜態多車輛撥招公車問題(2004)，在流程上屬於先做乘客分群再進行路線排程之二階段方法。由於該作法於分群階段使用基因演算法進行，因此在基因編碼上採用由各乘客所被派至之車輛編號所組成之數列，並由輪盤法選取母代與最佳個體之父代進行交配。交配方式採用 Pereira 所提出之 Uniform 交配運算元，所產生出之子代透過固定機率進行突變與否之判斷，若需突變則以單點位移之方式進行。第二階段則利用改良之 Time-Space Closest 法安排各車路線並進行適應度計算。模式上採用具懲罰值之最小操作成本模式，以限制式與懲罰成本反應顧客不舒適度，並以 Cordeau 與 Laporte 之例題進行驗證，結果顯示在乘車時間與停等時間上具有明顯改善。

在動態問題研究上，Teodorovic et al.(2003)曾利用模糊邏輯求解動態多車輛撥招公車問題，其演算法之設計目的在於即時處理大型問題之能力，有別於靜態問題所探討之路線最佳化作法。當新的需求產生後，根據各車服務該需求所增加之行車距離與停等時間，以及乘客加入此路線之後較原直接路線所增加之旅行時間與旅行距離，利用模糊邏輯判斷服務此顧客之績效，再決定是否接受該預約以及前往服務之車輛。測試結果發現該演算法具有快速求解 900 人 30 台車輛之大型問題之能力，且能在不違反現制條件下服務所有顧客。

另外 Attanasio et al.(2004)改良 Cordeau 等人所提出用以求解靜態問題之禁忌演算法以求解多車輛動態問題。於服務開始時先利用已存在之預約顧客構建基礎路線，新需求發生時以 Cordeau 等人之禁忌演算法流程平行搜尋最適插入路線，最後再對改變過之路線進行最佳化排程。為了加速演算法之運行以增加即時性，Attanasio 等並將模式所使用之自我調整非負參數調整至無限大以過濾任何違反限制式之不可行解。下表為撥召服務問題相關研究之彙總。

表 2-1 撥召公車問題相關文獻彙總表

作者	年份	問題類型	目標式	時窗類型	其他限制	求解演算法
Psaraftis	1980	單一車輛動態 DARP	最小化繞徑時間、顧客乘車時間與等候時間	無時窗	車輛容量 最大位置偏移	動態規劃，可求得最佳解
Psaraftis	1983	單一車輛靜態 DARP	最小化繞徑時間	上下車時間	車輛容量 最大位置偏移	動態規劃
Sexton Bodin	1985	單一車輛靜態 DARP	最小化加權下車時間差與乘車時間差之總和	上車與下車時間上限	車輛容量	啟發式解法求解繞徑與排程規劃
Desrosiers Et al.	1986	單一車輛靜態 DARP	最小化繞徑時間	上車或下車時間	車輛容量	動態規劃
游進俊	1993	單一車輛靜態 DARP	最小化路線長度	無時窗	車輛容量	先進行起始路線構建，再利用啟發式解法進行改善
Jaw Et al.	1986	多車輛靜態 DARP	多種乘客不舒適度函數非線性加總	上車或下車時間	車輛容量 乘車時間	插入法啟發式解法
Bodin Sexton	1986	多車輛靜態 DARP	最小化加權下車時間差與成車時間差之總和	上車與下車時間上限	車輛容量	啟發式解法求解繞徑與排程規劃
Toth Vigo	1997	多車輛靜態 DARP	最小化總服務成本	上下車時間	車輛容量 乘車時間	結合 Tabu Thresholding 與插入法之啟發式解法求解路線
Cordeau Laporte	2003	多車輛靜態 DARP	最小化總路線長度	上車或下車時間	車輛容量 最大繞徑時間 乘車時間	結合節點再插入(Vertex Reinsertions)之禁忌演算法
Cordeau	2003	多車輛靜態 DARP	最小化繞徑成本	上下車時間	車輛容量 最大繞徑時間	Branch-and-Cut
Diana Dessouky	2004	多車輛靜態 DARP	最小化加權總行駛距離、超時乘車時間與車輛待機時間	上下車時間	車輛容量	悔恨插入式啟發式解法求解 500 與 1000 筆需求之問題
Bergvinsdottir	2005	多車輛靜態 DARP	最小化操作成本與乘客不舒適度	上下車時間	車輛容量	基因演算法

資料來源：[10]，本研究整理。

2.2 基因演算法

2.2.1 基因演算法概述

基因演算法為 Holland 於 1975 年根據達爾文進化論所發展之隨機搜尋法，進化論之原理為「物競天擇、適者生存」，不適合生存之個體將會逐漸消失，最後仍存在於環境中之個體皆具有相當的適應能力。基因演算法為根據此種原理，將問題之解視為存在於大自然中之生物染色體，每次迭代中複製出一組適合生存之族群，並由族群內之染色體編碼相互交配以產生新一批之族群。舊有之染色體族群稱為父代，新產生之染色體族群則稱為子代，由於子代源於父代相互交配所產生，若經過長時間之演化，兩群體之染色體相似程度將會提高，為了讓群體具有再度進化之可能，基因演算法利用突變機制讓子代具有跳脫趨於穩定之染色體之能力。在每個迭代結束之後利用代表環境之適應度函數決定各染色體之生存能力，並進行「物競天擇、適者生存」之染色體汰換，經過反覆多次迭代之後將可尋找出適應度較佳之染色體。

基因演算法主要組成要素如下：

1. 編碼(Encoding)

在進行基因演算法尋優過程之前，必須先對問題之解進行染色體編碼，將解之內容轉換成由數個基因所組成之染色體，每個基因代表問題之特徵值(Feature Value)。一般常使用之編碼方式為二元型編碼及字串型編碼，其中二元型編碼為使用「0」與「1」之二進位方式構建染色體，多利用於數值型問題；字串型編碼則直接利用數字排列組成染色體，多應用於具順序關係之組合問題中。

2. 母體(Population)

母體為由數個染色體所構成之群體，初始母體即為第一代染色體，諸多研究多採用隨機方式產生，亦有配合啟發式方法或根據目前最佳解進行構建者。母體之大小取決於染色體之數量並對於求解效率具有直接影響，若母體過大將會耗費過多之搜尋時間；母體過小則會增加及早收斂之可能。一般母體大小為 30 至 200，較能兼顧求解效率與解之品質。

3. 適性度函數(Fitness Function)

適應度函數決定每一染色體於環境中之適應能力，即評估該染色體優劣程度之依據，故適應函數必須能夠區分各染色體之優劣，因此通常依系統針對求解問題之目的而設計。

4. 基因運算元

基因運算元為基因演算法最主要之部分，由複製、交配與突變三個基本運算元於迭代中重複運作以達到演化之目的。

(1) 複製運算元

複製運算元為利用適應度函數計算每一染色體之適應程度，並遵循預先設定之法則進行挑選，保留較佳之染色體進行交配以產生下一代並淘汰適應度較差之染色體。複製之主要想法為染色體之複製機率與其適應度成正比，但適應度之分布情形與問題類型有關，故比例複製並不一定合適，此種情況需採用適應度尺度轉換方式進行調整。

(2) 交配

交配運算元之作用為透過複製運算元所給予之機率，於父代族群中隨機選取兩者以組合成新一代之染色體。交配運算元於字串空間中進行有效搜索之同時必須降低對有效模式之破壞機率，以確保新子代染色體同時具有兩父代染色體中之優秀基因，使得子代之適應度較父代染色體為高。交配運算元為基因演算法有別於其他進化演算法之最大特徵，針對不同之染色體編碼方式所採取之交配策略與方式亦有所不同。

(3) 突變

突變為透過染色體本身基因排列之變換以激發群體中潛在之求解能力，其過程雖然會破壞基因迭代中染色體之穩定性，卻可擴大問題之搜尋空間。突變運算元為選取一染色體，由突變機率決定其突變與否，並於隨機產生之突變點改變染色體內部之基因資訊。此過程最重要之意義在於可增加母體內之變異程度，有助於跳脫陷入局部最佳解之母體。

5. 結束法則

基因演算法搜尋終止之條件為當所有染色體之基因排列趨於一致，使得不再有適應度更加之染色體出現時，然在實際運用上多以此四種作為搜尋終止之條件：

- (1) 已達到設定之最大演化迭代數。
- (2) 已達到設定之最大運算時間。
- (3) 已無法出現更優秀之解。
- (4) 已找出符合需求之解。

2.2.2 基因演算法相關研究應用

在各種組合最佳化問題中有相當多的研究使用基因演算法作為搜尋方法，其在各種車輛繞徑問題亦有多篇研究進行求解設計與求解能力之探討，然而在撥召服務問題中僅有 Bergvinsdottir(2004)曾應用基因演算法求解靜態多車輛問題。其所使用之二階段作法之第一階段採用基因演算法進行乘客之分群，在選出父母染色體之後，採用 Pereira 所提出之 Uniform 交配運算元進行交配，作法為亂數產生一串長度同於個體且由 0 與 1 構成的二元字串，其中 1 代表該基因遺傳自父代，而 0 則由母代基因加以替換。各子代透過預先設定之固定突變機率以決定是否需要突變，若需突變則以單點位移之方式進行。第二階段則利用改良之 Time-Space Closest 法安排，此法為一貪婪啟發式解法，其安排方式為：

- (1) 找出最早上車節點作為第一個行經之節點。
- (2) 若車上已有乘客，則找出時間與空間上最鄰近之下車點作為候選節點之一。

- (3) 若車輛尚有座位，則優先選取最鄰近之上車點，否則選取最鄰近之下車點直到四點。
- (4) 計算此四候選節點之成本值。
- (5) 選取成本值最小者作為下一到達點。

當子代路線完成構建之後將計算適應度並進行族群更新。Bergvinsdottir 模式上採用具懲罰值之最小操作成本模式，並以乘客停等時間、乘車時間以及時窗遵守與否反應乘客之不舒適度。在經過求解 Cordeau 與 Laporte 之例題並比較結果後發現，其所設計之基因演算法雖無法降低車輛總繞徑時間，但在乘車時間與停等時間上具有明顯改善。

在問題形式相近之具時窗檢取配送問題上，紀婉容等人(民 93)曾利用族群競爭式基因演算求解單車輛問題，此法為 Yang 於 2000 年所提出，其特點在於母體中之所有個體於迭代中將輪流擔任父代，並透過亂數選取之另一染色體進行交配並產生等同於父代族群大小之子代族群。由於族群競爭式基因演算法於一次迭代中之搜尋範圍較傳統基因演算法為廣，使問題在求解過程中更具多樣性與變化，進而提高找到可行解之機率以及最後產出之解之品質。

2.3 蟻群演算法

2.3.1 蟻群演算法概述

蟻群演算法最早由 Dorigo et al.(1991)根據螞蟻族群搜尋食物之現象所發展之巨集啟發式解法。螞蟻於搜尋食物時所殘留之特殊分泌物—費洛蒙(Pheromone)以告知其他螞蟻其所行進之路徑，當越來越多的螞蟻經過相同路徑之後，該路徑之費洛蒙濃度將隨之增加，而螞蟻久未行經之路徑其費洛蒙將逐漸揮發，因此每隻螞蟻可以根據費洛蒙資訊以尋找離食物最近之路線。Dorigo 等人根據此種現象提出螞蟻系統(Ant System)之概念，並用以求解旅行推銷員問題等各類繞徑與排程問題，並於 1997 年針對螞蟻系統進行部份修改而提出蟻群演算法(Ant Colony System)。

蟻群演算法之求解方式為利用一群人工螞蟻逐步構建出完整之解，並持續重複迭代以進行解品質之最佳化。螞蟻在構建解時主要依據為節線上之費洛蒙強度以及啟發式運算值兩類資訊，啟發式運算值之種類根據問題類型而有所差異，在各類型車輛繞徑問題中常以節線之長度做為此值，螞蟻即根據路線構建準則以及此二值所產生之轉換機率進行下一到達點之挑選。當一隻螞蟻完成路線構建之後即進行路線中各節線上之費洛蒙更新，而在每次迭代中所有螞蟻皆完成路線構建之後，再根據「菁英策略」加強最佳路線之費洛蒙濃度並進行整體之費洛蒙揮發計算。其演算法操作步驟為：

1. 起始狀態設定

包括設定費洛蒙起始濃度與揮發速度、螞蟻隻數、迭代次數、轉換規則參數以及 α 與 β 值。

2. 構建螞蟻路線

每一隻螞蟻根據路線構建準則產生一條完整路徑，通常為使用 \arg 值以及轉換機率函數配合亂數產生下一到達點之方式逐步構建，其中 \arg 值與轉換機率為利用節線之費洛蒙濃度與長度進行計算，近期諸多研究為針對節線距離進行改善，如 Doerner(2001)與陳家和(民 92)等。

3. 費洛蒙區域更新(Local Update)

當每隻螞蟻完成路線構建之後即進行該路線包含之節線上的費洛蒙更新，區域更新為針對迭代中每隻螞蟻隻路線，其更新量亦與螞蟻路線表現與選擇結果無關。

4. 費洛蒙全域更新(Global Update)

當迭代中所有螞蟻皆完成路線構建與區域更新後，即利用全域更新加強較佳路線中之費洛蒙濃度。全域更新之對象可能為目前之最佳路線(邱裕鈞，民 92)或較佳之數條路線(陳家和，民 92)。

5. 結束法則

蟻群演算法之停止條件一般設定為達到設定之迭代數。

2.3.2 蟻群演算法相關研究應用

目前尚未發現探討螞蟻系統相關演算法於撥召服務問題上之研究，應用於問題形式相近之檢取配送問題之文獻亦少。然而蟻群演算法已廣被應用於各類組合最佳化問題並具有多種改良形式，在進行演算法設計之前仍須進行回顧。文獻中主要改良部分為啟發式運算值、菁英策略與費洛蒙更新策略等方面，以下針對此三方面做一探討。

1. 啟發式運算值

一般車輛繞徑相關問題多使用節線長度作為啟發式運算值，然而在多篇使用最小化成本模式之研究中發現，利用節省值作代替節線長度之作法求解效果較佳(Bullnheimer，1997)(Doerner，2001)(陳家和，民 92)。

2. 菁英策略

菁英螞蟻為在每次迭代完成時需再次進行費洛蒙更新之螞蟻路線，Dorigo et al.(1997)將表現最佳之螞蟻視為菁英螞蟻，僅此一條路線進行全域更新，然而此種作法在經過多次迭代之後常因該路線之費洛蒙濃度過高，導致搜尋廣度不足並陷入區域最佳解。為改善此種缺陷，諸多研究使用多隻菁英螞蟻(陳家和，民 92)，或是透過挑選機制找出表現較差但可能具有潛力之路線進行更新(林典翰，民 95)。

3. 費洛蒙更新策略

蟻群演算法經過多次迭代之後，表現較佳之數筆路線將會因為節線上之費洛蒙濃度過高，且其他節線之費洛蒙因逐代揮發而逐漸稀少，使得搜尋落入區域最佳解。為避免此種狀

況發生，許多研究使用「費洛蒙重置」之方式進行改善(陳家和，民 92)(林典翰，民 95)，再經過指定之迭代數目或最佳解發生連續未更新之狀況時，將路網之費洛蒙濃度調回初始狀態再進行搜尋。

2.4 小結

綜觀過去相關之研究發現，由於撥召服務問題已被證實為 HP-Hard 問題，而其問題具有起迄點成對需求之特性，且必須符合一位乘客由一台車輛服務以及先接載後送達等定義限制，因此在求解上多使用插入式之作法進行尋優搜尋。然而插入式做法在尋優過程耗時甚久，因此在動態問題之討論重點多著重在一合理時間內求得較佳之可行解，或設計出一套得以於該時間內處理大型問題之作法。由於本研究之目的在最適求解演算法之設計，必須針對所求得之解進行評估，因此在設計上仍以靜態問題為處理對象。在求解演算法上，撥召公車仍以傳統的啟發式解法為主，本研究綜觀過去文獻並未發現任何研究將蟻群演算法應用於最佳化撥召服務問題，而基因演算法僅有 Bergvinsdottir 進行設計與探討，因此本研究將嘗試利用此二種演算法求解靜態多車輛撥召服務問題，並針對其最佳化成效做一探討。



第三章 模式構建

3.1 撥召服務問題描述

撥召服務問題為求解車輛接送數名乘客前往起迄點間之路線排程問題，通常問題包含一固定場站，所有車輛須於於此場站出發，完成所有乘客之接送之後亦回到此場站。在問題形式上，撥召公車常被設計為以下兩種問題(Cordeau, 2003)：

1. 在服務所有乘客與滿足所有目標式下之最小操作成本問題。
2. 在滿足車隊服務容量與其他限制式下最大化服務數量之問題。

近期研究多以第一種問題形式配合乘客時窗以構成最小化操作成本並最大化服務水準或滿足相關限制式之問題形式，影響操作成本之因素為車隊大小、車輛種類以及是否允許從其他運輸系統調派車輛以滿足需求等；對於服務水準之評估則多以最小化乘客不舒適度進行，其影響因素為停等時間、乘車時間與時窗限制等。另外，對於系統績效而言，模式中常以繞徑之距離與時間長度進行反應，並將其最小化以求得較好之績效。

在求解流程上，撥召服務問題可分為三項重要求解步驟(Cordeau, 2003)：

1. 定義各路線所需服務之乘客。
2. 將被分配於各路線中之乘客進行初步編排。
3. 根據乘客之起迄點資訊進行路線內最佳化排程。

在求解流程之設計上並非一定要區分出此三大步驟，文獻中較常被使用之作法為二階段式，或為乘客依路線分群並簡單排至路線中兩步驟為同一階段(Toth, 1997)(Cordeau, 2003)(Attanasio, 2004)，或將乘客分群完後直接依起迄點資訊進行最佳化路線排程(Bergvinsdottir, 2004)。最小操作成本模式常配合配合時窗、車輛容量、總繞徑長度、總繞徑時間或總操作成本、定義限制與其他限制式進行求解，其內容根據求解過程是否允許鬆弛解而有所不同。求解搜尋若僅允许可行解，則最小操作成本模式通常為最小總行駛距離；若允許鬆弛解之存在，則會在模式中加入懲罰成本以反應違反限制式之部分，以確保其最後產出仍是可操作之可行解。本研究所探討之撥召服務問題架構為：

輸入：路網資料、需求點與時窗、車隊規模與車輛容量。

目標：最小化總路線長度。

限制：車輛服務容量限制、車輛運行長度限制、顧客時窗限制、顧客乘車時間與停等時間限制。

求解：以巨集啟發式解法進行一階段與兩階段求解。

輸出：繞徑排程、總繞徑時間、總乘載時間與總停等時間。

3.2 基本假設

以大眾運輸系統營運者之角度來看，理想中之撥召服務車輛運行路線必須具有操作成本最小化之特性，於最短之繞行距離或時間內服務最多旅客。對於使用撥召服務之乘客而言，雖然已有自己可能無法以直接路線前往目的地，但仍希望能在所預定之時限內到達，並對於車輛在運行時之狀況如停等與繞行上具有一定限度之忍耐力。靜態撥召服務問題多著重於解品質之探討，其問題假設、模式構建與求解流程等部分皆針對此議題進行設計，與實務操作上較大的不同在於車輛停等狀況之發生與否，因為乘客通常無法忍受任何因素所造成之停等，甚至連車輛駕駛者亦無法接受，故依照演算法所求出之路線可能為最佳，但實務運作上仍以直接路線所形成之花型路線(Flower Shape)做為主要之排程方式，如圖 3-1 所示。

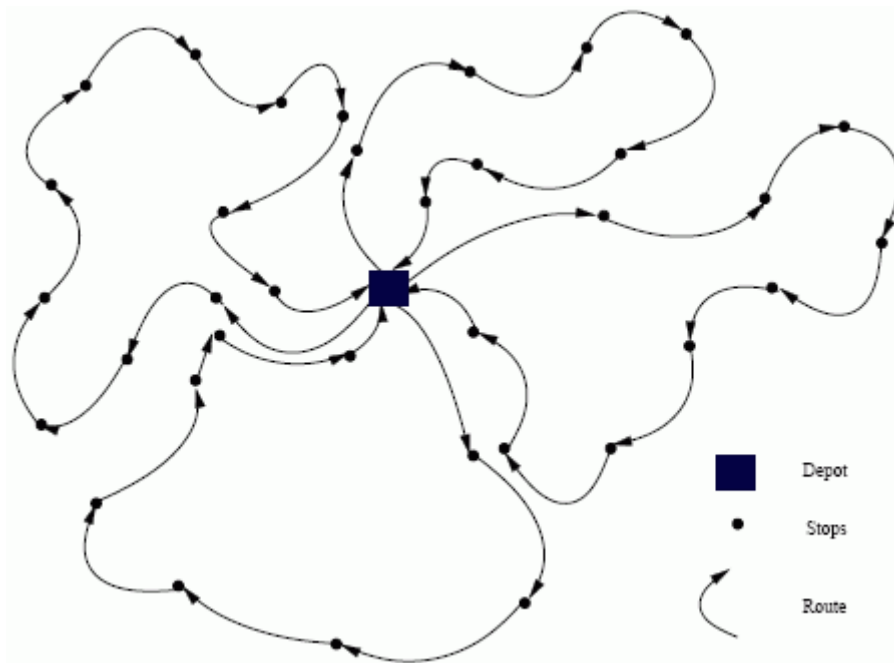


圖 3-1 花型路線示意圖

資料來源：Bergvinsdottir，2004。

由於本研究著重在於不同演算法於靜態多車輛問題求解能力之探討，因此仍需針對部分實務操作上所不允許之狀況進行假設，所有前提與假設如下：

1. 路網結構

本研究之路網根據乘客起迄點位置資訊形成，乘客之起迄點散佈於 $[-15,15] \times [-15,15]$ 之平面上，路網中任何距離之單位為分鐘。各節點間之距離計算採用歐氏距離，並假設各節點間完全互連，且各節線並無方向性。

2. 需求資料

需求分為起點需求與迄點需求，一名乘客同時具有此兩種需求，故需求總數為顧客數之

兩倍。每筆需求內容包括所在位置之二維座標、時窗範圍以及需求數，座標值之範圍為 $[-15,15]$ ，起迄點之時間間隔必須大於該乘客直接路線之長度，否則即為不合理之需求。

3. 乘客配送

乘客必須於滿足各限制之條件下，由起點運送至迄點。在路線安排上每名乘客之兩需求點順序必須符合「先接載後送達」之定義。每位乘客必須恰有一車服務，不允許取消預約等未服務之狀況。

4. 需求狀態

靜態問題中路網之需求數量與內容為固定且已知，需求量依據撥召服務問題之設定為1，而所有需求點必須恰有一台車輛通過並服務，已經服務之節點不再進行服務。

5. 繞徑狀態

為符合各需求點之時窗限制，本研究允許車輛於各種載客狀態下進行停等。停等之發生時機為車輛到達該節點之後，並結束於車輛開始進行該節點之服務時。當停等發生時，其長度將被加至車輛上所有乘客之停等時間內。

6. 車隊規模與車輛容量

場站所擁有之車輛數量為已知且形式相同，車輛以接載普通客人為主要目的，並無裝載特殊設施，故所有車輛之容量相同。研究中並不考慮外租車輛，因此所有需求只能由場站之車隊進行服務。

3.3 模式構建

依前提與假設之考量，並根據過去相關研究之數學模式進行修改及補充，本研究所採用之撥召服務問題數學模式將於此節作一介紹。在撥召服務問題相關研究中，在路線排程優劣之評估上多採用總路線行駛時間作為基準，並以最小化其值式作為目標式，故本研究亦以此作為目標式。

在演算法迭代搜尋時，本研究採用一具懲罰成本之模式反應各解之優劣，此模式根據文獻結合業者操作成本與乘客最小不舒適度兩方面，其中業者操作成本部分採用總路線長度作為基本之成本項，並配合容量限制與各車路線長度限制兩違反值；乘客乘客最小不舒適度將對服務水準之造成負面影響，此為撥召服務問題與一般檢取配送問題最大之差別所在，文獻中曾用使用過之不舒適度項目為候車時間、停等時間、時窗違反、超時乘車與路線偏移等，本研究使用停等時間、時窗違反與超時乘車作為反應乘客不舒適度之指標，並以限制式進行限制與違反值之計算，並以懲罰成本於模式中反應。

3.3.1 符號介紹

1. 圖形

$$G = (V, A)$$

2. 標號變數

i, j : 節點標號

n : 乘客標號

k : 車輛標號

3. 集合

$\{0\}$: 場站

N : 乘客集合, $N = \{1, 2, \dots, n\}$

P : 起點需求集合, $P = \{1, 2, \dots, n\}$

D : 迄點需求集合, $D = \{n+1, \dots, 2n\}$

V : 節點集合, $V = P \cup D$

K : 車輛集合

A : 節線集合, $A = \{(v_i, v_j) : i < j\}$

4. 決策變數

$$x_{ij}^k = \begin{cases} 1 & \text{車輛 } k \text{ 經由節點 } i \text{ 至節點 } j \\ 0 & \text{其他} \end{cases}$$

5. 變數

L_i^k : 車輛 k 到達節點 i 之乘載人數

T_s^k : 車輛 k 路線起始時間

T_e^k : 車輛 k 路線結束時間

AT_i : 車輛到達節點 i 之時間

ST_i : 車輛開始服務節點 i 之時間

WT_i : 車輛於節點 i 之停等時間, 即 $AT_i - ST_i$

6. 參數

m : 車輛數

Q : 車輛容量限制

t_{ij} : 節點 i 至節點 j 之旅行時間

q_i : 節點 i 之需求人數

d_k : 車輛 k 之最大繞行時間

r_n : 乘客 n 之最大乘車時間

s_n : 乘客 n 之服務時間

e_i : 需求節點 i 之時窗下限

l_i : 需求節點 i 之時窗上限

e_0 : 系統起始時間

l_0 : 系統結束時間

7. 違反值

vd_k : 車輛 k 路線長度違反值

vq_i^k : 車輛 k 於節點 i 時車輛容量違反值

vw_i : 節點 i 時窗之停等時間

vtw_i : 節點 i 時窗違反值

vr_n : 乘客 n 最大乘車時間違反值

8. 懲罰成本

w_1 : 車輛路線長度違反值之懲罰成本

w_2 : 車輛容量違反值之懲罰成本

w_3 : 停等時間之懲罰成本

w_4 : 需求點時窗違反值之懲罰成本

w_5 : 乘客最大乘車時間違反值之懲罰成本

3.3.2 數學模式

撥召服務問題文獻中多使用總路線長度最短作為求解目標式(Toth, 1997)(Cordeau, 2003), 並根據此目標式進行演算法結果之比較。Cordeau 與 Laporte(2003、2004)由於使用鬆弛解作為起始解, 為求其求解結果具可行性, 於迭代中另使用亦具懲罰成本之模式進行搜尋, 透過各違反值項對應之自我調整參數於迭代中調整懲罰成本大小, 藉此逐漸改善解之可行性, 而在各解之優劣比較上仍使用最小總路線長度作為目標式。Bergvinsdottir(2004)之基因演算法亦為採用鬆弛解作為起始解之作法, 透過具違反值之最小化目標式調整可行性, 並利用該目標值進行解之分析, 然而在與文獻結果比較時, Bergvinsdottir 亦使用最小總路線長度作為比較基準。由於本研究亦採用鬆弛作法, 故搜尋時應使用一具懲罰成本之最小化目標式以期可行性能於迭代中逐漸改善, 而解之優劣比較時則採用基本之總路線長度進行比較。

一、最小成本目標式

一般車輛繞徑問題多使用總行使長度相關成本項作為評估業者成本之最小化目標式，所使用之單位為繞徑所需之時間或金錢等，即以最小化總 $x_{ij}^k t_{ij}$ 做為目標式，本研究亦使用最小總繞徑時間作為業者之成本目標式，然而在允許車輛進行停等之撥召服務問題中，車輛即使進入停等狀態亦會消耗其最大運行時間，使用 $x_{ij}^k t_{ij}$ 做為評估並無法反應停等所佔用之運行時間，故本研究於業者最小成本目標式中採用各車輛總行駛時間之總和反應，其計算方式為各車輛回站時間 T_e^k 與車輛離站時間 T_s^k 差之總和。此目標式將用於不同演算法與模式參數設定之求解結果比較分析以及文獻結果之比較上。本研究使用之最小成本目標式為：

$$\min \sum_{k \in K} (T_s^k - T_e^k) \quad (3.1)$$

二、迭代搜尋目標式

此目標採用具懲罰成本之模式型態，將所有對系統績效產生負面影響之項目與違反值反應於目標值中，藉以評估迭代中各解之優劣。在使用巨集啟發式解法之文獻中，Cordeau et al.(2003、2004)使用總繞徑時間以及車輛乘載量、繞徑時間、時窗限制與乘客乘車時間限制四項違反值，配合違反值之自我調整非負參數(Self-Adjusting Positive Parameters)以構成最小化目標式，並根據此式進行搜尋；Bergvinsdottir(2004)則使用總運送時間、乘客乘車時間違反值、乘客所承受之總停等時間、時窗限制違反值、路線時間長度違反值等項配合固定權重構建最小化目標式，並作為尋優與比較之依據。本研究則根據文獻中所使用之項目以及撥召服務問題之特性構建最小成本目標式，其中屬於業者操作成本之部分為繞徑長度 $x_{ij}^k t_{ij}$ 、車輛路線長度違反值 vd_k 與車輛容量為反值 vq_i^k ；乘客不舒適度方面則以停等時間 vw_i 、時窗限制違反值 vtw_i 與乘車時間違反值 vr_n 表示。搜尋所使用之最小化目標式由此六項構成，違反值部份則配合懲罰成本反映於目標值中，並藉由 Cordeau 所提出之自我調整非負參數隨迭帶調整其值大小。整體最小成本目標式為：

$$\min \sum_{k \in K} \sum_{i,j \in V} x_{ij}^k t_{ij} + w_1 \sum_{k \in K} vd_k + w_2 \sum_{k \in K} \sum_{i \in V} vq_i^k + w_3 \sum_{n \in N} vw_n + w_4 \sum_{i \in V} vtw_i + w_5 \sum_{n \in N} vr_n \quad (3.2)$$

各項目依序為代表業者操作成本之總繞徑時間、總車輛路線長度違反值與車輛容量違反值，其後為反應乘客不舒適度之停等時間、時窗限制與乘車時間之違反值項，其中各違反值之計算方式為：

1. 車輛路線行駛時間為回站時間 T_e^k 與出站時間 T_s^k 之差值，若此時間值大於車輛最大繞行時間 d_k 則為不可行之解，路線長度與最大繞行時間之差值即為違反值。

$$vd_k = \begin{cases} T_e^k - T_s^k - d_k & T_e^k - T_s^k - d_k > 0 \\ 0 & \text{其他} \end{cases} \quad (3.3)$$

2. 車輛容量違反值為逐車逐點檢查該到達點之乘客數目 L_i^k ，若大於車輛容量 Q 則違反。

$$vq_i^k = \begin{cases} L_i^k - Q & L_i^k - Q > 0 \\ 0 & \text{其他} \end{cases} \quad (3.4)$$

3. 停等時間發生於車輛到達之後 AT_i 至車輛開始服務時 ST_i ，此二時間之差值即為該節點之停等時間。

$$vw_i = \sum_{i \in V} (ST_i - AT_i) \quad \sum_{i \in V} (ST_i - AT_i) > 0 \quad (3.5)$$

4. 時窗限制之檢查為逐點檢查，由於車輛到達節點之時間 AT_i 若在該點時窗下限之前 e_i 則發生停等，因此僅需檢查到達時間是否超過時窗下限 l_i 。

$$vtw_i = \begin{cases} AT_i - l_i & AT_i - l_i > 0 \\ 0 & \text{其他} \end{cases} \quad (3.6)$$

5. 乘客於下車之後即檢查其總乘車時間，即乘客於車上之時間，正在服務之狀態並不列入計算，因此必須扣除上車點之服務時間 s_n 。

$$vr_n = \begin{cases} ST_{n+|N|} - ST_n - s_n - r_n & ST_{n+|N|} - ST_n - s_n - r_n > 0 \\ 0 & \text{其他} \end{cases} \quad (3.7)$$

相關限制式為：

1. 流量守恆限制

$$\sum_{k \in K} \sum_{j \in V} x_{ij}^k = 1, \quad \forall i \in V \quad (3.8)$$

$$\sum_{k \in K} \sum_{i \in V} x_{ij}^k = 1, \quad \forall j \in V \quad (3.9)$$

$$\sum_{k \in K} \sum_{i \in V} \sum_{j \in V} x_{ij}^k = m + N \quad (3.10)$$

$$\sum_{i \in V} x_{ih}^k - \sum_{j \in V} x_{jh}^k = 0, \quad \forall k \in K, \quad \forall h \in V \quad (3.11)$$

(3.8)式代表只允許一台車從節點 i 離開，(3.9)式代表只允許一台車到達從節點 j 。(3.10)式限制排程所通過之節線數量以避免子迴路發生。(3.11)式則表示若車輛 k 到達節點 h 仍會由節點 h 離開。

2. 車輛容量限制

$$x_{ij}^k (L_i^k + q_j - L_j^k) = 0, \quad \forall k \in K, \quad \forall i, j \in V \quad (3.12)$$

$$0 \leq L_i^k \leq Q, \quad \forall k \in K, \quad \forall i \in V \quad (3.13)$$

$$L_0^k = 0, \forall k \in K \quad (3.14)$$

在車輛容量相關限制式裡，(3.12)式表示車輛 k 於在服務完節點 j 之後其乘載量將從 L_i^k 變成 L_j^k ，(3.13)式則限制車輛於任何時候之乘載量必須屬於合理範圍。(3.14)則表示最後當車輛於開始服務離開場站或完成服務回到場站時皆必須處於空車狀態。

3. 時間限制

$$x_{ij}^k (ST_i + s_i + t_{ij} - ST_j) \leq 0, \forall k \in K, \forall i, j \in V \quad (3.15)$$

$$ST_i + s_i + t_{i,i+n} - ST_{i+n} \leq 0, \forall k \in K, \forall i \in P \quad (3.16)$$

$$T_s^k \geq e_0, \forall k \in K \quad (3.17)$$

$$T_e^k \leq l_0, \forall k \in K \quad (3.18)$$

(3.15)式表示節點 j 開始服務之時間必須於上一節點 i 開始服務時間之後，並且須相差服務時間與旅行時間等合理時間間隔；(3.16)式則表示任何乘客之迄點開始服務時間必須晚於期起點開始服務之時間。(3.17)式與(3.18)式之目的在限制各車之出站與回站時間必須位於系統營運時間內。

4. 定義限制

$$x_{ij}^k \in \{0,1\}, \forall k \in K, \forall i, j \in V \quad (3.19)$$

(3.19)為二元整數限制。

3.4 小結

本章除了對研究主題作一較明確之定義外，同時構建符合本研究主題之數學模式。其中以最小化總路線長度作為目標式，並以一包含之業者成本項與乘客不舒適度之具懲罰成本模式作為搜尋依據，以期在符合各項限制下幫業者追尋一最小之營運成本與乘客不舒適度之撥召服務排程。

第四章 求解策略

撥召服務問題已被證實為 HP-Hard 問題，當需求量增多在合理時間內求得最佳解將更形困難，因此在求解演算法上多採用啟發式解法進行，尤以節點再插入式之法為多。巨集啟發式解法應用於各類型車輛繞徑問題已經過多年之研究，然而在撥召服務問題等檢取配送問題類型之研究甚少。由於此種問題形式之需求具成對與順序關係，在迭代中無論各解的內容如何變動，這些限制仍需遵守以確保基本之可行性。傳統之處理方式為嚴格遵守所有限制，在迭代中僅針對可行解進行搜尋與比較(Toth, 1997)(Diana, 2004)，而近期多採用懲罰成本模式之研究，雖然演算法起始階段將部分限制式鬆弛，但「一人一車」與「先接載後送達」之定義限制仍必須保留，為此多採用乘客分群與路線排程二階段作法(Cordeau, 2003)(Attanasio, 2004)(Bergvinsdottir, 2004)，在分群階段將可避開這些限制並於排程時再行考慮。本研究在求解流程上保留二階段之形式，並依照所採用之演算法設計完整一階段求解之作法，並利用 Visual Basic 撰寫程式作為求解工具，將各演算法在此兩類作法之表現進行比較，以找出求解撥召服務問題最適之方式與演算法。

4.1 繞徑策略

由於撥召服務問題與具時窗之檢取配送問題皆具有需求節點時窗限制，為避免違反此項限制，在進行繞徑排程時必須針對時窗選擇繞徑策略，以避免違反限制而發生繞徑失敗之狀況。撥召服務問題與一般車輛繞徑問題較為不同之一點為其允許車輛於繞徑時進行停等 (Idle)，當車輛處於此種狀態時將產生停等時間 (Waiting Time)。對於業者而言，處於停等狀態之車輛由於暫停所有服務且其位置並不會改變，將會減少該車輛之可用時間並增加成本；對於乘客而言，停等亦會徒增其乘車時間，然而此時期對其前往迄點並無任何幫助。由於停等是一為符合節點時窗限制所使用之不得已之手段，對於乘客與業者雙方皆無益處，故有必要針對最小化停等時間之繞徑作法進行一探討。

撥召服務問題中車輛於一節點所發生之時間參數可分為到達時間、停等時間、開始服務時間與離開時間等四項時間值，本研究將停等時間之發生時機設於車輛到達節點之後與發生服務之前，如圖 4-1 所示：

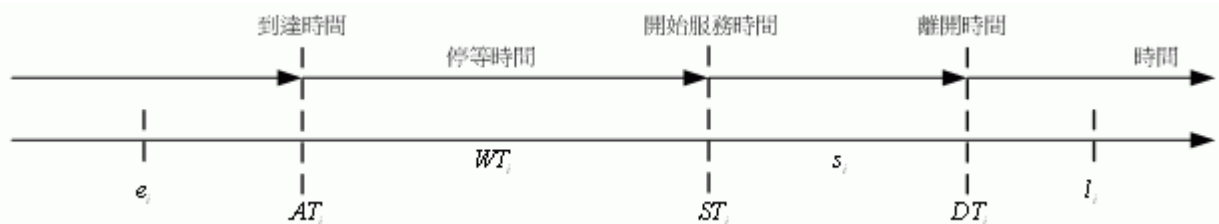


圖 4-1 車輛於各節點之到達時間、停等時間、開始服務時間與離開時間示意圖

由於停等時間之意義為為符合時窗限制所發生之暫停，若車輛於該節點之服務完成後再

發生停等，雖對於迄點需求之乘客較無損失，然而對於起點需求之乘客則會增加其乘車時間，並減少其可用於運送之時間。將停等時間發生於開始服務之前雖亦會佔用乘客之乘車時間，然而對於上車之乘客而言，停等發生於其上車之前，對其乘車時間並無影響，對於在該節點下車之乘客而言，此停等是為符合其時窗限制所作之調整，因此此種作法對於乘客乘車時間之影響較小，故將其作為繞徑停等之策略。

根據本研究所擬定之停等發生策略，車輛於節點之到達以及離開方式分為下列三種：

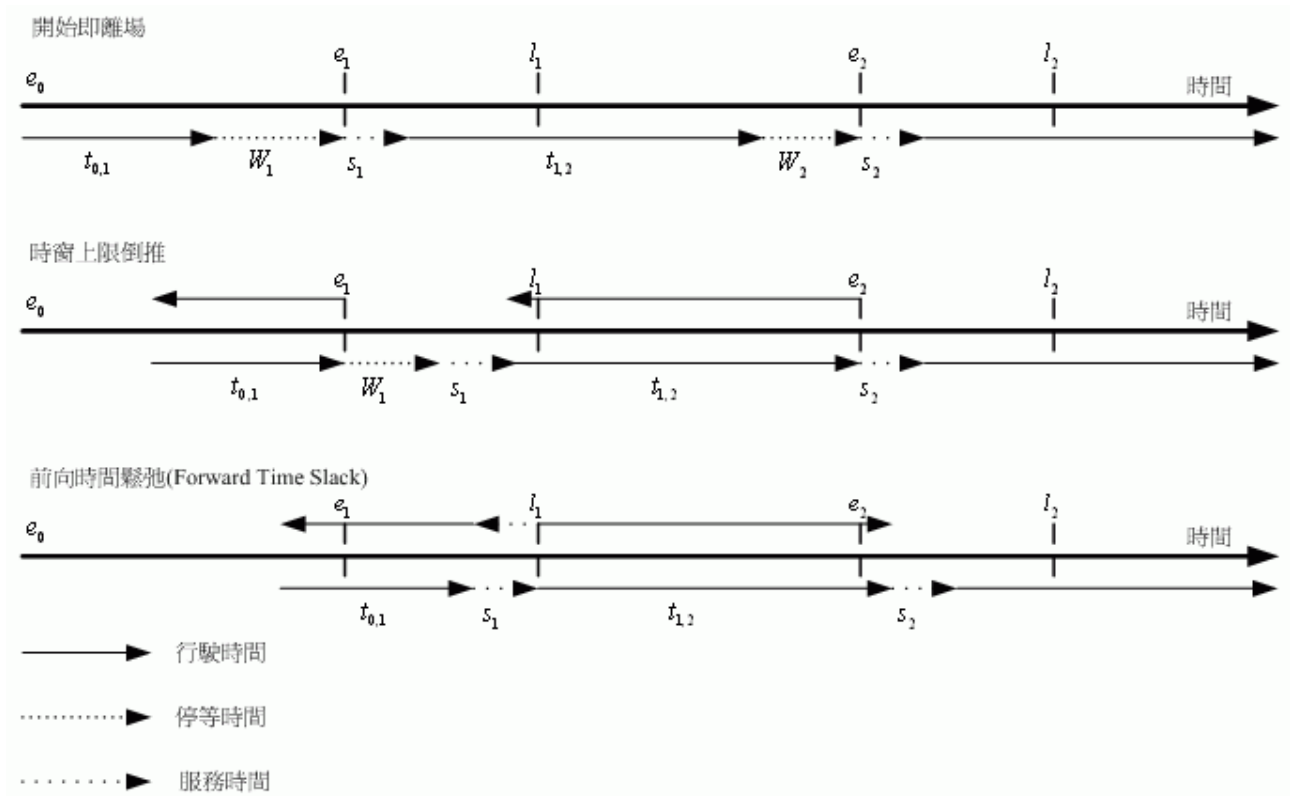


圖 4-2 不同到達與離開策略比較圖

1. 開始即離場

所有車輛於系統開始時間即離開場站前往第一個到達節點，且於各節點完成服務後立刻離開該節點前往下一節點。此作法為最基本之車輛運作方式，然而為符合各節點之時窗將會發生大量之停等，並增加車輛繞行時間與乘客乘車時間。

2. 時窗上限倒推

此作法根據路線中各節點之時窗上限與車輛前進至該節點所需旅行時間以倒推之方式決定車輛於各節點之離開時間，若節點之到達時間、服務時間與前進至下一點之旅行時間總和小於下一節點之時窗上限時則將發生停等。此種作法雖能有效減少停等時間，但仍會因為各節點時窗之關係而發生停等，而且此策略並無法有效控制停等發生之時機，故乘客仍會因為停等而徒增乘車時間。

3. 前向時間鬆弛(Forward Time Slack)

此為 Savelsbergh 於 1992 所提出一應用於具時窗車輛繞徑問題之修正方式，作法即為根

據時窗以及節線之繞行時間計算各節點之時間鬆弛量，並將各節點開始服務時間往後移此鬆弛量之長度，接著再度修正其後節點之各項時間，即可達到路線長度與總停等時間極小化之目標。節點*i*之前向時間鬆弛 F_i 計算方式如：

$$F_i = \min_{i \leq j \leq q} \left\{ l_i - \left(ST_i + \sum_{i \leq p \leq j} (t_{p,p+1} + s_p) \right) \right\} \quad (4.1)$$

$$= \min_{i \leq j \leq q} \left\{ \sum_{i \leq p \leq j} WT_p + (l_j - ST_j) \right\} \quad (4.2)$$

q ：路線最末節點

l_i ：節點*i*時窗上限

s_i ：節點*i*之服務時間

$t_{i,j}$ ：節點*i*至節點*j*之旅行時間

ST_i ：節點*i*開始服務之時間

WT_i ：發生於節點*i*之停等時間， $WT_i = ST_i - AT_i$

AT_i ：車輛到達節點*i*之時間

根據此二式可得知各節點之鬆弛量為累積至該節點之停等時間，加上開始服務時間與時窗上線之差值。若車輛於繞徑途中發生空車，由於在下一位乘客上車之前所發生之停等皆不影響乘客之不舒適度，故可重新計算該節點之鬆弛量以最佳化下一個乘客起點之最佳時間值。車輛離開場站之時間亦可根據此方式進行延遲時間長度計算，除可改善路線長度之外，亦可減少車輛於各節點所發生之停等，進一步減少乘客之乘車時間。故利用前向時間鬆弛改善路線之步驟為：

步驟一：令 $DT_0 = e_0$

步驟二：計算路線中各節點之到達(AT_i)、停等(WT_i)、開始服務(ST_i)與離開時間(DT_i)。

步驟三：計算 F_0

步驟四：延後 DT_0 ， $DT_0 = e_0 + \min \left\{ F_0, \sum_{0 < p < q} WT_p \right\}$

步驟五：更新各節點之 AT_i 、 WT_i 、 ST_i 與 DT_i 。

步驟六：計算所有乘客之乘車時間。

步驟七：若車輛到達節點*j*之前處於空車狀態則進行步驟八，否則進行步驟十三。

步驟八：計算 F_j

步驟九：延後 ST_j ， $ST_j = ST_j + \min\left\{F_j, \sum_{j < p < q} W_p\right\}$ ，並計算 $DT_j = ST_j + s_j$ 。

步驟十：更新節點 j 其後所有節點之 AT_i 、 WT_i 、 ST_i 與 DT_i 。

步驟十一：重新計算節點 j 之後所有乘客之乘車時間。

步驟十二：回步驟七重新搜尋節點 j 。

步驟十三：計算所有違反值與該路線之目標值。

綜合以上繞徑策略，本研究於排程構建時先採用時窗上限之作法進行初始之路線時間安排，完成後再利用前向時間鬆弛改善路線中之時間值，以其整體之停等與繞徑長度能獲得最佳之結果。

4.2 求解流程

撥召問題相關文獻中多採用二階段之求解流程，此類流程作法為先將乘客加以分群並略做路線編排，再以路線內與路線間改善之方式完成整體之排程規劃。本研究為探討求解流程之設計配合不同具及啟發式演算法對求解效益之影響，除了分群繞徑二階段流程之外，亦設計在完整一階段作法以進行比較。一階段求解部份直接對問題中之各車路線同時進行最佳化改善搜尋；二階段做法則於第一階段進行顧客分群，再透過第二階段最佳化各車路線排程，完成之後再由第二階段所產生之整體繞徑計畫績效計算各分群之表現，並進行分群演算法迭代之更新。

4.2.1 完整一階段流程

完整一階段在求解時直接對完整之繞徑計畫進行最佳化搜尋，搜尋時僅遵守「一人一車」與「先接載後送達」此二定義限制，業者操作成本與使用者不舒適度相關限制式將予以鬆弛，所產生之違反值部份經由懲罰成本反應至最小成本模式中，並以此成本值做為各解比較更新之依據。完整一階段流程如圖 4-3 所示：

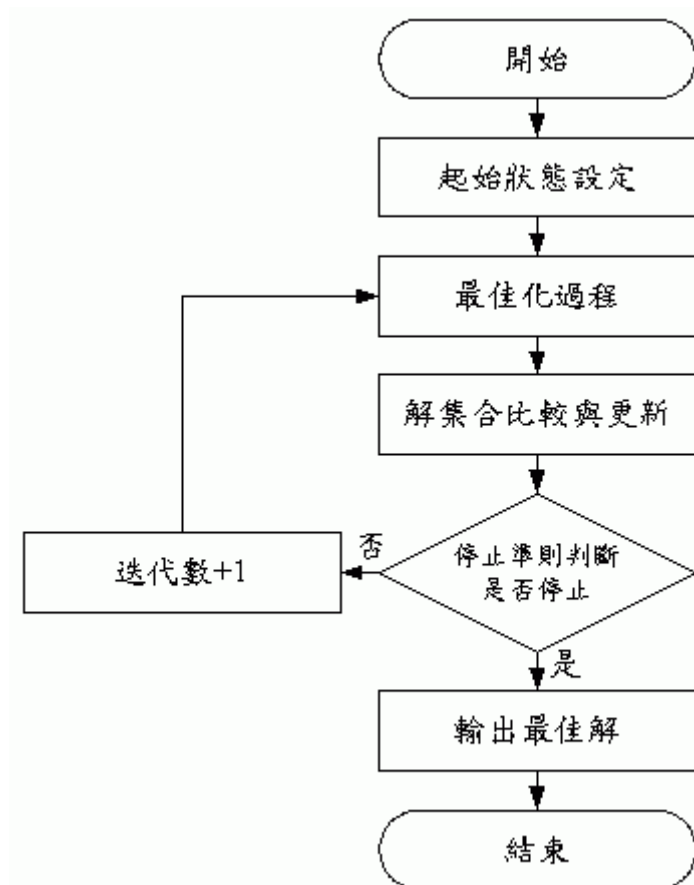


圖 4-3 完整一階段求解流程圖

各步驟說明如下：

1. 起始狀態設定

此步驟根據不同演算法設定起始狀態，包括所需使用之參變數以及解集合形式之設定。

2. 最佳化過程

此步驟根據各演算法進行臨近解之搜尋，在產生完各演算法所設定之數量之解後，即進入下一步驟進行比較。

3. 解集合比較與更新

解之比較與更新為利用(3.2)最小成本式進行，根據此式與該迭代之自我調整參數值計算各解之目標值，並依照各演算法之更新規則進行較佳解之比較更新。

4. 停止判斷

演算法迭代停止與否為根據停止準則進行，本研究所採用之停止準則為最大迭代數限制，若迭代數已達到則停止搜尋並輸出目前最佳解做為此次搜尋之結果。

4.2.2 分群繞徑二階段流程

分群與繞徑二階段求解流程在文獻中較為常見，因分群時限制較小，而分群後又將形成數量等同於路線數或車輛數之單車輛問題，在求解設計上較為便利。本研究在此二階段接使用巨集啟發式解法進行最佳化求解，當分群階段透過最佳化過程產生新分群後，將所有分群移至路線排程階段進行各路線之排程。每筆路線之分群於路線排程階段構建完整之繞徑路線並計算其目標值，完成所有路線之計算之後將目標值回饋於其所屬分群以作為各分群之績效表現。各分群完成績效比較後除須進行分群解集合之更新外，亦需紀錄集合中各解之路線，其中最佳者將於完成所有迭代之後輸出。分群繞徑二階段流程如圖 4-4 所示：

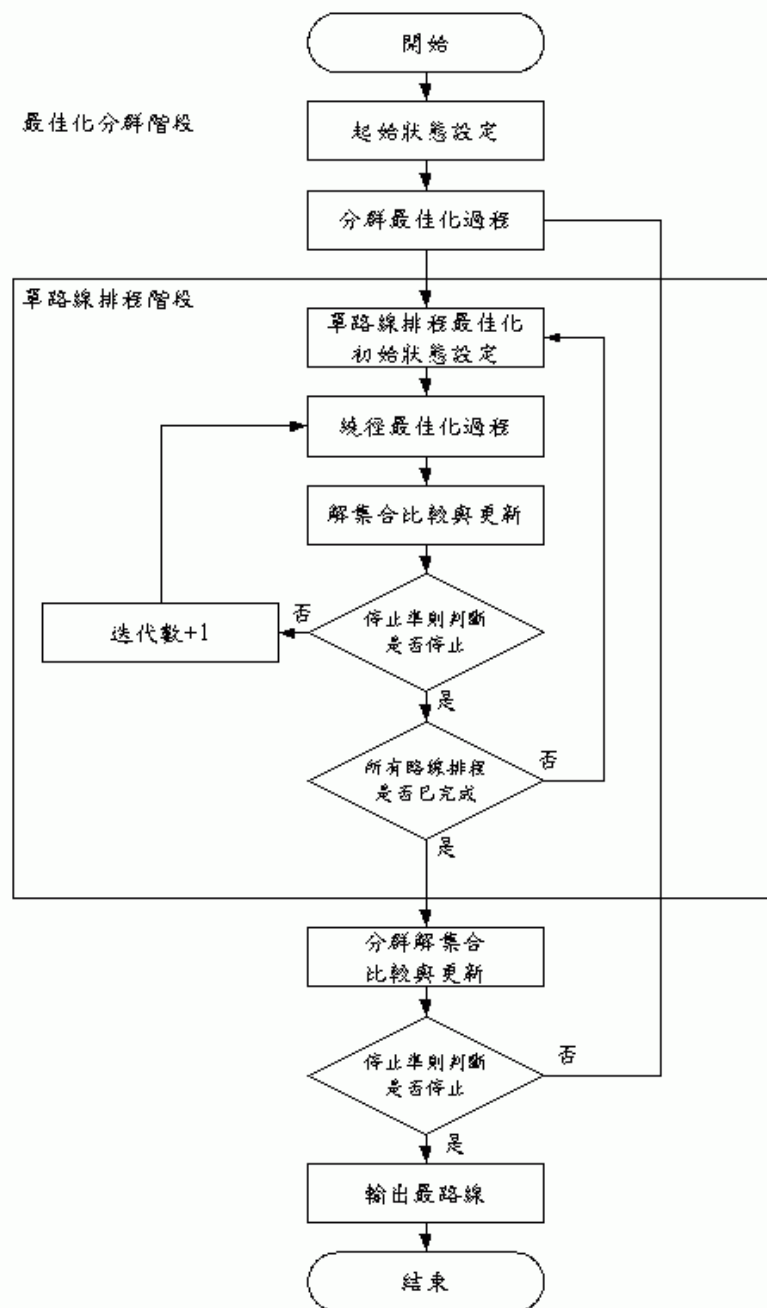


圖 4-4 分群繞徑二階段求解流程圖

二階段演算流程各步驟說明如下：

1. 分群階段：起始狀態設定

此步驟根據不同演算法設定起始狀態，包括所需使用之參變數以及解集合形式之設定。

2. 分群階段：最佳化過程

此步驟根據各演算法進行臨近解之搜尋以產生新的顧客分群，並將新的分群交由繞徑階段以產生符合分群內容之最適排程計畫，並藉此評估分群之優劣。

3. 繞徑階段：起始狀態設定

繞徑階段根據分群階段所產生之分群內容進行各分群(路線)之排程，此階段各路線所需服務之乘客已由分群階段劃分，各路線並不再進行乘客交換之動作，故路線安排為逐分群進行，每次僅進行一路線之編排。

4. 繞徑階段：最佳化過程

根據各乘客分群利用巨集啟發式演算法進行最適路線尋優，此部份為逐分群進行，各路線之表現並不會直接影響其他路線，所包含之乘客亦不會有置換之情況。

5. 繞徑階段：解集合比較與更新

根據式(3.2)進行各排程之優劣比較，以搜尋較佳之排程。

6. 繞徑階段：停止判斷

繞徑階段之搜尋停止判別分為兩部份，當正在進行某一分群之路線編排時，其停止準則為以最大迭代數判別，若達到最大迭代數則輸出解集合中最佳者作為該分群之路線；當一分群完成編排之後則判斷是否所有分群路線皆已完成，若無則回到繞徑階段之起始狀態設定並進行下一分群之路線編排。

7. 分群階段：解集合比較與更新

當所有分群解集合之路線皆已完成編排後則進行各分群解之比較，比較之依據則為各分群所包含之路線於(3.2)式之總合表現，選出較佳之數個解之後進行解集合之更新。由於演算法最終之輸出為整體之路線排程，故在解之更新上除需紀錄分群狀態之外，亦須記錄所對應之路線排程。

8. 分群階段：停止判斷

演算法所使用之停止準則為最大迭代數，若達到則書出最佳之排程計畫。

4.3 演算法設計

將巨集啟發式解法應用於組合啟發式解法中，必須針對問題特性、模式架構與求解流程進行細部設計。本研究採用適應性基因演算法(Adaptive Genetic Algorithm)、族群競爭基因演算法(Family Competition Genetic Algorithm)與蟻群演算法(Ant Colony Optimization)作為求解工具，主要想法為基因演算法與蟻群演算法於各類車輛繞徑問題中以廣被應用，而 Bergvinsdottir(2004)更將基因演算法應用於撥召供車問題上，為求有更佳之表現故選擇此二種較為先進之基因過程進行求解。另外，本研究基於採用具懲罰成本之最小化總成本目標式，在搜尋時允許違反限制式之鬆弛解，然而鄰近解之架構上仍需遵守同一需求之起迄點為同一台車輛服務之「一人一車」限制，以及起迄點於路線中「先接載後送達」之順序，因此必須針對此二定義限制進行演算法搜尋上之設計。

4.3.1 適應性基因演算法

基因演算法乃是 Holland 於 1975 模擬生物演化過程所推衍出來之演算法，演算流程大致可分為複製、交配與突變三個主要步驟，將編碼過後之解基因進行不斷迭代交換以搜尋更加解之求解演算法。傳統基因演算法常因為搜尋時母體變異性(Diversity)快速流失而導致陷於區域最佳解，雖然可利用突變之方式試圖跳脫，但跳脫後之解常常於後續迭代中因為適應不佳而被淘汰(Zhu, 2003)，使得突變機制形同虛設。為此諸多演算法研究在嚐試分析並保持母體之變異性以維持基因演算法於搜尋上之廣度，並提出各種改良過後具適應性之基因演算法，使用可變動之交配率與突變率以維持搜尋績效，其中以 Zhu 所提出之變異性控制(Diversity-controlling)基因演算法為較為新式之適應性作法(Zhu, 2003)。

Zhu 針對基因演算法每次迭代母體中各染色體之顯性(Phenotype)、標準差(Standard Deviation)、基因型(Genotype)與遺傳代號(Ancstral ID)四種變異性進行評估，並發現基因型的差異性於迭代中具有平穩下滑之趨勢，且當其變異性越接近零收斂速度越快，故 Zhu 提出一隨染色體基因型變異性而調整交配率與突變率之作法以保持搜尋之廣度。基因型變異值之計算方式為加總各染色體對母體中其他染色體之 Hamming Distance，計算方式如下：

$$Hamming(u, v) = \sum_i |\text{sgn}(u[i] - v[i])| \quad (4.3)$$

其中 $u[i]$ 與 $v[i]$ 為染色體 u 與染色體 v 中第 i 個基因，因此染色體基因型變異值計算方式為：

$$gtype(P) = \frac{1}{2} \sum_{i \neq j} Hamming(P[i], P[j]) \quad (4.4)$$

其中 $P[i]$ 與 $P[j]$ 為母體 P 中第 i 與 j 個染色體編碼。完成該迭代之變異值計算之後，再利

用該值求取交配與突變機率，計算方式為：

$$p' = \max \left(p_{\min}, \min \left(p_{\max}, p \left(1 + \frac{\xi(d_t - d)}{d} \right) \right) \right) \quad (4.5)$$

其中 p 為交配機率現值； p' 為新的交配機率； d 為個基因型變異值； d_t 為預設之目標變異性目標值； ξ 為預設之敏感度參數，其值越小則迭代中所產生之變異性變化越小； p_{\max} 與 p_{\min} 為機率之上下限，新機率必須界於此二值之間。Zhu 將此方法應用於多車輛具時窗之車輛繞徑問題，其結果顯示可於較短之迭代中找到近似於已知最佳解之較佳解。一階段求解之適應性基因演算法流程如下圖所示：

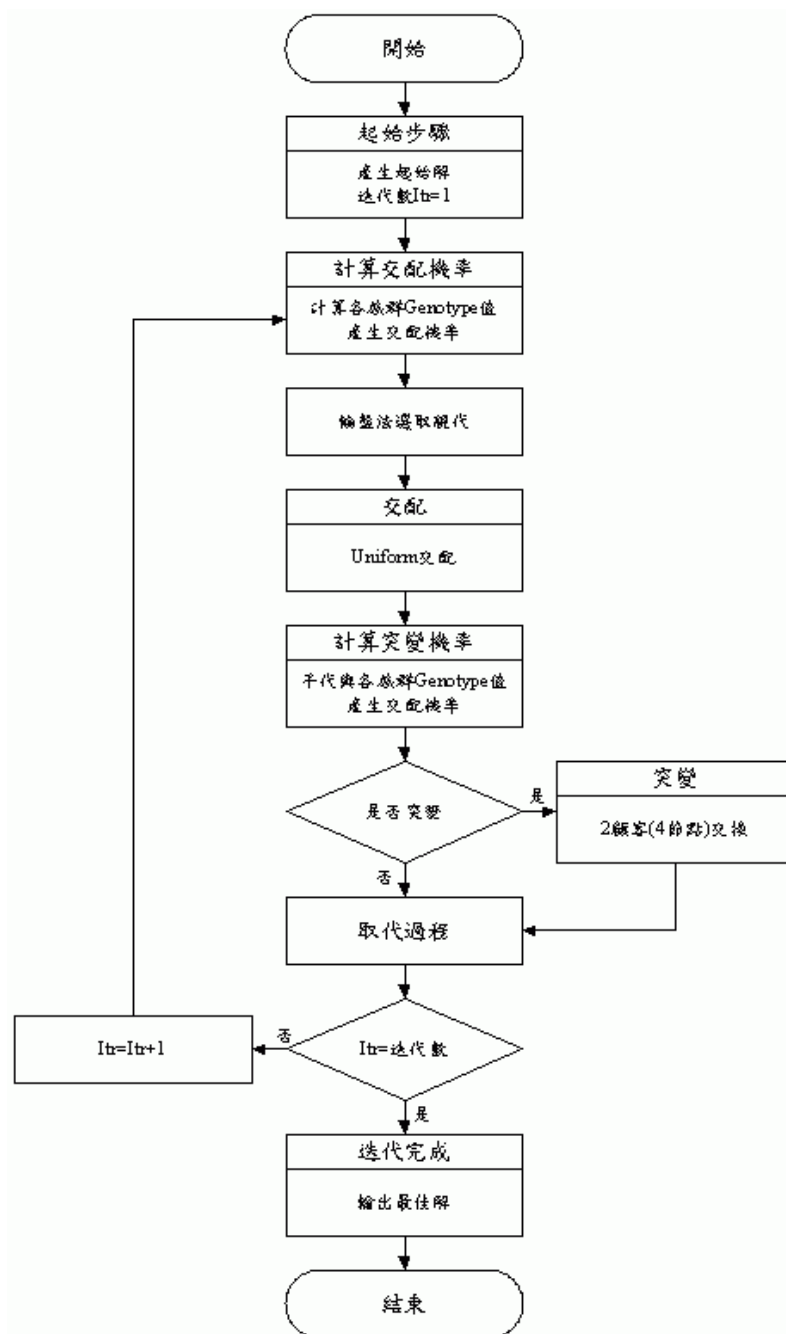


圖 4-5 適應性基因演算法求解流程圖

各步驟之說明如下：

1. 起始步驟

此步驟為演算法之初始階段，主要工作為設定各項演算法參數與起始群體之產生。本研究在搜尋過程中允許不合限制式之鬆弛解產生，其違反部分將以懲罰成本之方式反應於最小成本模式中，因此在起始解的產生上使用亂數編排之方式，主要目的是希望能藉此增加搜尋之廣度。起始解產生之方式如圖 4-6 所示：

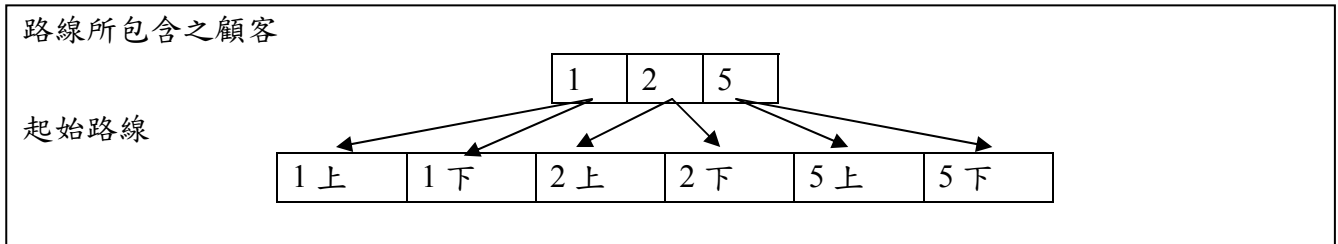


圖 4-6 基因演算法起始解產生方式

2. 計算交配機率

Zhu 所提出之適應性基因演算法並非使用固定之交配機率，改採於迭代中依據群體中染色體之組成內容以決定各染色體之交配機率，計算方式為先以(4.4)式計算群體之 gtype 值，再帶入(4.5)式分別計算各染色體之交配機率。此種機率計算方式亦使用於突變機率之計算，Zhu 即透過此種機制以控制基因迭代中群體變異程度之收斂程度，以達到維持搜尋廣度之目標。各染色體之交配機率完成計算之後採用輪盤法選取交配親代。

3. 交配

本研究採用 Uniform 運算元做為基因交配運算元，主要考量因素為此種交配方式較符合撥召服務問題之特性與基因編碼方式。由於本研究所採用之編碼方式於交配時可能會導致節點重複或起迄順序不符等狀況，雖然搜尋迭代中允許鬆弛解之產生，然為保持各解之基本合理性，仍須維持「一人一車」與「先接載後送達」此二定義限制，Uniform 運算元在此方面之維持程度較佳，故採用此種交配方式。

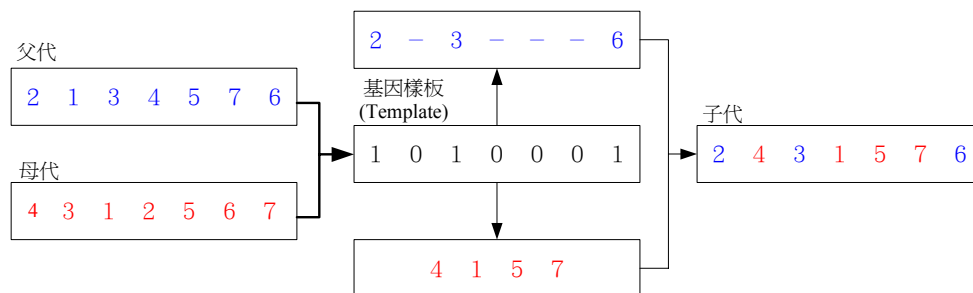


圖 4-7 Uniform 交配示意圖

Uniform 交配運算元為透過一基因樣板(Template)以決定其交配時父代所保持之基因數量與位置，此樣板為透過一預設之保留率與隨機亂數所產生，父代個體根據此樣板保留其基因，

而未保留之基因則由母代依序填補，並以此方式產生子代。適應性基因演算法之樣板保留率亦透過機率運算式(4.5)於迭代中進行改變，並由被選出之兩親代染色體互為父母帶以產生兩子代進行比較與更新。

完成交配後本研究利用數個檢驗與再插入機制以保證子代具有基本之合理性。兩個體經過交配互換後極有可能產生節點重複或未服務之情形，因此本研究設計移除、重劃與再插入三個步驟進行子代路線之檢查與重建。

- (1) 移除：移除為搜尋子代解中重複服務或服務不完全之節點，並將其服務狀態設為未服務，於再插入步驟中重新搜尋較佳之插入位置。節點編號之設計為前 n 筆需求節點為起點需求，後 n 筆需求節點為迄點需求，在總共 $2n$ 筆之需求節點資料中，第 i 筆與第 $i+n$ 筆資料即為顧客 i 之起點與迄點需求。若子代中某一節點重複出現或是起迄點於單一路線中僅出現一點，則將這些節點全數由路線中移除，並於再插入步驟中利用關鍵節點(Critical Node)之方式進行最佳插入位置搜尋。
- (2) 重劃：經過移除之個體基因編碼會出現數個空白欄位，在進行再插入之前必須將這些欄位進行清除以維持路線之連續性，避免後續工作進行上之困難。

表 4-1 移除與重劃步驟示意

父代	2 上	1 上	3 上	1 下	3 下	4 上	4 下	5 上	2 下	5 下
母代	4 上	5 上	2 上	3 上	2 下	4 下	1 上	1 下	5 下	3 下
基因樣板	1	0	0	1	0	0	0	1	1	1
子代	2 上	4 上	3 上	1 下	4 下	1 上	3 下	5 上	2 下	5 下
移除並重劃	2 上	4 上	3 上	4 下	3 下	5 上	2 下	5 下		
待插入顧客	1									

- (3) 再插入：此步驟為逐點搜尋重劃後之子代路線中的所有區間，並將待插入顧客之起迄節點插入最佳之區間。再插入搜尋相當費時，但此種方式為目前處理 DARP 或 PDVRP 此類具成對需求點之車輛繞徑問題最為合適之方式，為有效減少搜尋次數，本研究使用 Cordeau 與 Laporte(2003)所提出之「關鍵節點」作法，先進行每位顧客之關鍵節點判斷並優先搜尋該節點最佳插入位置，再根據其節點類型於相對之區間搜尋非關鍵節點最佳位置。關鍵節點之判斷方式為，若某一節點 e_i 不等於系統起始時間且 l_i 不等於系統終止時間，則此節點為該顧客之成對節點中的關鍵節點，並須優先進行再插入搜尋。Cordeau 與 Laporte 所使用之例題中許多顧客具有此種單時窗特性，為更貼近實際狀況，本研究將關鍵節點之判斷依據修改成根據時窗鬆緊程度進行判別，即比較 $l_i - e_i$ 與 $l_{i+n} - e_{i+n}$ ，時窗長度較短者定為關鍵節點。

完成路線檢查與重建之子代將進行突變運算之判別，並在完成適應度之計算後進行群體取代檢查。

4. 突變

基因突變之目的在於嘗試跳脫區域最佳解，而突變過程包含突變機率與突變運算方式。若決定發生突變，適應性基因演算法將依照子代之基因型變異值計算突變機率，計算

方式與交配機率相同，皆使用(4.4)與(4.5)式進行計算。突變運算元採用兩點互換之方式，透過亂數選出交換之兩位顧客，接著於所在路線相同位置互換起迄點。

5. 取代過程

此步驟為比較子代與親代族群，利用最小成本目標式之計算以決定子代之目標值，若子代出現表現較佳者則取代親代表現最差者。

6. 停止判斷

基因演算法根據停止準則進行迭代停止之判斷，本研究採用最大迭代數做為停止準則，若已達到則輸出族群之最佳解。

在二階段求解上，與一階段之主要差異在於兩階段之染色體編碼方式，不同流程之編碼方式如圖 4-8 所示：

一階段求解編碼

2上	1上	1下	2下	0	3上	4上	4下	5上	5下	3下	0
----	----	----	----	---	----	----	----	----	----	----	---

二階段求解分群編碼

顧客編號	1	2	3	4	5
路線編號	1	1	2	2	2

二階段路線排程編碼

2上	1上	1下	2下		
3上	4上	4下	5上	5下	3下

圖 4-8 不同求解流程基因演算法染色體編碼示意圖

各編碼方式說明如下：

1. 一階段求解流程編碼

完整一階段求解流程其群體中任一染色體即為一完整之繞徑計畫，在起始解之產生上如圖 4-6 採用按照乘客順序亂數產生其所屬路線之方法，完成所有乘客之起始位置產生之後則將所有路線合併成完整之染色體，路線間則以場站節點(即圖 4-8 中之節點 0)做為劃分，故染色體長度為需求數加上車輛數。

2. 二階段分群編碼

分群所使用之基因編碼為按乘客順序之所屬分群編號，即編碼第一之位置即為乘客 1 之所屬分群，以此類推，故其長度為乘客數。

3. 二階段繞徑編碼

路線排程所使用之編碼方式類似一階段流程所使用之作法，在編碼內容與起始解產生

方式採用相同之方法，唯其僅根據目前所處理之分群進行編排，故染色體長度根據分群包含之乘客數而有所法不同。

4.3.2 族群競爭式基因演算法

族群競爭式基因演算法為 Yang 等人於 2000 年所提出之適應性機因演算法，主要之特點為在迭代中加入族群競爭之概念以增加搜尋之廣度與多變性，避免傳統基因演算法常遇到之過早收斂問題。

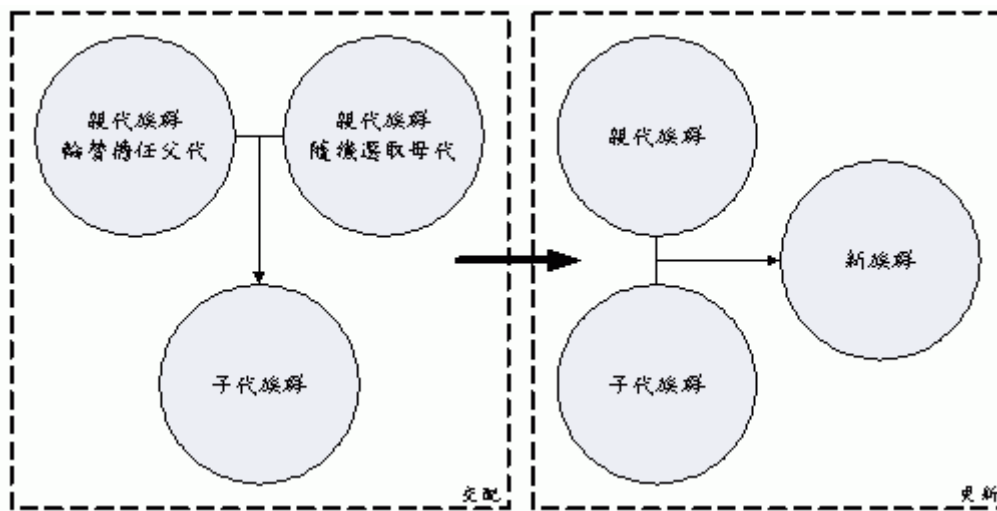


圖 4-9 族群競爭基因演算法示意圖，每次迭代將產生與親代族群大小相同之子帶族群，並以兩族群相互競爭之方式進行解集合之更新。

族群競爭基因演算法之特點在於交配時並不使用一般基因演算法常見之交配機率進行父代選取，而是將母體內所有染色體以輪流之方式作為父代之一，並配合一隨機挑選之配偶以產生子代，因此其子代將會是一大小同於母體數目之染色體族群。由於在迭代中必須維持母體之大小，故在更新上將會是父代族群與子帶族群相互競爭之狀況，最後將保留兩族群表現優秀之前半染色體，後半則被淘汰，此為族群競爭之意義。族群競爭基因演算法流程如圖 4-10 所示：

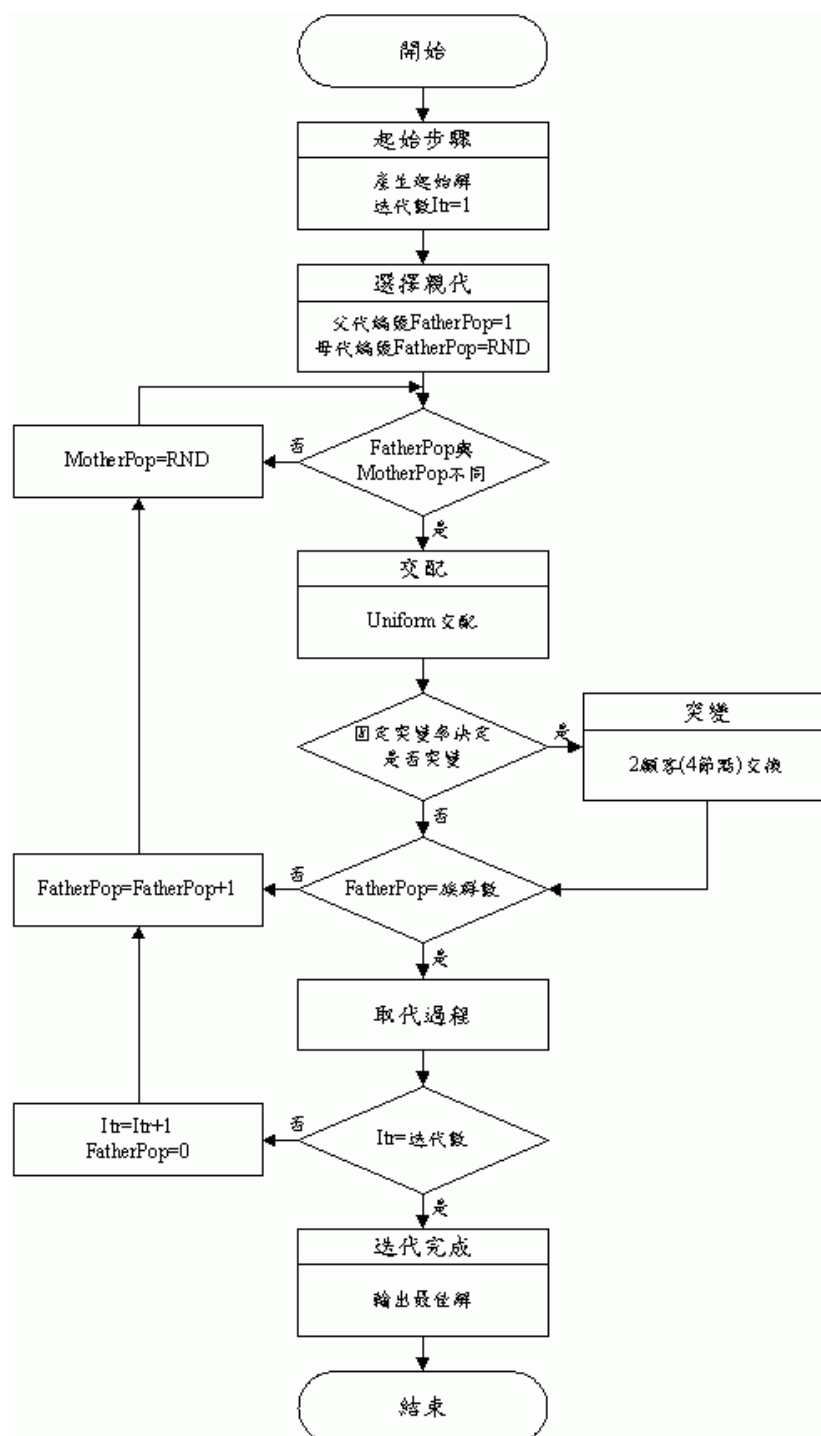


圖 4-10 族群競爭基因演算法求解流程圖

一階段族群競爭基因演算法各步驟說明如下：

1. 起始步驟

此步驟與適應性基因演算法相同，皆為初始參數以及初使解群體之產生，產生之方式亦使用亂數編排。

2. 選擇親代

族群競爭式基因演算法親代之選擇方式並非透過傳統之選擇機率進行，而是族群中所

有染色體皆輪流擔任父代，並配合一亂數選取之母代進行交配以產生新子代，如此將會產生一大小與親代族群相同之子代族群，並從兩族群中選取最優秀之數條染色體構成新親代族群以進入下一迭代。由於亂數選取之作法有可能選到與父代相同之染色體，如此交配將形同虛設，故在選取須避免此種狀況發生。

3. 交配

族群競爭式基因演算法亦採用 Uniform 運算元進行交配，惟其不具樣板保留率計算機制，故使用一固定機率產生基因樣板。在產生一子代之後亦須透過移除、重劃與再插入三個步驟以確保子代路線符合「一人一車」與「先接載後送達」之基本定義限制。

4. 突變

突變之發生採用一固定突變機率配合亂數決定，若發生突變則以兩點交換之方式進行。

5. 取代過程

族群競爭式基因演算法於此步驟進行親代與子代兩族群之競爭，在計算完子代族群中各染色體之適應度之後，由親子兩族群中選出數量符合族群大小之數個最佳染色體，並由其構成新親代進入下一搜尋迭代，其他表現較差之染色體族群則加以淘汰。

6. 停止判斷

基因演算法根據停止準則進行迭代停止之判斷，此部份亦採用最大迭代數做為停止準則，若已達到則輸出族群之最佳解。

在不同求解流程之差異上，族群競爭演算法與適應性基因演算法相同，亦以染色體編碼以及相關之起始解產生方式為主要差別。

4.3.3 蟻群演算法

螞蟻演算法為 90 年代根據螞蟻族群搜尋食物之行為模式所發展之巨集啟發式演算法，最早由 Dorigo(1991)應用於旅行推銷員問題，其求解方式為模擬螞蟻爬行所遺留下之費洛蒙濃度來進行最佳解搜尋，費洛蒙會隨著時間與螞蟻經過次數發生變化，而螞蟻會選擇費洛蒙濃度較高之路線作為行進方向。

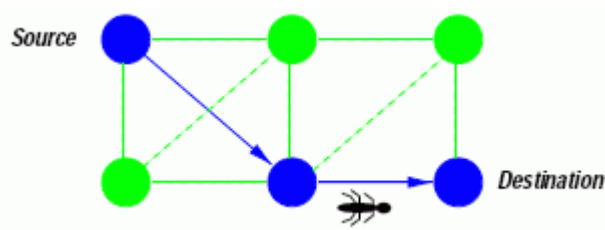


圖 4-11 蟻群演算法螞蟻構建路線示意圖

資料來源：Dorigo，1999

螞蟻演算法與其後改良而成之螞蟻群演算法已被應用於各類型 HP-Hard 問題中且具有不錯之求解績效，然國內外文獻尚無在 DARP 之應用。本研究根據 DARP 之問題特性提出一改良式之螞蟻群演算法，以候選名單之方式因應此種具成對需求點之問題形式。圖 4-12 為本研究所使用之螞蟻群演算法示意圖與各步驟說明。

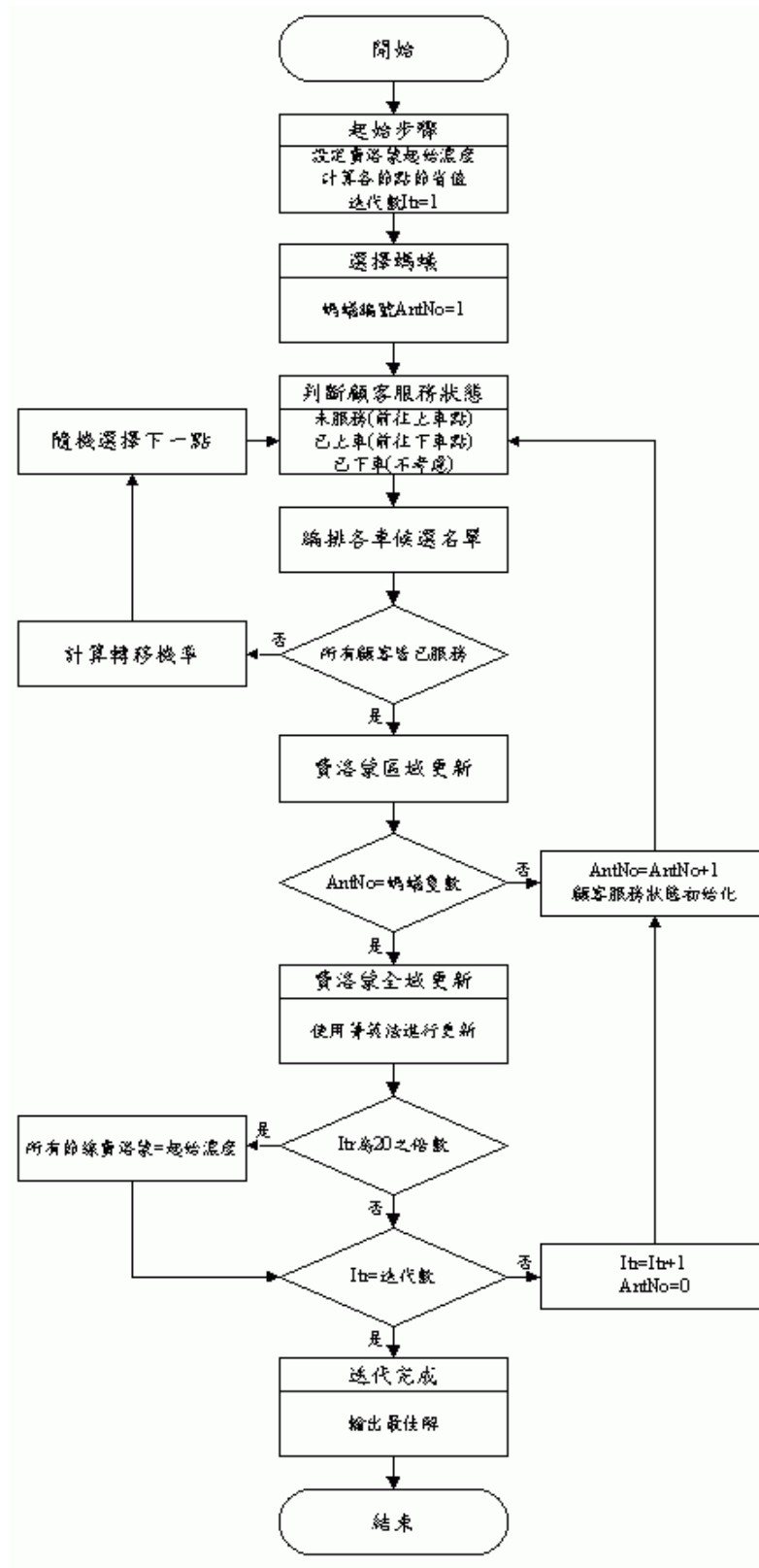


圖 4-12 螞蟻演算法求解流程圖

1. 候選名單

由於 DARP 同一顧客之起迄二節點必須位於同一路線之中，且具有先上車後下車之順序，因此本研究利用編排候選節點名單之方式限制螞蟻的下一到達點，以避免違反此二定義限制。候選名單基於顧客服務狀況依路線編排，對於每位顧客而言，若其為處於未服務之狀態，則將其起點需求填入各車候選名單；若顧客狀態為已上車，則將迄點需求填入該路線之候選名單。由於未上車之顧客由於可被各車服務，因此其起點將填入各車之候選名單之中，再透過路線構建步驟以決定下一到達點。其候選名單編列方式如圖 4-13 所示：

顧客狀態					
顧客	1	2	3	4	5
狀態	未上車	於 1 車上	未上車	已下車	於 2 車上
1 車候選節點	1 上	2 下	3 上		
2 車候選節點	1 上		3 上		5 下

候選名單				
1 車	1 上	2 下	3 上	
2 車	1 上	3 上	5 下	

圖 4-13 完整一階段候選名單編排示意圖

2. 路線構建

在蟻群演算法之中，螞蟻要從目前之節點 i 決定下一到達點 j 時，將依照轉換機率進行選擇，其公式如下：

$$v = \begin{cases} \arg \max_{j \in U} [(\tau_{ij})^\alpha (\eta_{ij})^\beta] & q \leq q_0 \\ V & q > q_0 \end{cases} \quad (4.6)$$

$$\text{其中 } V : P_{ij} = \frac{(\tau_{ij})^\alpha (\eta_{ij})^\beta}{\sum_{j \in U} (\tau_{ij})^\alpha (\eta_{ij})^\beta} \quad (4.7)$$

式中 U 為未拜訪之節點集合，於本研究中則為候選名單； τ_{ij} 為節線 ij 之費洛蒙數量； η_{ij} 為節線 ij 之距離倒數； α 與 β 則為決定 τ_{ij} 與 η_{ij} 之相對重要性參數 ($\alpha, \beta > 0$)； q 則是一個符合均一分布 (Uniform Distribution) 0 到 1 之間的隨機亂數。下一到達節點之選擇方式為若 q_0 大於 q 時採用準則 v ，否則採用準則 V 。準則 v 代表若節線 ij 之 \arg 值為最大，則節點 j 即為下一到達點；而準 V 則根據各節線 ij 之轉換機率 P_{ij} ，以 0 到 1 均一分布隨機亂數決定下一到達點。

3. 區域更新

區域更新為蟻群眼算法兩種費洛蒙更新步驟之一，為完成每個解完成構建之後立即進行的更新方式，本研究所採用的更新方式如下：

$$\tau'_{ij} = \tau_{ij} + \rho\tau_0 \quad (4.8)$$

其中 ρ 為費洛蒙揮發速度， τ_0 為費洛蒙起始濃度。

4. 全域更新

全域更新為每次迭代完成之後所進行的更新方式，本研究在全域更新方面採用菁英螞蟻策略，僅有屬於精英螞蟻路線中之節線才進行全域更新。更新方式如下：

$$\tau'_{ij} = (1 - \rho)\tau_{ij} + \rho\Delta\tau_{ij}^k \quad (4.9)$$

$$\Delta\tau_{ij}^k = \frac{L_k}{\sum_{u \in E} L_u} \quad (4.10)$$

其中 E 為精英螞蟻集合，即最佳螞蟻路線集合。

5. 費洛蒙濃度初始化

螞蟻演算法常由於經過數次迭代後現行最佳解節線上之費洛蒙不停累積，而其他節線之費洛蒙不斷揮發，導致路線選擇將更集中於現行最佳解上而陷入區域最佳解。節線費洛蒙初始化之目的在於利用一定迭代後初始化所有節線費洛蒙之作法，嘗試跳脫區域最佳解以增加搜尋之廣度，更避免發生及早收斂之問題。本研究於每 20 次迭代即進行一次費洛蒙初始化。

6. 迭代更新

螞蟻演算法於此步驟保留每次迭代中之最佳解，並清除剩餘之螞蟻路線。

7. 停止判斷

本研究在螞蟻演算法所採用之停止準亦為最大迭代數，若已達到則輸出族群之最佳解。

螞蟻演算法於候選名單編排方式、轉換機率計算方式與費洛蒙更新方式根據不同之求解流程與目的採用不同之方式進行，一階段所使用之作法已於上述，以下為二階段之作法：

1. 分群階段

本研究於分群階段之作法採用 Shelokar et al.(2004)用於分群問題之方式，其作法為螞蟻由一虛擬起點出發並依順序經過所有資料點，而螞蟻所行經之節點則為該資料所屬之分群並於其所行經之節線上留下費洛蒙以作為後續螞蟻搜尋路徑之依據。因此螞蟻之下一到達點為所有分群，此即為分群候選名單之內容。

當候選名單決定之後則須計算轉移機率以作為螞蟻挑選到達點之依據，由於到達點為節點之分群，並無距離可供計算 η_{ij} ，因此 Shelokar 等人僅以費洛蒙作為機率計算依據，其公式如下：

$$v = \begin{cases} \arg \max_{j \in U} [(\tau_{ij})^\alpha] & q \leq q_0 \\ V & q > q_0 \end{cases} \quad (4.11)$$

$$\text{其中 } V: P_{ij} = \frac{(\tau_{ik})^\alpha}{\sum_{k=1}^K (\tau_{ik})^\alpha} \quad (4.12)$$

分群階段之費洛蒙區域更新於各碼以完成路線之後，亦採用(4.8)式進行更新；全域更新則發生於路線排程完成之後，根據各路線之表現利用(4.9)式更新。

2. 繞徑階段

蟻群演算法繞徑階段所使用之操作方式與一階段作法相同，唯其候選名單編排之範圍受於分群內容之限制，而費洛蒙全域更新亦僅處理分群內乘客之相關節線。

4.4 小結

本章主要探討撥召服務最適繞徑策略，包括停等發生時機並使用前向時間鬆弛計算最適之車輛離開時間，以減少路線中支停等時間，藉此降低其所帶來的負面效應。

在演算步驟上除採用文獻中常用之分群與繞徑二階段求解方式，本研究另外設計完整一階段之求解流程，希望藉由統一演算步驟以加速搜尋，並比較兩種演算流程在求解效益上之不同。

在演算法上本研究選擇適應性基因演算法、族群競爭式基因演算法與蟻群演算法進行求解設計，配合鬆弛機制與具懲罰成本之最小成本模式以擴大迭代中之搜尋範圍，並利用子代檢驗機制與候選名單之方式維持基本合理性。根據本章所發展之求解演算法，將進一步於下一章利用例題進行測試以驗證演算法之正確性，並與文獻結果進行求解效益之比較。

第五章 案例測試

在探討與介紹完求解流程與演算法架構之後，本章依據所提出之求解演算法進行案例之測試。本研究測試分為三個階段，首先針對各演算法之求解能力利用小型問題進行測試，並比較最佳解與各演算法所求得之解，以評估演算法之求解能力；接著針對各演算法於不同參數設定之情況下對於求解效果之影響，並找出最佳之參數設定；最後針對各求解流程與演算法之最佳結果，與 Cordeau 以及 Bergvinsdottir 之結果進行比較。

本研究使用 Intel Pentium 4 2.8GHz 之個人電腦作為求解設備，並以 Cordeau 與 Laporte 撥召公車系列例題進行求解比較，此系列例題為 Cordeau 與 Laporte 根據丹麥運輸系統實際營運資料加以轉換而成，總共有包括撥召服務問題與具時窗車輛繞徑問題等 13 種車輛繞徑問題形式。所有問題可於 <http://neumann.hec.ca/chairedistributique/data/> 取得。

5.1 求解能力

本節針對各演算法之求解能力進行測試與驗證，作法為設計一簡單問題並使用窮舉法搜尋最佳解，再與各以演算法所產生之解進行比較。本研究所採用之問題規模為 2 台車輛載送 5 名乘客，乘客需求數據由 Cordeau 之撥召公車系列例題中挑選，而其他設定如顧客乘車時間與最大繞行時間限制皆沿用至其例題之設定。本研究所設計之問題需求內容為：

表 5-1 簡單問題節點設定

節點	X 座標	Y 座標	時窗下限	時窗上限
場站	-1.044	2	0	1440
1 上	-2.973	6.414	0	1440
2 上	-3.066	0.546	0	1440
3 上	-7.667	9.934	325	358
4 上	-2.067	5.789	111	152
5 上	-5.204	0.657	395	421
1 下	-5.476	1.437	258	287
2 下	-4.933	3.337	329	361
3 下	0.435	1.469	0	1440
4 下	-5.066	-2.313	0	1440
5 下	-2.283	-0.981	0	1440

在問題設計之意涵上，問題應具有不同時窗階層與空間分布之狀況，本研究希望藉此反映出撥召服務問題裡可能遇到的多種排程狀態。時窗階層上以乘客 1 與乘客 4 反應，此兩名與其他乘客在時窗上較具斷層；在空間分布上，乘客 1 與乘客 4 屬於縱向需求，而乘客 2、乘客 3 與乘客 5 屬於橫向需求，然而乘客 3 具有最長之直接路線距離，其起迄點亦與其他需

求相隔較遠，然而其具有時窗階層相近的顧客，此狀況可透過限制式以判斷是否可以進行共乘服務以及服務型式。問題採用窮舉法進行各種車輛分群以及排程之比較以找出最佳解，其最佳繞徑排程為：

表 5-2 簡單問題最佳繞徑排程

第 1 車	繞徑長度	38.461			第 2 車	繞徑長度	154.965		
節點	到達或 出發時間	停等時間	運送時間	乘載量	節點	到達或 出發時間	停等時間	運送時間	乘載量
出發	107.075				出發	266.612			
4 上	111	0		1	1 上	271.429			1
4 下	129.639	0	8.639		1 下	287		5.571	
回站	145.537				3 上	325	19.225		1
					3 下	334.021		8.775	
總繞徑長度	193.427				2 上	347.642			1
總停等時間	40.531				2 下	361		3.358	
	平均	4.053			5 上	395	21.306		1
總運送時間	29.692				5 下	408.349		3.349	
	平均	5.9384			回站	421.577			

所選擇之各節點於二維空間之分布情況與最佳繞徑排程如下圖所示：

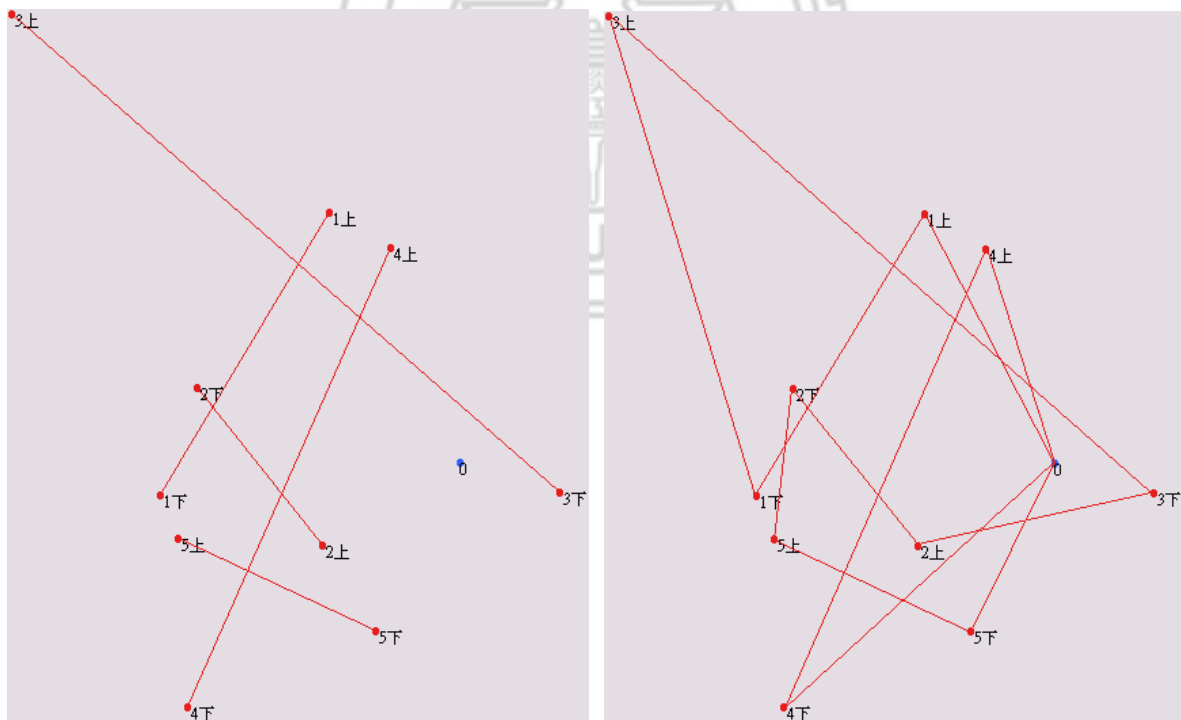


圖 5-1 簡單問題之需求狀況與最佳繞徑圖

此階段測試之演算法參數設定上族群數與迭代數之部分以最小化進行，其他參數則使用文獻最佳設定，並將各演算法進行 10 次之求解，詳細求解結果如附表 1，下表為各演算法結果與最佳解之比較：

表 5-3 簡單問題各演算法求解結果

	繞徑時間		停等時間		運送時間		是否求得最佳解
	平均	最佳	平均	最佳	平均	最佳	
最佳解		193.426		27.835		30.769	
適應性基因	224.449	193.426	31.721	21.306	48.789	30.769	是
族群競爭基因	217.927	193.426	52.247	27.835	34.68	30.769	是
蟻群演算法	250.116	205.429	82.205	43.59	77.019	37.916	否

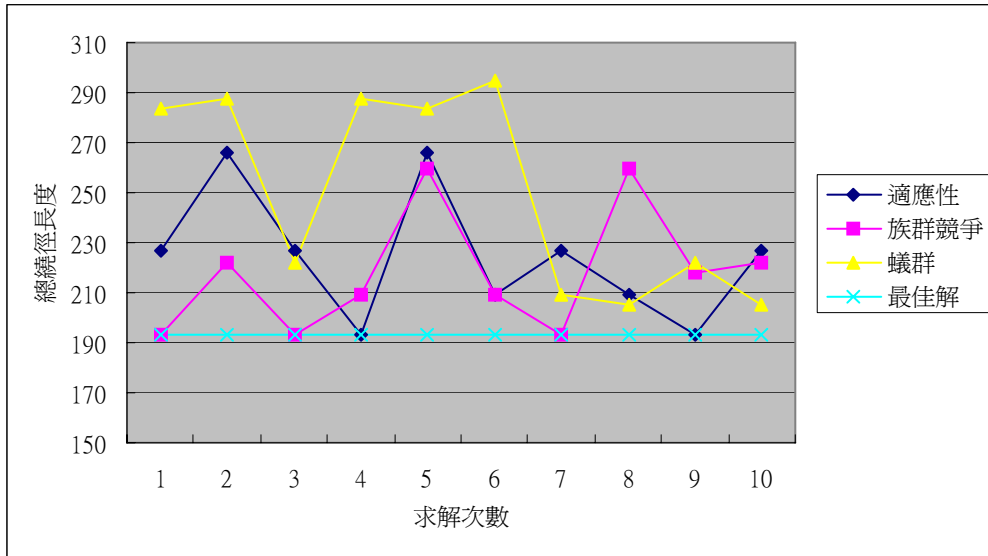


圖 5-2 適應性基因演算法、族群競爭式基因演算法與蟻群演算法簡單問題求解結果分佈

其中適應性基因演算法與族群競爭式基因演算法皆可找出最佳解，其次數分別為適應性基因演算法之 2 次與族群競爭基因演算法之 3 次；蟻群演算法雖無法找出最佳解，然而其在求解過程中車輛容量使用率較高，較能找出具有共乘行為之路線。

5.2 參數設定

此部分根據各演算法之參數進行調整，並比較調整結果以找出最適設定。各演算法所需調整之參數範圍由經由文獻與演算法設計取得，而其中部份文獻已針對某些參數進行最適比較，此部份則直接延用。尋找參數之方法為由文獻中找出該參數可能之範圍以及調整量逐次調整進行求解，完成設定之後，此部分使用 Cordeau 系列例題 pr07(4 輛車 96 個節點)進行計算，並根據各參數設定之結果進行比較以找出最適設定。比較之依據主要為透過總繞徑時間之平均值與最佳解值，並輔以總運送時間與總停等時間之平均值與最佳解值進行，而調整之作法則採用單項調整，即在進行一參數之比較求解時，其他須調整之參數為固定值。

5.2.1 適應性基因演算法

在參數設定上，由於適應性基因演算法採用迭代中自動調整交配機率、突變機率與基因樣板保留率(Template)，作法為根據上一代之機率值並利用式(4.4)計算目前迭代中各解之 gtype 值，藉此透過式(4.5)求取新的機率值。由於具有機率自我調整之機制，故適應性基因演算法僅須找出最佳族群數與迭代數。族群數與迭代數之尋找範圍如下表：

表 5-4 適應性基因演算法調整參數範圍表

參數種類	數值
族群數	{30,40,50}
迭代數	{ 10^2 , 10^3 , 10^4 , 10^5 }

適應性基因演算法其它法相關參數為交配機率、突變機率與基因樣板保留率跟此三種機率之上下限，以及變異性目標值與機率敏感度參數，此部份於 Zhu(2003)之文獻中以進行最佳參數設定之比較分析，因此本研究直接採用此部份之設定。各項參數之數值為：

表 5-5 適應性基因演算法各參數設定表

參數種類	設定值	參數種類	設定值
起始基因樣板保留率	0.4	起始突變機率	0.1
最小保留率	0.1	最小突變機率	0
最大保留率	0.4	最大突變機率	1
起始交配機率	0.8	變異性目標值	0.5
最小交配機率	0	機率敏感度參數 ξ	0.01
最大交配機率	1		

1. 族群數

基因演算法之族群數部分皆進行 30、40 與 50 個染色體數量之計算與比較，族群越大每一迭代之搜尋範圍亦越大，然而過多之族群將使得求解時間過於冗長，若較大之族群並未產生明顯之改善則代表此設定數量並無意義。因此為求計算時間之經濟，有必要進行最適族群數設定之比較。適應性基因演算法在族群數之比較表為(詳細求解結果如附表 2)：

表 5-6 適應性基因演算法族群大小比較表

族群大小	繞徑時間		停等時間		運送時間	
	平均	最佳	平均	最佳	平均	最佳
30	1414.825	1157.99	241.89	119.681	1519.554	1194.438
40	1412.306	1292.811	231.189	135.002	1515.539	792.328
50	1347.011	1180.559	212.392	106.916	1655.524	1147.763

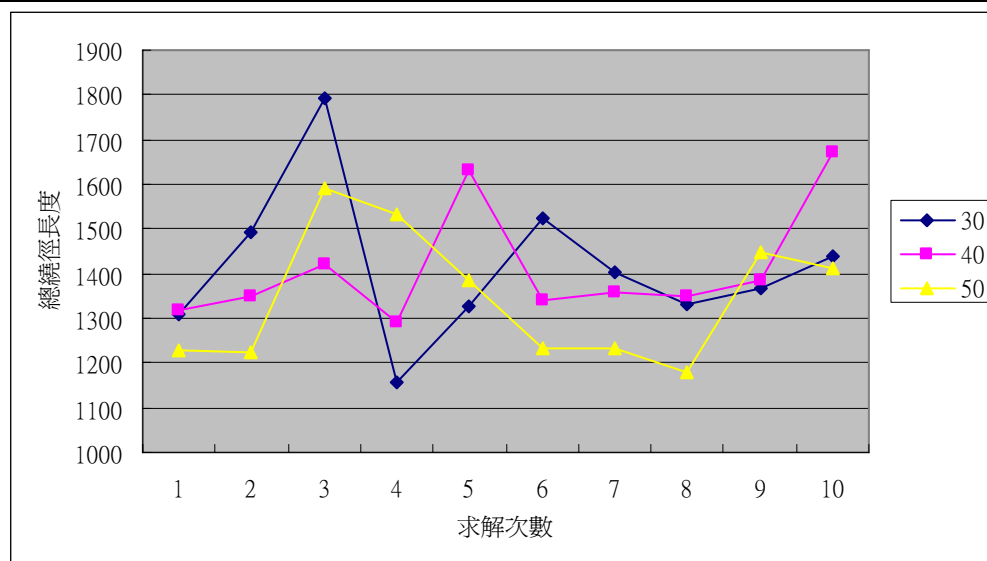


圖 5-3 適應性基因演算法各族群數求解結果分布

由表 5-6 與圖 5-3 可以看出族群數量為 50 時，所尋得之解具有較短之總繞徑時間，而其結果分布也較為集中，所產生之停等時間亦較少，故為較佳之設定。

2. 迭代數

迭代數之測試部分，適應性基因演算法測試範圍為 10^2 至 10^5 ，每次測試增量為 10 倍。迭代數影響演算法之搜尋深度，然而若演算過程已產生收斂，過多之迭代只會造成時間之浪費，固需針對最佳之迭代數進行比較。適應性基因演算法在迭代數之比較表為(詳細求解結果如附表 3)：

表 5-7 適應性基因演算法迭代數比較表

	繞徑時間		停等時間		運送時間	
	平均	最佳	平均	最佳	平均	最佳
10^2	1414.825	1157.99	241.89	119.681	1519.554	1194.438
10^3	1371.524	1183.836	218.256	114.001	1615.105	1014.083
10^4	1389.324	1209.681	189.07	130.552	1459.828	780.335
10^5	1354.178	1136.864	169.052	116.493	1579.2	1184.369

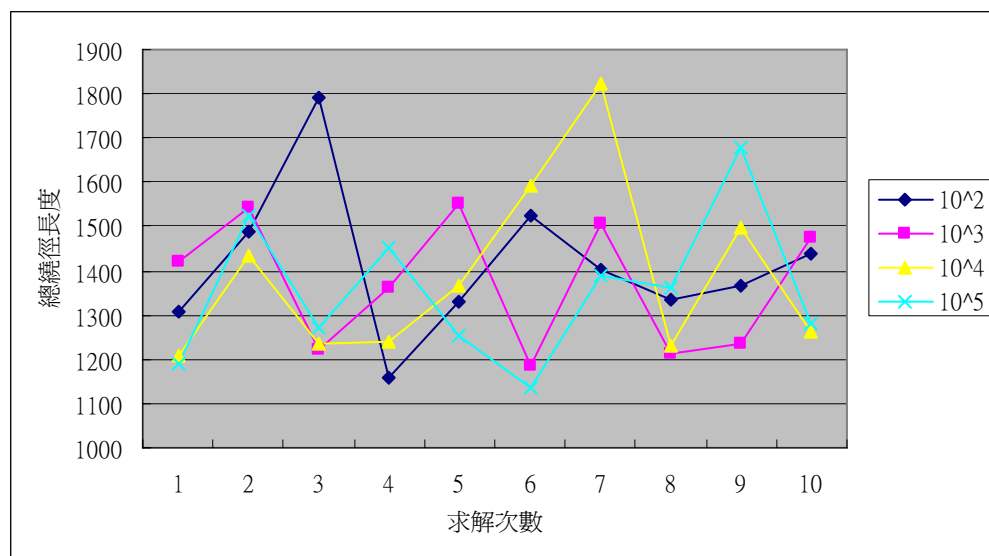


圖 5-4 適應性基因演算法各迭代數求解結果分布

由表 5-7 與圖 5-4 可看出最佳解出現於迭代數為 10^5 時，且整體尋優表現亦較其他設定值為佳且分部亦較集中，路線亦具有較少之停等時間，故為較佳之設定值。

5.2.2 族群競爭基因演算法

族群競爭基因演算法除需對族群數與迭代數進行調整外，所使用之固定突變率以及 Uniform 交配之基因樣板保留率亦必須進行最佳數值分析。各參數尋找範圍如下表：

表 5-8 適應性基因演算法調整參數範圍表

參數種類	數值
族群數	{30,40,50}
迭代數	$\{10^2, 10^3, 10^4\}$
突變率	{0.01,0.05,0.1}
樣板保留率	{0.3,0.5,0.7}

1. 族群數

族群競爭式基因演算法在族群數之比較如表 5-9 所示(詳細求解結果如附表 4)：

表 5-9 族群競爭式基因演算法族群大小比較表

	繞徑時間		停等時間		運送時間	
	平均	最佳	平均	最佳	平均	最佳
30	1417.307	1234.509	249.877	101.208	1541.703	874.941
40	1337.815	1184.128	188.190	121.923	1750.713	1123.187
50	1323.505	1172.193	186.235	127.522	1533.822	832.383

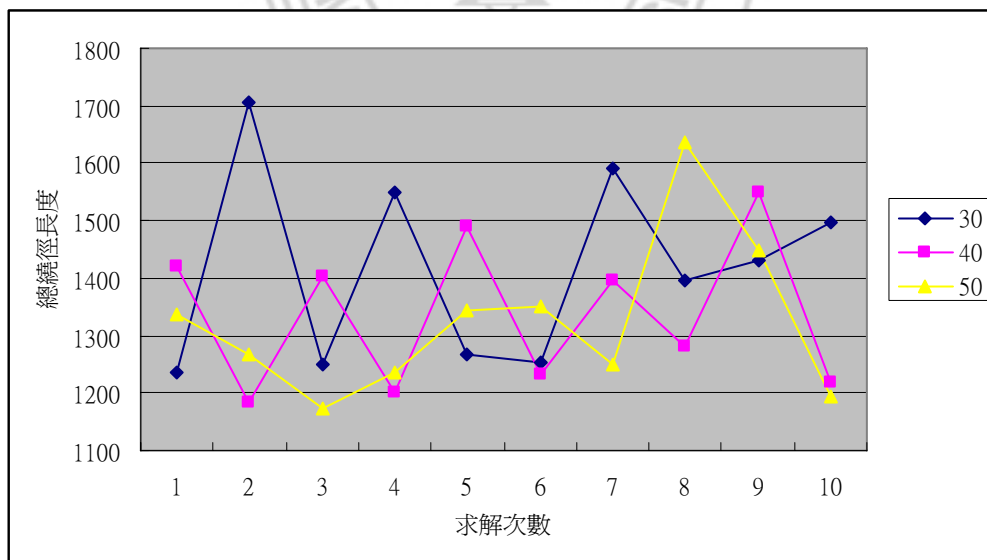


圖 5-5 族群競爭基因演算法各迭代數求解結果分布

根據表 5-9 與圖 5-5，可發現族群數為設定為 50 時其尋優能力較其他設定值為佳，且多數解之總繞徑長度分布於 1400 以下，在平均停等時間與運送時間上亦具有較好的表現，是為較佳之設定。

2. 迭代數

族群競爭式基因演算法在迭代數之比較表如表 5-10 所示(詳細求解結果如附表 5)：

表 5-10 族群競爭基因演算法迭代數比較表

	繞徑時間		停等時間		運送時間	
	平均	最佳	平均	最佳	平均	最佳
10^2	1417.307	1234.509	249.877	101.208	1541.703	874.941
10^3	1339.592	1205.720	183.898	120.263	1530.588	121.606
10^4	1287.040	1223.589	190.712	112.497	1277.725	773.790

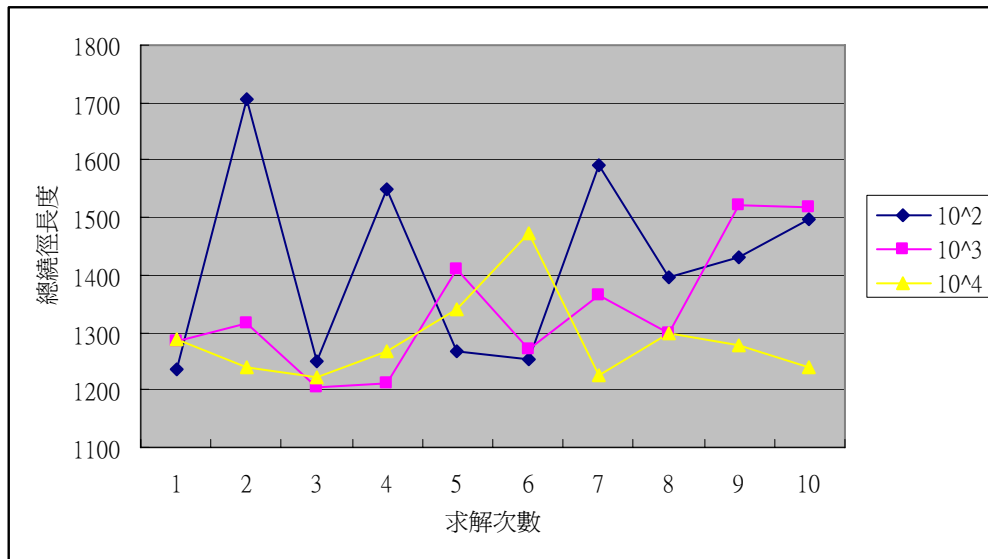


圖 5-6 族群競爭基因演算法各迭代數求解結果分布

由表 5-10 與圖 5-6 可得知最佳解出現在迭代數為 10^3 時，然而整體搜尋表現以 10^4 為佳，且其解之總繞徑時間多分布於 1300 以下，故本研究採用 10^4 做為族群競爭基因演算法之迭代數設定值。

3. 突變率

族群競爭式基因演算法在突變率之比較表如表 5-11 所示(詳細求解結果如附表 6)：

表 5-11 族群競爭基因演算法突變率比較表

	繞徑時間		停等時間		運送時間	
	平均	最佳	平均	最佳	平均	最佳
0.1	1417.307	1234.509	249.877	101.208	1541.703	874.941
0.05	1508.934	1271.416	231.097	137.091	1727.800	838.547
0.01	1446.711	1336.609	224.549	136.062	1556.026	986.850

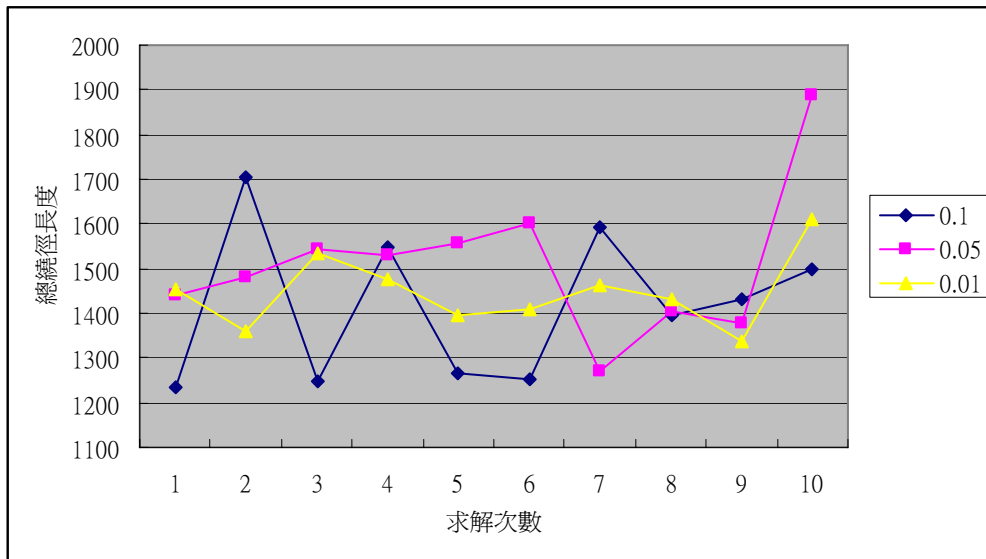


圖 5-7 族群競爭基因演算法各迭代數求解結果分布

根據表 5-11 與圖 5-7 可發現突變率為 0.01 時求解結果較為集中，然而其解並不如 0.1 時來得優秀，因此本研究選擇 0.1 作為突變率，以期在尋優過程中能找到較佳之解。

4. 樣板保留率

基因樣板保留率為 Uniform 交配時所使用之參數，代表父代基因流傳至子代之比例，一般而言此機率越高求解過程越容易收斂，亦代表越容易陷入區域最佳解。然而較低之樣板保留率雖代表較大之求解空間，但搜尋容易發散，尤其是搭配族群競爭此種針對迭代變異性改善之演算法，故須對找出較佳之樣板保留率設定。族群競爭式基因演算法在基因樣板保留率之比較表如表 5-12 所示(詳細求解結果如附表 7)：

表 5-12 族群競爭基因演算法基因樣板保留率比較表

	繞徑時間		停等時間		運送時間	
	平均	最佳	平均	最佳	平均	最佳
0.7	1417.307	1234.509	249.877	101.208	1541.703	874.941
0.5	1259.288	1162.095	143.950	101.223	1401.680	1007.253
0.3	1463.412	1254.471	285.128	107.253	1631.666	1139.059

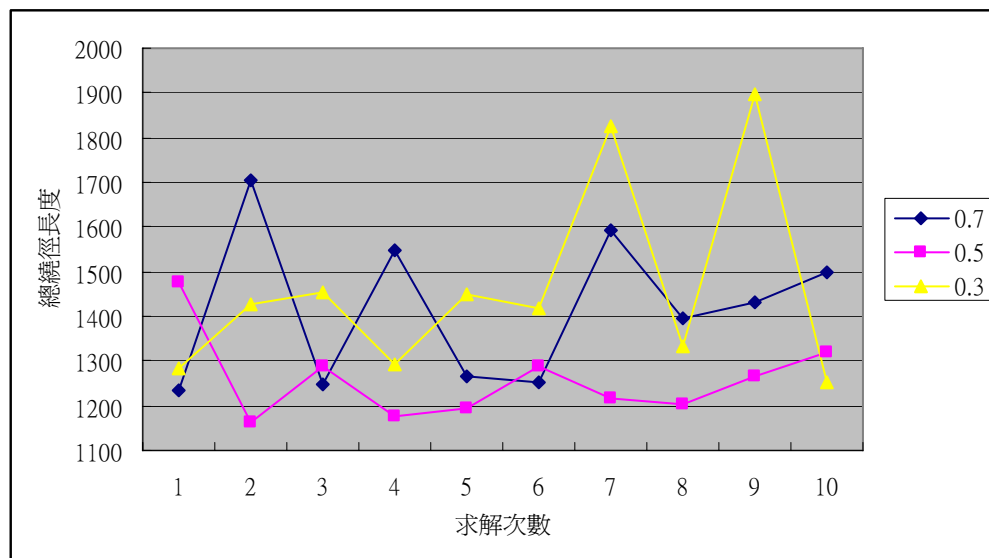


圖 5-8 族群競爭基因演算法各基因樣板保留率求解結果分布

由表 5-12 與圖 5-8 可發現樣板保留率為 0.5 時不但結果較為集中，且其表現亦較其他設定為佳，因此本研究採用 0.5 作為族群競爭基因演算法之樣板保留率。

5.2.3 蟻群演算法

近期各類組合問題之研究中對於蟻群演算法之迭代數與螞蟻隻數設定常依據需求數進行調整，另外亦有研究針對費洛蒙起始濃度進行比較。綜合相關文獻結果，本研究所需比較之參數尋找範圍如表 5-13 所示：

表 5-13 蟻群演算法調整參數範圍表

參數種類	數值
螞蟻隻數	$\{0.1N, 0.5N, N\}$
迭代數	$\{0.5N, N, 2N\}$
費洛蒙起始濃度	$\{10^{-5}, 10^{-3}, 10^{-1}\}$

其他相關參數為精英螞蟻隻數、費洛蒙揮發速度、轉換規則參數以及 α 與 β 值則沿用文獻中之常見最佳設定，此部份各參數之數值如表 5-14：

表 5-14 蟻群演算法各參數設定表

參數種類	設定值	參數種類	設定值
菁英螞蟻數	3	α	1
費洛蒙揮發速度	0.1	β	1
轉換規則參數	0.8		

1. 螞蟻隻數

對於較為簡單之問題，若螞蟻隻數過多則容易局部最佳解累積大量之費洛蒙，導致即早收斂；對於較為複雜之問題則需要較多之螞蟻隻數以期增加搜尋之多樣性，近期諸多研究則依據問題規模設定螞蟻隻數。蟻群演算法在螞蟻隻數之比較如表 5-15 所示(詳細求解結果如附表 8)：

表 5-15 蟻群演算法螞蟻隻數比較表

	繞徑時間		停等時間		運送時間	
	平均	最佳	平均	最佳	平均	最佳
0.1N	1621.390	1302.968	457.399	156.230	5704.729	4996.696
0.5N	1484.672	1213.241	354.775	172.436	5341.274	4910.761
N	1445.598	1214.685	340.752	169.768	5508.175	4935.359

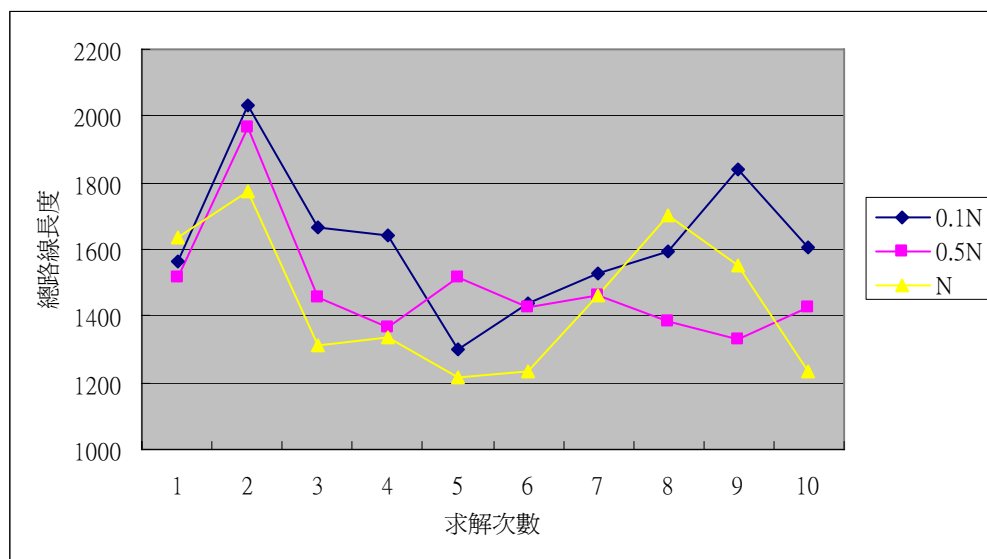


圖 5-9 蟻群演算法螞蟻隻數求解結果分布

根據表 5-15 與圖 5-9 得知，螞蟻隻數較多之設定其求解結果較佳，螞蟻隻數設定為 N 時其平均總繞徑時間以及平均停等時間皆有較佳之表現，故本研究採用 N 作為螞蟻隻數。

2. 迭代數

對於較複雜之問題需要較多之迭代數，近期研究多根據需求數來調整所需之搜尋迭代。蟻群演算法在迭代數之比較如表 5-16 所示(詳細求解結果如附表 9)：

表 5-16 蟻群演算法迭代數比較表

	繞徑時間		停等時間		運送時間	
	平均	最佳	平均	最佳	平均	最佳
0.5N	1621.390	1302.968	457.399	156.230	5704.729	4996.696
N	1547.674	1364.540	465.616	174.677	5472.963	5009.259
2N	1414.506	1245.288	270.360	157.815	5501.642	5120.965

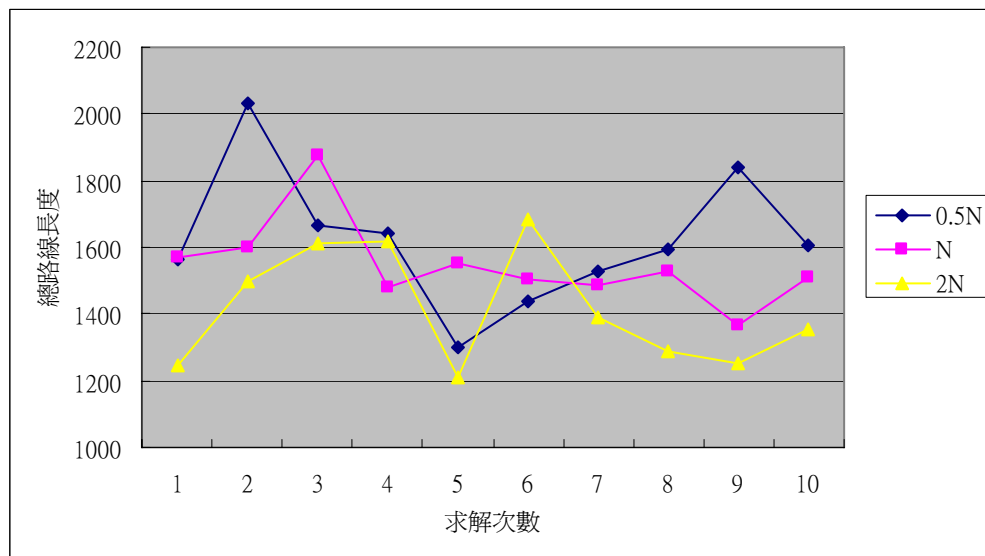


圖 5-10 蟻群演算法迭代數求解結果分布

根據表 5-16 與圖 5-10 顯示，總繞徑時間最短之解出現於迭代數為 2N 時，且其整體之求解結果優於其他設定值，平均停等時間亦較少，是為較佳之設定。

3. 費洛蒙起始濃度

考量實際狀況，由於所有路線尚未有螞蟻通過，因此費洛蒙起始濃度應該為 0。本研究嘗試以趨近為 0 之 10^{-5} 、 10^{-3} 與 10^{-1} 進行比較，其結果如表 5-17(詳細求解結果如附表 10)：

表 5-17 蟻群演算法費洛蒙起始濃度比較表

	繞徑時間		停等時間		運送時間	
	平均	最佳	平均	最佳	平均	最佳
10^{-1}	1621.390	1302.968	457.399	156.230	5704.729	4996.696
10^{-3}	1633.723	1192.121	676.009	350.847	5537.245	5127.885
10^{-5}	1674.674	1486.972	350.360	339.662	5513.437	5189.531

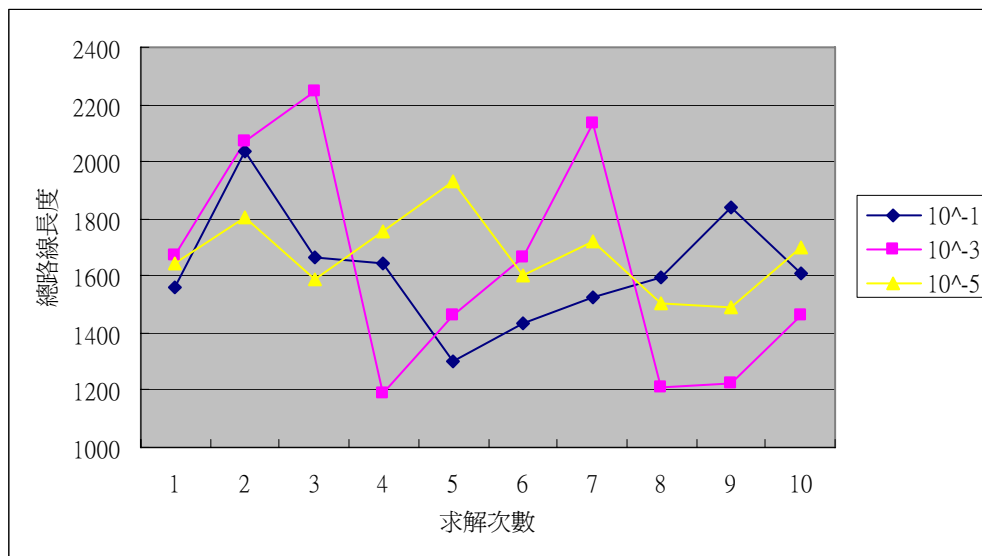


圖 5-11 蟻群演算法費洛蒙起始濃度求解結果分布

由表 5-17 與圖 5-11 所示，費洛蒙起始濃度於 10^{-5} 時分部較為集中，而最佳解出現於 10^{-3} 時，然而使用 10^{-1} 時平均求解表現較佳，其分布亦較 10^{-3} 集中，故本研究採用 10^{-1} 作為費洛蒙起始濃度之設定值。

5.2.4 參數一覽

以上各節所進行之三種演算法參數比較結果彙整如表 5-18，後續章節則使用表內各項數值以及前述章節所列之文獻最佳設定作為演算法參數。

表 5-18 各演算法參數比較結果

參數種類	數值	參數種類	數值	參數種類	數值
族群數	50	族群數	50	螞蟻隻數	N
迭代數	10^5	迭代數	10^4	迭代數	2N
		突變率	0.1	費洛蒙起始濃度	10^{-1}
		樣板保留率	0.5		

5.3 模式參數

本研究在模式關於懲罰成本之部分採用 Cordeau 之自我調整非負參數，在進行演算法參數比較時採用與文獻中相同之各模式參數起始值為 1 之設定，然而在求解過程中發現各路線中常發生成本值互補之作用，某條路線之違反常被其他路線之優勢所替補，因此所求得之解常為鬆弛解。由於此種設定方式無法充分反應違反狀況，故本研究針對經常導致解為不可行解之繞徑時間與乘車時間，將此二違反值之參數起始值加以擴大，並在迭代中進行參數自我調整時，將此二參數之最小值限制於起始值，以期能充分反映違反狀況。所有測試配合最佳設定下之適應性基因演算法進行，並根據所得解之違反狀況逐步調升數值，直到 10 次求解皆為可行解為止。

5.3.1 路線最大時間長度

Cordeau 系列例題所使用之路線時間長度限制為 480 分鐘，然而當模式所使用之自我調整參數起始值設為 1 時，常會發生違反狀況。由於繞徑時間違反值為各路線超過長度限制部份之加總，若鬆弛路線改善幅度大於此違反值，模式仍會認為此鬆弛解為較佳解，然而演算法運送結果必須為一可行解，因此必須調高此違反成本以加強反應。本研究共測試 1、5 與 10 三個數值，當違反成本放大至 10 時所有繞徑時間已符合限制式，故在此參數設定上採用此值。比較結果如表 5-19(詳細求解結果如附表 11)：

表 5-19 繞徑時間違反成本起始值設定比較表

參數值	繞徑時間		停等時間		運送時間		總違反值
	平均	最佳	平均	最佳	平均	最佳	
1	1411.959	1226.575	228.533	170.691	1648.856	1007.12	366.0403
5	1409.797	1158.469	196.551	206.359	1418.053	1702.225	135.4228
10	1488.23	1430.541	206.501	250.517	1204.057	1071.189	0

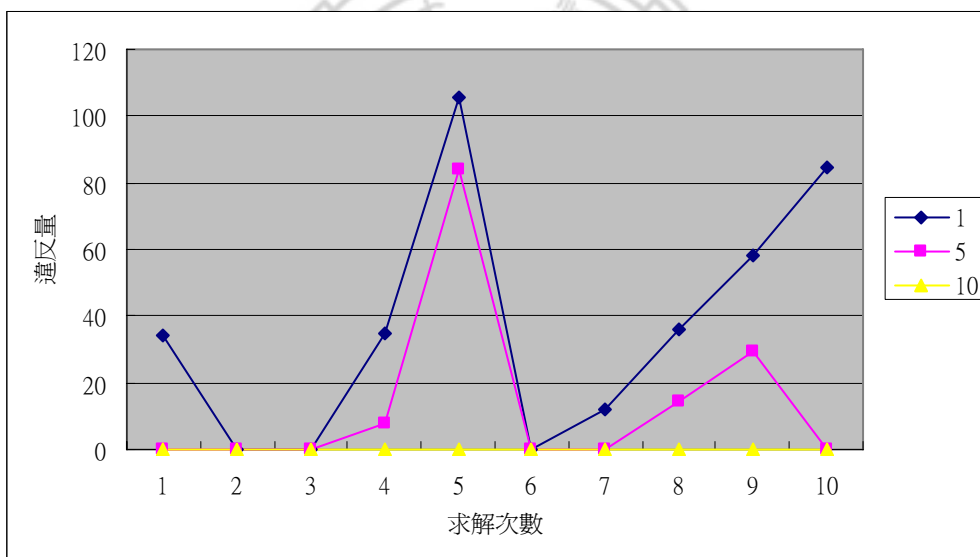


圖 5-12 繞徑時間違反成本起始值設定求解結果分布

表 5-19 為各參數值下之求解結果以及所產生之總違反值，而求解分布則如圖 5-12 所示，由其可知當路線時間違反懲罰成本之起始值設定為 10 時所有求解結果已無違反情形產生，故本研究採用 10 做為參數起始設定。

5.3.2 最大乘車時間

乘客乘車時間限制為規範乘客於車上之時間長度，此時間為乘客於車上所有時間中總和，包括車輛停等與服務其他乘客之時間。Cordeau 系列例題之乘客乘車時間限制一律為 90 分鐘，然而當其懲罰成本起始值設定為 1 時常會發生違反狀況。本研究共測試 1、5、10 與 20 四個數值，並配合繞徑時間違反成本結果進行比較，當違反成本放大至 20 時所有乘客乘車時間皆於限制之內。比較結果如表 5-20(詳細求解結果如附表 12)：

表 5-20 乘車時間違反成本起始值設定比較表

參數值	繞徑時間		停等時間		運送時間		總違反值
	平均	最佳	平均	最佳	平均	最佳	
1	1453.745	1320.431	218.081	328.918	1725.062	2029.102	810.103
5	1417.077	1303.422	226.829	197.517	1533.528	892.501	786.116
10	1552.874	1470.206	210.643	213.346	1390.225	1413.505	94.928
20	1535.133	1447.668	200.795	182.779	1329.319	1234.549	0

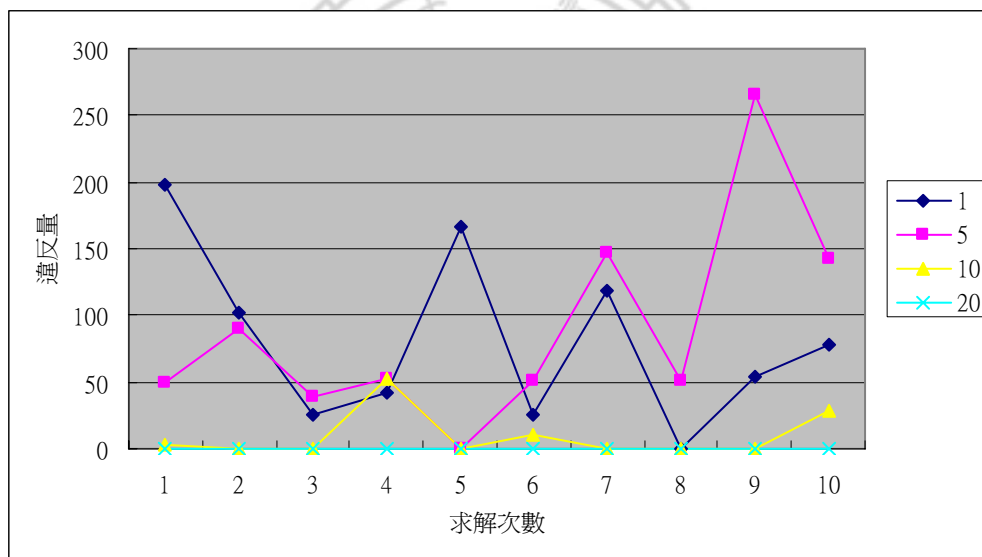


圖 5-13 繞徑時間違反成本起始值設定求解結果分布

由表 5-20 與圖 5-13 可知，當參數起始值之設定逐漸加大時解之可行性將逐漸改善，當增加至 20 時所有解之乘客乘車時間已無違反產生。

5.4 結果比較

結果比較部份亦採用 Cordeau 系列例題進行，於例題中選擇規模不同之二提例題進行測試，小型題目中採用 pr1，而大型題目則選擇 pr5，二例題之規模如下表：

表 5-21 結果分析選用例題規模

題目	顧客數	車輛數
Pr1	24	3
Pr5	120	11

各演算法對各例題進行 10 次之求解並與 Cordeau 以及 Bergvinsdottir 之結果進行比較。二階段演算法部分，分群之參數設定按照分析結果進行設定，而路線排程部分為數筆單車輛問題，其形式與規模都較為簡單，因此路線排程上僅進行 10 次迭代以加速求解。在結果比較方面，由於各文獻所使用之成本模式皆不相同，亦與本研究有所出入，因此僅以解之繞徑時間、停等時間與運送時間作為績效指標進行比較。演算法相互比較方面將針對業者與乘客兩方面進行，業者方面應希望總繞徑時間越短越好以減少整體之操作成本，而乘客方面則期待能儘速到達目的地，故越短之總運送時間對於乘客越為有利。然而繞徑時間與運送長度兩者間具有某種程度之衝突，一般而言運送時間越短代表乘客多以直接路線被運送，由於共乘狀況較少，為連結兩直接路線車輛常以空車之狀態行駛，而共乘狀況較為頻繁之路線較因為同時服務同方向之乘客以獲得繞徑時間上的節省，但是對於乘客而言則必須承受服務其他乘客所進行的繞行時間。所有演算法進行 $\alpha=0.05$ 之 t 檢定以找出對雙方較有利之求解方式。各演算法將分成一階段求解組、適應性基因演算法分群組、族群競爭演算法分群組與蟻群演算法分群組等四組，各組中包含三組求解結果，先從此三組找出最佳者再進行各組間之比較，以找出整體表現最佳之演算方式。

5.4.1 小型問題 pr1 結果比較

小型問題總繞徑時間 t 檢定結果如下表：

表 5-22 小型問題總繞徑時間各分組 t 檢定結果比較

演算方式	演算法組合	t 統計量	P 值	最佳者
一階段求解組	Adp&FC	-5.2855	0.0005	ACO
	Adp&ACO	2.2184	0.0537	
	FC&ACO	5.8564	0.0002	
Adp 分群組	Adp&FC	-2.1847	0.0567	ACO
	Adp&ACO	3.4508	0.0073	
	FC&ACO	3.7154	0.0048	
FC 分群組	Adp&FC	-3.4501	0.0073	ACO
	Adp&ACO	3.4700	0.0070	
	FC&ACO	4.5666	0.0014	
ACO 分群組	Adp&FC	-2.5065	0.0335	無
	Adp&ACO	-0.3259	0.7519	
	FC&ACO	1.5499	0.1556	

在一階段求解上以及二階段之適應性基因分群與族群競爭分群上，配合蟻群演算法進行繞徑排程之表現較為優異，而二階段之蟻群分群無法找出最適合搭配進行繞徑排程之演算法，故將三種演算法一併進行比較。整體比較表如下：

表 5-23 小型問題總繞徑時間整體 t 檢定結果比較

	P 值				
演算方式	Adp+ACO	FC+ACO	ACO+Adp	ACO+FC	ACO+ACO
ACO	-2.4421	-2.5551	-1.9882	-2.8600	-6.6114
Adp+ACO		-1.8542	0.0774	-2.2903	-0.3667
FC+ACO			1.9766	-0.7194	1.3020
ACO+Adp				-2.5065	-0.3259
ACO+FC					1.5499

由檢定結果可知一階段蟻群演算法與蟻群分群搭配適應性基因之作法在整體表現上較為顯著，較能編排出總繞行距離最短之路線。

5.4.2 大型問題 pr5 結果比較

大型問題總繞徑時間 t 檢定結果如下表：

表 5-24 大型問題總繞徑時間各分組 t 檢定結果比較

演算方式	演算法組合	t 統計量	P 值	最佳者
一階段求解組	Adp&FC	-5.3923	0.000305	Adp
	Adp&ACO	-8.2324	9.15E-06	
	FC&ACO	-7.1366	3.15E-05	
Adp 分群組	Adp&FC	-4.2201	0.001771	Adp
	Adp&ACO	-22.766	6.03E-10	
	FC&ACO	-7.7638	1.53E-05	
FC 分群組	Adp&FC	-7.3445	2.47E-05	Adp
	Adp&ACO	-9.7805	1.95E-06	
	FC&ACO	-7.0445	3.52E-05	
ACO 分群組	Adp&FC	-3.9026	0.002949	Adp ACO
	Adp&ACO	-2.1494	0.057122	
	FC&ACO	2.44143	0.034764	

大型問題之求解上適應性基因演算法之求解能力較佳，一階段、適應性基因分群以及與族群競爭分群之搭配上表現較佳；蟻群演算法分群搭配適應性基因演算法與蟻群演算法進行路線編排較能找到長度較短之路線。整體比較如下表：

表 5-25 大型問題總繞徑時間整體 t 檢定結果比較

	P 值			
演算方式	Adp+Adp	FC+Adp	ACO+Adp	ACO+ACO
Apd	-5.681	-11.417	-14.132	-7.3685
Adp+Adp		-3.179	-13.392	-5.0698
FC+Adp			-4.2335	-4.584
ACO+Adp				-2.1494

根據結果所示，大型問題中以適應性基因演算法進行一階段求解之結果較佳，較能找到總繞徑時間較短之路線。

在小型問題之處理上，蟻群演算法進行完整求解或繞徑排程之作法較能取得較短之繞行路線，而大型問題則以適應性基因演算法之表現皆較其他兩種演算法為佳。在求解流程上，一階段完整求解之表現則較二階段作法為佳。

5.4.3 文獻結果比較

在找出各演算方式之最佳結果後，需與文獻之結果作一比較。根據比較結果，一階段蟻群演算法能找到共乘率較高之路線以節省繞行時間，但也增加各旅客之乘車時間與車輛於各節點之停等時間，若以乘客之角度來看，多期望能夠以最短之時間來回起迄點，故在運送時間方面兩兩種基因演算法較為合適。在二階段演算法組合搭配上，兩種基因演算法之搭配較能找到運送時間較短之路線。整體而言，多種演算方式皆能找到總繞徑時間較 Bergvinsdottir 之結果為短之排程，而停等時間與運送時間亦比 Cordeau 之最佳解略短。

表 5-30 例題 pr1 結果表

			停等時間				運送時間				
	繞徑時間		平均		最佳		平均		最佳		
	平均	最佳	總和	平均	總和	平均	總和	平均	總和	平均	求解時間
Adp	991.435	918.185	253.419	5.28	255.964	5.333	709.112	14.773	723.733	15.078	639.650
FC	1095.063	912.625	288.225	6.005	241.608	5.749	780.384	16.258	572.937	11.936	687.583
ACO	957.809	918.185	368.723	7.682	275.964	3.666	1284.054	26.751	1423.733	29.661	424.200
Adp+Adp	1034.391	885.549	262.147	5.461	182.523	3.803	730.932	15.228	1079.949	22.499	757.183
Adp+FC	1070.746	910.795	290.573	6.054	235.01	4.896	681.768	14.204	578.962	12.062	784.333
Adp+ACO	982.81	915.141	220.706	4.598	193.252	4.026	1001.17	20.858	1066.088	22.21	702.550
FC+Adp	1087.453	941.525	288.095	6.002	233.331	4.861	702.076	14.627	630.091	13.127	868.183
FC+FC	1121.339	957.038	295.386	6.154	287.902	5.998	804.374	16.758	646.201	13.463	962.867
FC+ACO	1020.402	874.354	213.608	4.45	200.048	4.168	1093.638	22.784	1209.123	25.19	827.367
ACO+Adp	982.463	913.7	227.097	4.731	181.196	3.775	1048.961	21.853	1141.45	23.78	611.667
ACO+FC	1023.955	900.664	216.996	4.521	198.764	4.141	1060.034	22.084	1074.531	22.386	638.367
ACO+ACO	986.396	934.847	292.316	6.09	199.155	4.149	1325.698	27.619	1279.573	26.658	601.583
Cordeau		881.16			211.15	4.4			1094.99	45.62	190
Bergvinsdottir	1041	1039	252	5.25	260	5.42	477	19.86	310	12.9	557

例題 pr5 為一需利用 11 輛車服務 120 位顧客之大型問題。此部分一階段蟻群演算法仍具有較快之求解速度，二階段部分亦以使用蟻群演算法者於求解速度上具有較佳之表現，然而其在運送時間之表現並不佳。整體而言，適應性基因其求解速度雖較蟻群演算法為慢，但在求解品質上為本研究所設計之三種演算法最佳者。

在與文獻結果之比較中，總繞徑時間普遍較 Cordeau 等人之結果為長，路線所產生之停等時間亦較其結果為多，但在運送時間上已可找到較為迅速之解。與 Bergvinsdottir 之結果比較發現本研究之演算法可找到總繞徑時間較其為短之排程，且部分演算方式之送時間亦較短，然而停等時間並無法有效減少。

表 5-31 例題 pr5 結果表

			停等時間				運送時間				
	繞徑時間		平均		最佳		平均		最佳		
	平均	最佳	總和	平均	總和	平均	總和	平均	總和	平均	求解時間
Adp	4452.428	3922.886	1190.802	9.923	886.328	7.386	5626.401	46.887	4693.511	39.113	6193.300
FC	4543.621	3942.689	1282.031	10.684	1087.534	9.063	6085.503	50.713	4276.891	35.641	6306.100
ACO	4652.528	3973.553	1365.07	11.376	1181.558	9.846	7293.191	60.777	6496.211	54.135	4794.567
Adp+Adp	4569.041	3903.275	1227.108	10.226	854.89	7.124	5810.997	48.425	5109.942	42.583	6288.767
Adp+FC	4650.889	3999.047	1366.581	11.388	1162.073	9.684	6218.716	51.823	4594.319	38.286	6747.467
Adp+ACO	4805.818	4081.908	1390.95	11.591	1241.446	10.345	6925.197	57.71	6479.236	53.994	5839.217
FC+Adp	4627.315	3989.993	1340.233	11.169	1346.627	11.222	6162.527	51.354	5378.994	44.825	6891.133
FC+FC	4674.463	4070.327	1271.689	10.597	1091.195	9.093	6084.938	50.708	4095.17	34.126	6973.417
FC+ACO	4754.345	4100.632	1413.326	11.778	904.641	7.539	7082.46	59.021	6486.125	54.051	6201.800
ACO+Adp	4689.509	4062.05	1240.653	10.339	835.079	6.959	5922.187	49.352	5366.9	44.724	5676.150
ACO+FC	4796.288	4049.184	1432.368	11.936	980.27	8.169	6953.609	57.947	6729.76	56.081	6033.933
ACO+ACO	4735.835	4047.034	1412.115	11.768	1291.933	10.766	7123.112	59.359	6867.042	57.225	5264.467
Cordeau		3869.95			832.98	3.47			6156.48	51.3	4624
Bergvinsdottir	4250	4274	500	2.08	527	2.2	5099	42.49	4837	40.3	5823

5.5 小結

本研究於本章節中先利用一簡單例題針對各演算法之求解能力進行測試，再以一例題找出各演算法之最佳參數設定，最後進行大小兩種規模之例題求解並與文獻結果做一比較。

簡單例題透過窮舉法找出最佳解，再分別使用三種演算法進行求解以評估其求解能力。結果顯示兩種基因演算法皆能找到最佳解。

在各演算法參數設定上，根據文獻與演算法設計之部分尋找各參數之範圍與調整時之增量，逐次調整並求解例題已進行比較。完成各演算法最適參數搜尋之後則進行大小兩規模例題之求解，並與相關文獻進行結果比較。

1. 在求解方法上

本研究根據總繞徑時間與總乘載時間分別作為業者與顧客兩方面最適演算方式之探討。在小型例題求解上，一階段蟻群演算法與蟻群分群搭配適應性基因之作法較能找出總繞行時間最短之路線，而大型問題之求解結果則以一階段適應性基因演算法為佳。

2. 在求解時間上

根據結果比較發現，一階段之作法較為省時，若採用二階段求解流程，於分群或排程時應選用較為簡便之作法，或設置較為寬鬆之停止條件已加速流程。演算法之比較方面，蟻群演算法在求解時較為迅速，部分求解已較文獻所使用之方法為快。

第六章 隨機旅行時間問題分析

車輛繞徑問題中一個或一個以上之參數具有不確定性即為隨機性車輛繞徑問題，最常討論之隨機對象為顧客需求量、顧客需求點與旅行時間，如張世峰(民 91)曾針利用動態需求資訊探討最適車隊規模，曾惠鈺則利用動態之旅行時間分析物流車隊最佳之路徑導引策略(民 92)。本研究將針對旅行時間之部分，以表現較佳之適應性基因演算法，探討於動態旅行間下之最佳撥召服務路線排程策略。整體路線決策流程如下：

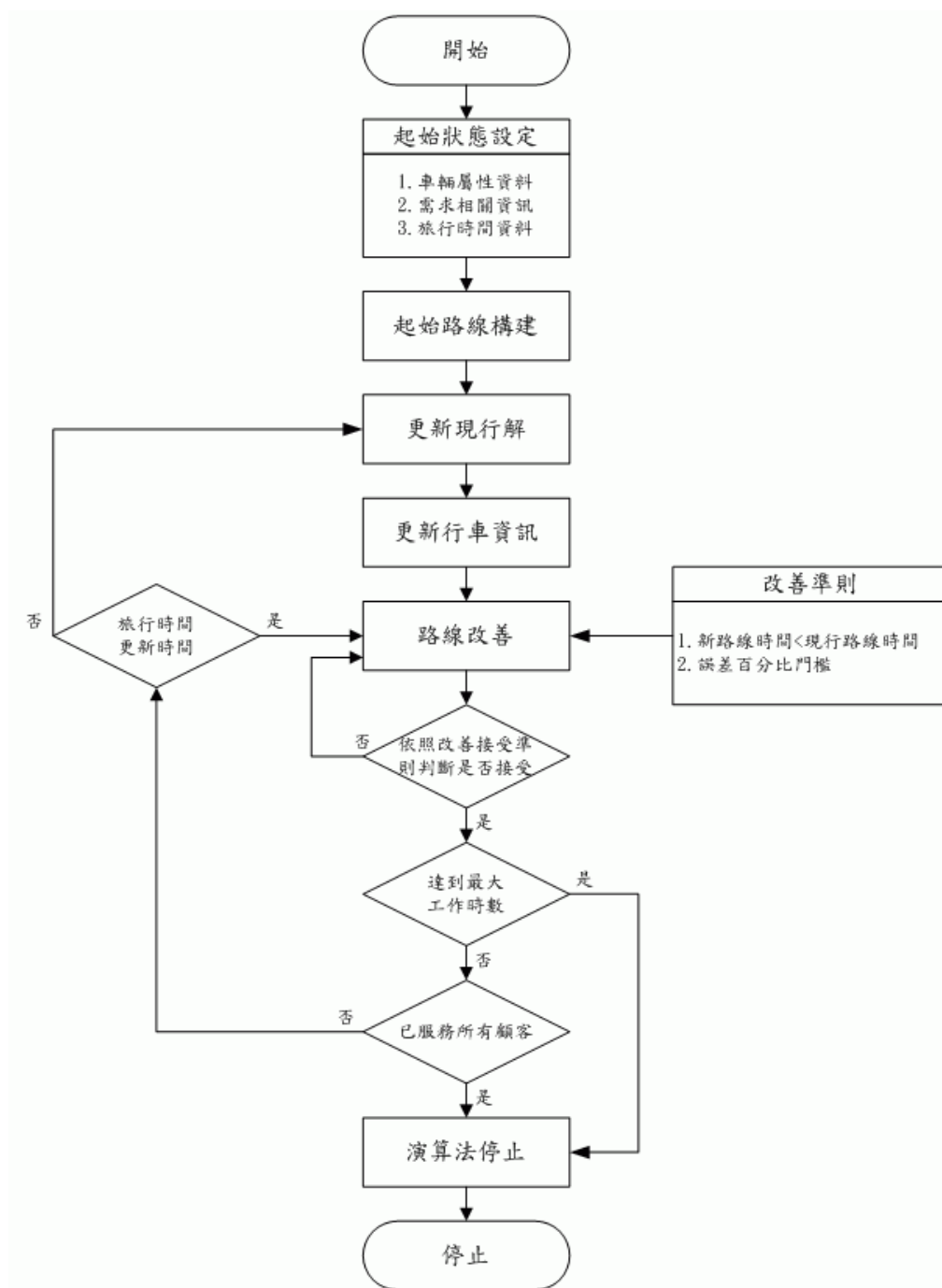


圖 6-1 隨機旅行時間問題決策流程

流程各步驟說明如下：

步驟一：起始狀態設定與起始路線構建。以例題所提供之旅行時間資料做為初使狀態，並產生起始路線排程。

步驟二：更新行車資訊。此步驟先將時間移至旅行時間發生變動之時間點，並根據變動過之旅行時間重新計算現行路線之各項時間值，然後判斷乘客之服務狀況將乘客分為未服務、已上車、已下車：

- (1) 未服務：未服務之乘客於後續路線改善中可改變其所屬路線(車輛)，以期整體路線排程具有較好之改善成效。
- (2) 已上車：已上車之乘客其起迄點資訊將不進入整體路線排程中，在新路線指派給各車輛後，再於其中搜尋最佳之插入點置入其迄點。
- (3) 已下車：已下車之乘客即已完成對其之服務，在任何變動時應略過不予計算以家數處理時程。

步驟三：路線改善。當搜尋出新排程後，根據此二改善準則進行接受與否之判斷：

- (1) 準則一：若新排程優於舊排程則接受。
- (2) 準則二：若新排程與舊排程之誤差超過門檻值則接受新路線。

步驟四：是否達到演算法最大工作時數，若達到則停止演算法更新搜尋，此後若旅行時間再度發生變動亦僅更新路線中各到達點之時間值。

步驟五：若服務完所有乘客則結束所有流程。

6.1 動態旅行時間

6.1.1 旅行時間之計算

在運輸系統規劃程序中，交通量指派之作法常用以分析運輸系統之交通需求，在考慮道路流量之下，旅行時間將會隨流量而增減，即旅行時間即為流量(或 V/C)之函數。在路網交通量指派之實務應用上，常以簡化之數學公式反應旅行時間與流量之關係，其中最被廣泛應用者為美國公共道路局(U.S. Bureau of Public Roads, BPR)之公式：

$$t_a = t_0 \left[1 + 0.15 \left(\frac{V}{C} \right)^4 \right]$$

其中

t_a ：在路段交通量為 V 之調整旅行時間

t_0 ：路段於自由車流(Free Flow)之旅行時間

V ：路段交通量

C ：道路容量

關於路網旅行時間發生變動之時間點，本研究採取車輛到達各節點之後即發生變動之作法，當路網中某輛車到達其路徑之節點時即以隨機亂數判斷路網旅行時間是否發生改變，若為是，則先產生隨機亂數作為 V/C 值，再以 BPR 公式求取新的路段旅行時間並藉此完成整個路網旅行時間之更新。

6.1.2 旅行時間統計分配假設

在旅行時間統計分配假設上，本研究假設旅行時間為滿足威伯分配(Weibull Distribution, $W(\alpha, \beta)$)之隨機變數。威伯分配為一隨機變數必大於或等於零之連續機率分配形式，其中 α 為控制機率密動函數圖形之參數(Shape Parameter)， β 為控制刻度比例之參數(Scale Parameter)。

威伯分配之定義為若一隨機變數 x 之分配符合下列機率函數，則稱其為威伯分配：

$$f(x) = \frac{\alpha}{\beta} (x - \lambda)^{\alpha-1} \exp \left[- (x - \lambda) \frac{\alpha}{\beta} \right] \quad x \geq \lambda \quad (6.1)$$

式中 α 、 β 與 λ 為此機率分配之母數， α 為型態母數(Shape Parameter)、 β 為尺度母數(Scale Parameter)且 λ 為位置母數(Location Parameter)。

當 $\alpha = 1$ 時，則威伯分配為：

$$f(x) = \frac{1}{\beta} \exp \left[- \frac{(x - \lambda)}{\beta} \right] \quad (6.2)$$

此與指數分配同型，又當 $\lambda = 0$ 時則：

$$f(x) = \frac{1}{\beta} e^{-\frac{x}{\beta}} \quad (6.3)$$

即為指數分配。

威伯分配之均數為：

$$E(X) = \lambda + \beta^{\frac{1}{\alpha}} \Gamma \left(\frac{1}{\alpha} + 1 \right) \quad (6.4)$$

變異數為：

$$V(X) = \beta^{\frac{2}{\alpha}} \left[\Gamma\left(\frac{2}{\alpha} + 1\right) - \Gamma^2\left(\frac{1}{\alpha} + 1\right) \right] \quad (6.5)$$

此分配之分配函數為：

$$F(x) = 1 - \exp\left[-(x - \lambda)\frac{\alpha}{\beta}\right] \quad x \geq \lambda \quad (6.6)$$

6.1.3 亂數產生之方法

隨機亂數(Random Number)為所有產生隨機變數(Random Variates)之基礎，真正之隨機亂數應是無法預知且無法自行產生的，因此一般常使用之隨機亂數產生方法所產生者為近似之隨機亂數。本研究則利用 Visual Basic 所提供之亂數產生功能配合反函數轉換法(Inverse Transformation)產生符合威伯分配之隨機變數，其作法為透過威伯分配之累積分配函數(Cumulative Distribution Function, CDF)求取其反函數，威伯分配之累積分配函數為：

$$F(x) = \begin{cases} 1 - e^{-\left(\frac{x}{\beta}\right)^{\alpha}} & x \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad (6.7)$$

而其反函數為：

$$F^{-1}(u) = \beta \sqrt[\alpha]{-\ln(1-u)} \quad (6.8)$$

故先產生一符合 $U(0,1)$ 之隨機亂數 u ，將其代入反函數式中即可求得符合威伯分配之隨機變數 X ，本研究則以此隨機變數作為 V/C 值更新各路段旅行時間。

6.2 撥召服務路線排程策略

6.2.1 路線之重新排程

在動態旅行時間問題裡，將因為路網旅行時間發生變動而使得原路線之表現亦有所變動，故在旅行時間改變後，將依據新旅行時間資料、車輛所在位置與乘客服務狀態等重新進行路線排程，以求得符合現況之繞徑計畫。

撥召服務旅客具有三種服務狀態：未服務、已上車與已下車。由於已下車者不再影響路線編排，縱使其旅行時間產生變動亦不再考慮，因此在路線排程更新上僅考慮未服務與已上

車者之狀況。新路線排程作法先以尚未服務之乘客以巨集啟發式解法進行路線重新編排，此步驟中所有未上車之乘客可能因為旅行時間之改變而離開原路線，並構成符合新時間資料之路線；已上車者縱使旅行時間有所改變亦無法換車，對其之處理為在完成未上車乘客路線之安排，將新路線以最鄰近法分派於各車後，再根據其所搭乘之車輛搜尋路線中之最佳插入區間將其插入。以此完成新路線之排程，並於迭代中進行路線比較以找出較佳路線。整體流程如下圖：

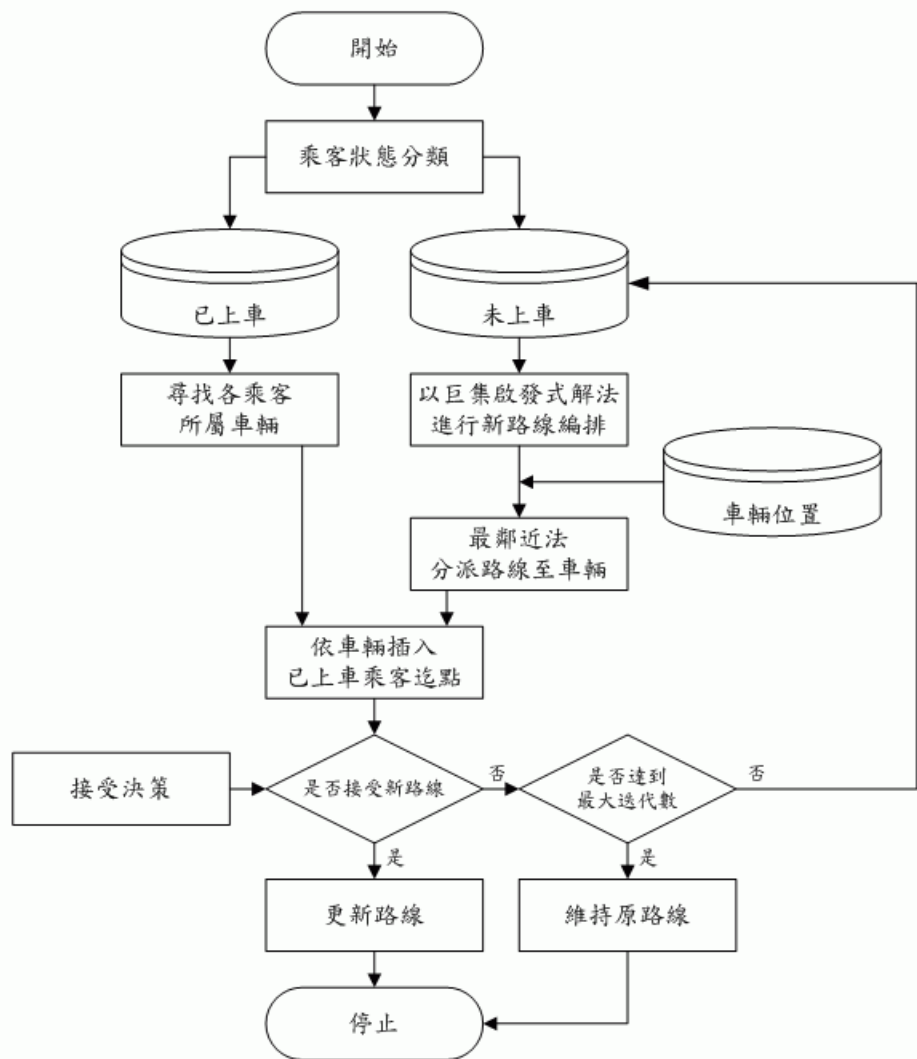


圖 6-2 動態旅行時間路線排程處理流程

6.2.2 排程接受策略

本研究於動態旅行時間之部分為探討撥召服務最佳之排程決定策略，依照所選策略之不同比較其對整體路線長度之影響，並藉此找出最佳之接受準則。在旅行時間更動之後所進行路線重新編排之搜尋中，將決策準則作為搜尋之停止條件之一，若找到符合決策準則之解則立即停止搜尋必加以更新；若已達到最大迭代數卻仍未尋獲符合條件之解則繼續以現行解作為繞徑路線。所擬定之決策準則為：

準則一：若新排程優於舊排程則接受新排程。

準則二：若新路線與舊路線之誤差超過門檻值則接受新路線。誤差之定義為兩路線旅行時間差距百分比，計算公式如下：

$$\text{誤差} = \sum \frac{\text{舊路線} - \text{新路線}}{\text{新路線}} \times 100\%$$

準則二之新舊路線門檻值之部分，本研究採用 Cordeau 系列例題中之 pr2(5 輛車 96 節點) 對 5%、10%、15%、20%與 25%各進行 10 次之測試，其結果如下所示：

表 6-1 門檻值百分比測試結果

門檻值	平均旅行時間	平均迭代數	執行時間(分)
5%	2050.520	258	152
10%	1993.641	308	207
15%	1997.356	546.5	259
20%	1930.338	601.6	281
25%	1994.486	822.9	342

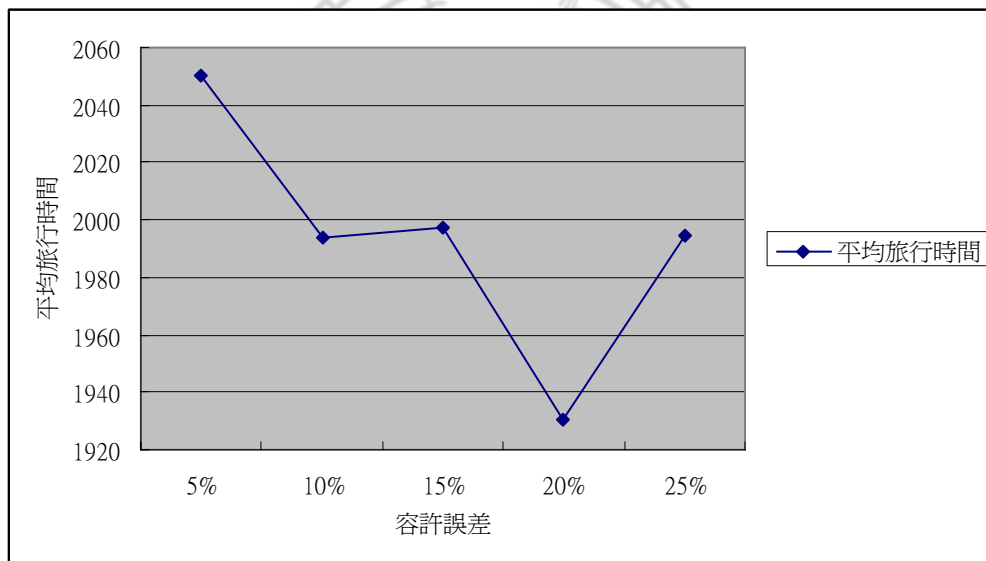


圖 6-3 門檻值百分比測試結果分佈圖

將不同門檻值在 $\alpha = 0.05$ 以 t 檢定成對比較之結果如下：

表 6-2 門檻值 t 檢定成對比較結果

門檻值		t 統計量	P 值
5%	10%	2.920706	0.017012
	15%	1.963418	0.081197
	20%	3.819886	0.004091
	25%	2.279707	0.048586
10%	15%	-0.17279	0.866637
	20%	2.913531	0.017212
	25%	-0.06621	0.948654
15%	20%	3.384837	0.008065
	25%	0.14094	0.891022
20%	25%	-3.6573	0.005257

根據上述結果發現當門檻值為 20%時求解結果較佳，且其求解時間較為合理，故在決策比較上採用門檻值 20%之設定。

6.3 策略比較

6.3.1 測試結果分析

根據新路線之接受決策準則，將兩接受策略分別進行例題 pr2 之 10 次求解，並以 t 檢定與 F 檢定等方法檢定兩決策之母體平均數差以及母體變異數是否具有差異。

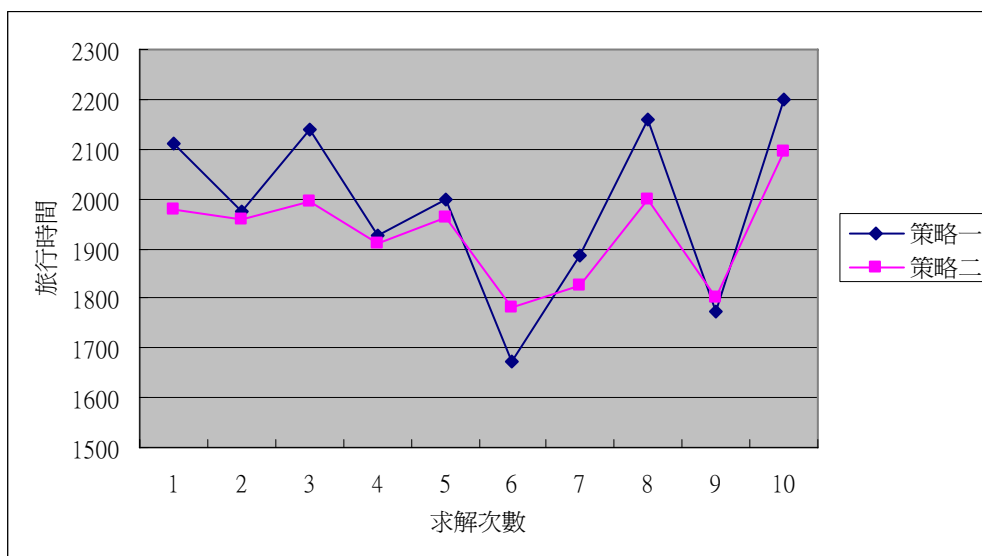


圖 6-4 不同求解策略旅行時間分布

1. 解集合範圍

針對兩種策略之求解結果集合範圍進行分析，以了解不同決策準則之表現，兩策略之最佳值與最差值如下表：

表 6-3 不同策略解集合範圍

策略一		策略二	
最佳值	最差值	最佳值	最差值
1673.576	2199.843	1780.278	2095.877

決策一之分布範圍為 526.267，決策二之分布範圍為 315.599。

2. t 檢定

在 $\alpha = 0.05$ 檢定結果如下表：

表 6-4 不同策略 t 檢定結果

	策略一	策略二
平均數	1983.921	1930.338
變異數	30236.58	10107.14
皮耳森相關係數	0.949849	
t 統計量	2.006127	
P(T<=t) 單尾	0.0379	
臨界值：單尾	1.833114	
P(T<=t) 雙尾	0.0758	
臨界值：雙尾	2.262159	

結果發現以 20% 做為路線更新接受門檻值之作法雖然平均旅行時間較低，但與策略一之差距並不顯著。

3. F 檢定

在 $\alpha = 0.05$ 檢定結果如下表：

表 6-5 不同策略 F 檢定結果

	策略一	策略二
平均數	1983.921	1930.338
變異數	30236.58	10107.14
F 統計量	2.991607	
P(F<=f) 單尾	0.059102	
臨界值：單尾	3.178897	

根據結果表示，兩者變異程度並無差異。

6.3.2 測試結果小結

繞徑排程接受策略檢定分析結果如下：

1. 在解答分布範圍上

配合門檻值 20%之接受策略其結果分佈範圍較小，解集合分佈較為集中。

2. 平均值之表現

以 10 求解之平均值來看，接受門檻值 20%之策略其平均旅行時間較低，然透過 t 檢定發現兩策略並無顯著差異。

3. 變異程度

透過 F 檢定發現兩決策之變異程度並無顯著差異。

整體看來以新舊路線旅行時間差之 20%做為路線改變之接受準則，其解答分佈較為集中，是為表現較佳之策略。



第七章 結論與建議

人類生活型式日趨多元，對於時間價值的判定與交通服務上的自主性之要求越來越高，傳統固定路線與固定班次之大眾運輸服務已經很難滿足所需。而在社會福利之角度，針對行動不便者所進行之無障礙設計，若只從系統硬體面下手不僅建置與維護成本過高，在使用與位置上亦無法滿足所有交通需求。撥召服務所具有之彈性路線與班次所帶來之及門性與即時性，不但能降低大眾運輸對於行動不便者之使用門檻，一般乘客也能從中獲得自主性並減少無謂之交通時間。

撥召服務問題為針對撥召服務系統之路線排程發展而成之車輛繞徑問題，為一種具時窗之撿取配送問題，除了必須遵守一般減取配送問題所具有之「一人一車」與「先接載後送達」之定義限制外，更必須於模式中反應乘客之不舒適度以突顯服務水準，此為撥召服務問題與其他車輛繞徑問題最大之不同點。

7.1 結論

本研究所得到的結論如下：

一、問題的定義上：

文獻回顧中發現撥召服務問題須針對乘客之不舒適度進行定義與反應，本研究於第二章針對各文獻所使用之反應方式進行探討，並作為模式構建之依據。另外，撥召服務問題於求解中允許停等之發生，此時乘客之不舒適度會大幅提高，因此應在何時發生停等亦是撥召公車問題研究所需探討的重點。經過文獻回顧後發現，將停等發生在車輛到達節點之後與車輛開始進行該節點之服務之前，此時發生停等對於乘客之影響最小。

二、模式的構建上：

模式構建上使用車輛總繞行時間以及各車路線長度兩項反應業者操作成本，並利用停等時間、時窗限制與乘車時間作為衡量服務水準之旅客不舒適度指標，透過具懲罰成本之最小成本模式將違反值反應至成本中。採用此種模式之另一種考量為增加最佳解搜尋之廣度。

三、求解演算法的設計上：

諸多文獻採用節點插入法與再插入法作為鄰近解搜尋方式，因其在設計上較容易尋獲可行解，然而此種作法在搜尋上之效率不彰，常需配合搜尋空間之限制機制以減少迭代中無謂之搜尋。在求解流程上本研究嘗試設計完整一階段之求解方式，並與較為常見之分群排程兩階段作法進行比較。演算法方面則嘗試利用巨集啟發式解法進行求解，發展出用以求解撥召服務問題之適應性基因演算法、族群競爭基因演算法與蟻群演算法，透過子代修正與候選名單機制以維持解之基本可行性，並配合最小成本目標式進行尋優。

四、案例測試上：

首先針對所設計之三種演算法其求解能力進行測試，本階段使用一小型問題，透過窮舉法找出最佳解，並與三種演算法所尋得之解進行比較，得知此三種演算法的確具有求解能力。另外針對各演算法所使用之參數進行比較分析，以找出最適合之求解設定，最後在進行例題之求解與比較。結果發現此三種演算法在撥召服務問題之求解上具有與Cordeau 等人之演算法相當之求解能力。

五、在隨機問題之探討上：

本研究嘗試將例題中之路段旅行時間以隨機之方式於迭代中進行變換，以模擬符合現實路往之動態旅行時間，並藉此探討撥召公車問題於隨機旅行時間下最佳之路線更新接受策略。結果發現以新舊路線總旅行時間差 20%做為接受門檻值之策略求解結果較為集中，是為較佳之策略。

7.2 建議

根據研究中，發現幾點值得特別注意，可供後續研究參考。

一、演算法方面：

對於基因演算法方面，本研究在考量維持解集合基礎合理性之下採用 Uniform 交配運算元作為交配方法，在突變運算中亦採用兩點交換之方式，應可思考新的編碼方式並配合其他交配運算元進行求解。

在蟻群演算法方面，本研究發現其在求解速度上較為迅速，但是其解之品質並不理想，可針對候選名單之編排或轉換機率之計算等方面進行改善。近期諸多研究皆針對費洛蒙更新之方式進行改善以避免即早收斂之問題，本研究在此方面採用多隻精英螞蟻以及費洛蒙重置機制進行，未來若有更具效益之更新方式應該進行測試與比較。

二、路線改善策略：

本研究於隨機旅行時間問題所使用之路徑改善準則採用門檻值之概念，當改善程度高於門檻值才接受新路線。本研究僅採用固定門檻值之作法，然而當系統之需求點越來越少時，新路線之改善幅度將越來越難以超越門檻，如此將產生無益之迭代搜尋，未來應可考慮變動門檻值之想法進行改良。

三、車隊規模

車輛方面，本研究使用一般車輛，然而現實生活之撥召服務常被作為身心障礙者之輔助運輸系統，例如國內各區之復康巴士。此類車輛通常具有特殊設施以服務具特殊需求之乘客，未來可針對此類課題進行特殊乘客或混合載客問題型態之探討。

車隊方面，本研究依據例題之設定採用固定規模，由於車輛繞行計畫對於業者而言屬於短期的決策，業者可根據當天的需求量指派適當數量之車輛進行服務。故對於業者而言車對規模為一重要議題，場站中是否有足夠車隊以供調度，以及容量不足時之應變策略應可為後續研究探討之方向。

四、在應用方面

在旅行時間之計算上，本研究是以各節點間之直線距離作為考量，隨機問題亦以此作為動態旅行時間模擬之基礎，然而現實路網之流量受上下游路段之影響，並非單純之兩點直線之思考方式可以處理，故在實際應用上勢必無法符合業者與乘客之需求，建議後續研究可根據實際路網資料進行旅行時間之設定與變動。



參考文獻

- [1] M. Dorigo, V. Maniezzo and A. Colony, Ant System: An Autocatalytic Optimizing Process, 1991, Technical Report, p.91-016, Politecnico di Milano, Italy, 1991.
- [2] M.W.P. Savelsbergh, The General Pickup and Delivery Problem, *Transportation Science*, vol.29 (1), p.17-29, 1995.
- [3] Bernd Bullnheimer, Richard F. Hart and Cristine Strauss, An improved ant system algorithm for the vehicle routing problem, 1997.
- [4] M.Dorigo and L.M. Gambardella, Ant colonies for the travelling salesman problem, *BioSystems*, vol.43, p.73-81, 1997.
- [5] Toth P and Vigo D, Heuristic algorithms for the handicapped persons transportation problem. *Transportation Science*, vol.31, p.60-71, 1997.
- [6] Marco Dorigo and Gianni Di Caro, The Ant Colony Optimization Meta-Heuristic, *New Ideas in Optimization*, 1999.
- [7] Dusan Teodorovic and Gordana Radivojevic, A fuzzy logic approach to dynamic Dial-A-Ride problem, *Fuzzy Sets and Systems* 116 p.23-33, 2000.
- [8] Karl Doerner, Manfred Gronalt, Richard F. Hartl, Marc Reimann, Christine Strauss and Michael Stummer, SavingsAnts for the Vehicle Routing Problem, *Vienna University of Economics and Business Administration*, December, 2001.
- [9] Jean-François Cordeau and Gilbert Laporte, A tabu search heuristic for the static multi-vehicle dial-a-ride problem, *Transportation Research Part B* 37 p.579–594, 2003.
- [10] Jean-François Cordeau and Gilbert Laporte, The Dial-a-Ride Problem (DARP): Variants, modeling issues and algorithms, *French and Italian Operations Research Societies* 1 p.89-101, 2003.
- [11] Kenny Q. Zhu, A Diversity-controlling Adaptive Genetic Algorithm for the Vehicle Routing Problem with Time Windows, *Department of Computer Science, National University of Singapore*, 2003.
- [12] Kenny Q. Zhu, Population Diversity in Genetic Algorithm for Vehicle Routing Problem with Time Windows, *Department of Computer Science, National University of Singapore*, 2003.
- [13] Andrea Attanasio, Jean-François Cordeau, Gianpaolo Ghiani and Gilbert Laporte, Parallel Tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem, *Parallel Computing* 30 p.377–387, 2004.
- [14] K.B. Bergvinsdottir, The Genetic Algorithm for solving the Dial-a-Ride Problem, *Kgs. Lyngby*, 2004.
- [15] Marco Diana and Maged M. Dessouky, A new regret insertion heuristic for solving large-scale dial-a-ride problems with time windows, *Transportation Research Part B* 38 p.539–557, 2004.
- [16] P.S. Shelokar, V.K. Jayaraman and B.D. Kulkarni, An ant colony approach for clustering, *Analytica Chimica Acta*, 509, p.187-195, 2004.

- [17] 游進俊，靜態撥召服務問題啟發式解法之研究，國立交通大學土木工程研究所，民國八十二年。
- [18] 蘇昭銘與楊琮平，先進撥召公車營運管理系統之研究，中華管理學報，第一卷第一期，第 89-114 頁，民國八十九年。
- [19] 張世峰，即時行車資訊下物流配送作業規劃之研究，淡江大學運輸管理學系運輸科學碩士班碩士論文，民國九十一年六月。
- [20] 邱裕鈞、馮正民與朱文正，考量旅行時間可靠度之車輛途程問題－螞蟻族群演算法之應用，中華民國運輸學會第 18 屆論文研討會，民國九十二年。
- [21] 陳家和與丁慶榮，應用螞蟻演算法於車輛途程問題之研究，中華民國運輸學會第 18 屆論文研討會，民國九十二年。
- [22] 曾惠鈺，即時行車資訊下物流配送作業規劃之研究，淡江大學運輸管理學系運輸科學碩士班碩士論文，民國九十二年六月。
- [23] 紀婉容與許永真，以族群競爭式基因演算法解決具有次序及時間限制的載運路徑規劃問題，台大工程學刊第九十期第 89-98 頁，民國九十三年二月。
- [24] 林典翰與楊烽正，優加劣減螞蟻擇段系統應用於零工式生產排程問題，第一屆台灣作業研究學會學術研討會暨 2004 年科技與管理學術研討會，民國九十五年。




附錄 求解結果表

附表一 簡單問題求解結果

演算法	Adp				FC			
	路線長度	停等時間	運送時間	求解時間	路線長度	停等時間	運送時間	求解時間
	193.426	27.835	30.769	1.667	193.426	27.835	30.769	1.567
	193.426	27.835	30.769	1.517	193.426	27.835	30.769	1.850
	209.499	43.590	30.769	1.600	193.426	27.835	30.769	1.450
	209.499	43.590	30.769	1.567	209.499	43.590	30.769	1.450
	226.835	21.306	75.821	1.350	209.499	43.590	30.769	1.800
	226.835	21.306	75.821	1.783	218.106	51.810	51.860	1.567
	226.835	21.306	75.821	1.700	221.743	52.815	37.916	1.400
	226.835	21.306	75.821	1.667	221.743	52.815	37.916	1.683
	265.651	44.568	30.769	1.533	259.201	97.174	32.634	1.867
	265.651	44.568	30.769	1.250	259.201	97.174	32.634	1.583
總和	2244.492	317.210	487.898	15.633	2179.270	522.473	346.805	16.217
平均	224.449	31.721	48.790	1.563	217.927	52.247	34.681	1.622
演算法	Ant							
	路線長度	停等時間	運送時間	求解時間				
	205.429	45.333	60.181	1.583				
	205.429	45.333	60.181	1.450				
	209.499	43.590	30.769	1.600				
	221.743	52.815	37.916	1.433				
	221.743	52.815	37.916	1.567				
	283.895	112.304	114.553	1.567				
	283.895	112.304	114.553	1.383				
	287.208	117.924	96.592	1.367				
	287.208	117.924	96.592	1.567				
	295.114	121.707	120.939	1.483				
總和	2501.163	822.049	770.192	15.000				
平均	250.116	82.205	77.019	1.500				

附表二 適應性基因演算法不同族群數求解結果

族群數	30				40			
	路線長度	停等時間	運送時間	求解時間	路線長度	停等時間	運送時間	求解時間
	1157.990	170.035	1242.038	12.950	1292.811	135.002	1560.037	33.733
	1309.056	125.193	1686.532	12.567	1319.650	142.019	1030.844	32.683
	1328.493	119.681	1699.300	13.333	1341.685	149.199	1334.787	33.583
	1332.619	148.410	1492.345	13.333	1349.444	161.169	1606.507	33.533
	1367.342	200.789	1194.438	13.733	1350.470	181.169	792.328	30.317
	1404.497	232.954	1459.402	13.600	1358.162	162.433	1553.494	31.433
	1439.779	276.115	1550.497	12.517	1385.593	225.942	1788.615	33.967
	1490.580	262.937	1442.270	13.200	1422.100	253.357	2129.671	31.850
	1526.013	254.651	1798.969	12.400	1629.441	436.566	1996.538	32.200
	1791.878	628.135	1629.748	12.683	1673.707	465.038	1362.565	32.683
總和	14148.246	2418.900	15195.539	130.317	14123.063	2311.894	15155.386	325.983
平均	1414.825	241.890	1519.554	13.032	1412.306	231.189	1515.539	32.598
族群數	50							
	路線長度	停等時間	運送時間	求解時間				
	1180.559	106.916	1147.763	67.433				
	1224.925	127.345	2741.708	66.467				
	1230.567	129.784	1204.378	62.867				
	1230.818	119.037	2007.120	64.567				
	1231.649	151.396	1568.141	65.350				
	1385.351	145.730	1499.616	63.383				
	1413.903	215.371	1308.178	66.700				
	1446.456	227.505	1288.867	62.783				
	1535.014	326.560	1907.881	62.900				
	1590.865	574.276	1881.587	64.267				
總和	13470.108	2123.920	16555.239	646.717				
平均	1347.011	212.392	1655.524	64.672				

附表三 適應性基因演算法不同迭代數求解結果

迭代數	10^2				10^3			
	路線長度	停等時間	運送時間	求解時間	路線長度	停等時間	運送時間	求解時間
	1157.99	170.035	1242.038	12.95	1183.836	135.505	1297.381	145.883
	1309.056	125.193	1686.532	12.567	1214.746	114.001	1373.324	150.95
	1328.493	119.681	1699.3	13.333	1219.372	151.806	1214.351	152.8
	1332.619	148.41	1492.345	13.333	1236.373	134.563	1668.291	147.367
	1367.342	200.789	1194.438	13.733	1361.45	142.041	1668.449	149.367
	1404.497	232.954	1459.402	13.6	1422.353	208.004	1545.216	165.417
	1439.779	276.115	1550.497	12.517	1475.921	222.673	1508.153	154.533
	1490.58	262.937	1442.27	13.2	1504.773	327.844	2672.528	157.05
	1526.013	254.651	1798.969	12.4	1542.508	350.422	2189.278	157.867
	1791.878	628.135	1629.748	12.683	1553.905	395.698	1014.083	154.967
總和	14148.25	2418.9	15195.54	130.317	13715.24	2182.557	16151.05	16151.05
平均	1414.825	241.89	1519.554	13.032	1371.524	218.256	1615.105	1615.105
迭代數	10^4				10^5			
	路線長度	停等時間	運送時間	求解時間	路線長度	停等時間	運送時間	求解時間
	1209.681	146.663	1023.375	466.7	1136.864	116.493	1184.369	1231.617
	1435.317	253.613	1352.896	490.233	1191.009	116.506	1219.259	1172.767
	1235.61	230.02	1471.57	456	1251.509	118.281	1911.838	1214.3
	1239.136	180.098	780.335	465.533	1273.577	155.96	1501.108	1224.6
	1368.022	157.699	1546.108	484.917	1281.404	129.454	1979.11	1204.833
	1593.249	154.407	1270.848	482.833	1362.856	173.236	1571.012	1179.267
	1820.988	215.528	1776.105	471.1	1389.182	248.035	1844.514	1184.65
	1229.676	130.552	1565.627	471.983	1451.919	152.94	1320.837	1198.467
	1497.127	256.354	2145.582	471.583	1525.804	236.519	1391.064	1211.05
	1264.433	165.761	1665.835	462.217	1677.65	243.096	1868.891	1195.283
總和	13893.24	1890.695	14598.28	1536.2	13541.78	1690.52	15792	12016.83
平均	1389.324	189.07	1459.828	153.62	1354.178	169.052	1579.2	1201.683

附表四 族群競爭基因演算法不同族群數求解結果

族群數	30				40			
	路線長度	停等時間	運送時間	求解時間	路線長度	停等時間	運送時間	求解時間
	1234.509	153.204	968.803	19.983	1184.128	136.547	1166.975	50.950
	1249.644	139.904	1541.199	21.583	1201.863	145.083	1123.187	51.400
	1252.105	178.669	2057.094	19.800	1217.976	121.923	1382.736	50.417
	1266.141	101.208	1293.191	22.100	1233.911	129.429	2374.892	52.150
	1394.292	199.480	874.941	21.033	1280.436	159.281	2222.130	51.650
	1432.556	196.326	2232.562	20.300	1396.601	183.262	2292.993	50.150
	1497.508	298.549	1785.721	20.383	1403.297	208.853	1129.662	49.983
	1548.466	351.798	1695.929	20.100	1419.454	182.708	2085.417	52.300
	1591.631	392.645	1198.040	20.283	1489.691	296.315	1184.967	51.267
	1706.219	486.990	1769.551	22.683	1550.795	318.501	2544.170	51.417
總和	14173.071	2498.773	15417.031	208.248	13378.152	1881.902	17507.129	511.684
平均	1417.307	249.877	1541.703	20.825	1337.815	188.190	1750.713	51.168
族群數	50							
	路線長度	停等時間	運送時間	求解時間				
	1172.193	143.486	963.269	104.283				
	1195.000	127.522	1736.286	102.700				
	1234.175	138.257	1141.646	101.933				
	1250.382	152.988	1495.014	102.433				
	1267.169	165.001	1517.505	97.383				
	1336.512	149.741	2201.318	101.783				
	1343.139	199.462	1563.991	103.633				
	1351.669	132.816	1502.428	99.633				
	1449.557	252.086	832.383	101.567				
	1635.254	400.988	2384.383	104.083				
總和	13235.050	1862.347	15338.223	1019.431				
平均	1323.505	186.235	1533.822	101.943				

附表五 族群競爭基因演算法不同迭代數求解結果

迭代數	10^2				10^3			
	路線長度	停等時間	運送時間	求解時間	路線長度	停等時間	運送時間	求解時間
	1234.509	153.204	968.803	19.983	1205.720	135.858	1112.497	267.167
	1249.644	139.904	1541.199	21.583	1210.411	136.421	1207.597	264.583
	1252.105	178.669	2057.094	19.800	1269.983	122.018	1331.045	264.533
	1266.141	101.208	1293.191	22.100	1284.155	153.983	1045.088	269.267
	1394.292	199.480	874.941	21.033	1297.870	120.263	2084.915	272.650
	1432.556	196.326	2232.562	20.300	1316.721	152.171	1601.574	268.317
	1497.508	298.549	1785.721	20.383	1363.761	239.227	1248.898	264.317
	1548.466	351.798	1695.929	20.100	1409.463	185.466	2372.553	268.500
	1591.631	392.645	1198.040	20.283	1516.840	245.334	1268.063	269.200
	1706.219	486.990	1769.551	22.683	1520.993	348.235	2033.645	272.717
總和	14173.071	2498.773	15417.031	208.248	13395.917	1838.976	15305.875	2681.251
平均	1417.307	249.877	1541.703	20.825	1339.592	183.898	1530.588	268.125
迭代數	10^4							
	路線長度	停等時間	運送時間	求解時間				
	1223.589	121.606	1399.544	1310.967				
	1223.635	143.297	1193.452	1326.767				
	1238.462	160.754	1023.836	1317.733				
	1240.721	179.098	2039.657	1326.533				
	1267.140	122.656	1429.814	1325.733				
	1276.921	221.729	1539.884	1317.933				
	1286.911	196.520	1320.066	1319.883				
	1299.730	182.500	773.790	1325.317				
	1339.373	377.044	1195.188	1324.817				
	1473.920	201.916	862.019	1316.350				
總和	12870.402	1907.120	12777.250	13212.033				
平均	1287.040	190.712	1277.725	1321.203				

附表六 族群競爭基因演算法不同突變率求解結果

突變率	0.1				0.05			
	路線長度	停等時間	運送時間	求解時間	路線長度	停等時間	運送時間	求解時間
	1234.509	153.204	968.803	19.983	1271.416	140.525	1100.554	21.067
	1249.644	139.904	1541.199	21.583	1375.582	137.091	1542.899	20.800
	1252.105	178.669	2057.094	19.800	1402.244	196.405	1686.281	20.117
	1266.141	101.208	1293.191	22.100	1441.291	213.323	838.547	19.367
	1394.292	199.480	874.941	21.033	1478.634	196.405	1686.281	21.483
	1432.556	196.326	2232.562	20.300	1528.917	286.474	1933.777	22.917
	1497.508	298.549	1785.721	20.383	1545.313	214.990	2290.170	20.117
	1548.466	351.798	1695.929	20.100	1555.557	290.990	1864.621	19.067
	1591.631	392.645	1198.040	20.283	1601.249	418.134	2616.977	19.233
	1706.219	486.990	1769.551	22.683	1889.140	216.636	1717.895	21.150
總和	14173.071	2498.773	15417.031	208.248	15089.343	2310.973	17278.002	205.318
平均	1417.307	249.877	1541.703	20.825	1508.934	231.097	1727.800	20.532
突變率	0.01							
	路線長度	停等時間	運送時間	求解時間				
	1336.609	136.062	1815.195	19.383				
	1357.895	170.735	1483.573	21.867				
	1394.821	136.450	1039.661	19.750				
	1410.200	172.627	1667.488	20.333				
	1430.099	213.696	1322.092	21.917				
	1451.649	456.645	1468.318	21.183				
	1462.069	228.305	1416.062	21.883				
	1477.933	268.907	2110.654	19.200				
	1535.339	264.586	2250.367	20.883				
	1610.496	197.474	986.850	20.217				
總和	14467.110	2245.487	15560.260	206.616				
平均	1446.711	224.549	1556.026	20.662				

附表七 族群競爭基因演算法不同基因樣板保留率求解結果

保留率	0.3				0.5			
	路線長度	停等時間	運送時間	求解時間	路線長度	停等時間	運送時間	求解時間
	1234.509	153.204	968.803	19.983	1162.095	101.391	867.402	21.050
	1249.644	139.904	1541.199	21.583	1175.514	101.223	1086.038	21.867
	1252.105	178.669	2057.094	19.800	1195.929	118.928	961.170	20.983
	1266.141	101.208	1293.191	22.100	1204.889	132.131	2578.003	21.433
	1394.292	199.480	874.941	21.033	1214.598	128.482	2607.401	22.500
	1432.556	196.326	2232.562	20.300	1267.394	155.072	988.302	22.450
	1497.508	298.549	1785.721	20.383	1286.761	190.189	1157.561	19.433
	1548.466	351.798	1695.929	20.100	1289.991	135.728	1273.953	20.700
	1591.631	392.645	1198.040	20.283	1317.452	121.692	1339.186	20.517
	1706.219	486.990	1769.551	22.683	1478.262	254.661	1157.783	21.883
總和	14173.071	2498.773	15417.031	208.248	12592.885	1439.497	14016.799	212.816
平均	1417.307	249.877	1541.703	20.825	1259.289	143.950	1401.680	21.282
保留率	0.7							
	路線長度	停等時間	運送時間	求解時間				
	1254.471	107.253	1139.059	19.350				
	1285.144	141.061	1585.972	21.217				
	1292.532	212.513	1811.014	21.967				
	1333.166	454.319	1403.680	22.500				
	1417.016	160.557	1691.607	21.500				
	1425.518	427.059	2006.987	22.367				
	1449.279	193.037	1416.643	19.183				
	1452.201	374.862	1245.952	21.817				
	1826.891	553.588	2302.328	19.550				
	1897.906	227.030	1713.418	19.867				
總和	14634.124	2851.279	16316.660	209.318				
平均	1463.412	285.128	1631.666	20.932				

附表八 蟻群演算法不同螞蟻隻數求解結果

隻數	0.1N				0.5N			
	路線長度	停等時間	運送時間	求解時間	路線長度	停等時間	運送時間	求解時間
	1302.968	156.572	5869.506	9.683	1332.322	172.968	4911.115	50.333
	1435.740	337.264	5565.946	10.483	1368.194	195.960	4937.605	51.417
	1527.186	362.102	5593.293	11.283	1382.670	295.473	5232.547	50.067
	1562.791	445.612	5637.061	10.383	1423.611	243.531	5519.034	49.583
	1596.903	390.486	5490.178	11.850	1427.224	248.634	5359.110	50.433
	1606.008	401.266	6105.669	9.700	1456.692	449.109	5498.887	50.700
	1642.802	487.528	4995.102	10.717	1460.679	430.402	5546.442	51.050
	1665.433	520.573	5351.761	11.933	1513.241	594.192	5357.857	51.933
	1840.681	679.066	5928.917	10.167	1514.538	167.758	4922.133	51.600
	2033.390	727.522	6507.718	11.733	1967.550	753.412	6131.173	52.833
總和	16213.902	4507.991	57045.151	107.933	14846.721	3551.439	53415.903	509.950
平均	1621.390	450.799	5704.515	10.793	1484.672	355.144	5341.590	50.995
隻數	N							
	路線長度	停等時間	運送時間	求解時間				
	1214.685	237.946	5764.550	106.050				
	1234.486	238.010	5763.841	109.533				
	1236.306	170.377	5470.999	105.367				
	1310.926	179.984	5181.788	108.700				
	1336.492	169.318	4934.837	105.317				
	1459.028	357.873	5758.621	108.400				
	1551.952	339.869	6239.529	107.367				
	1635.450	366.720	5614.267	107.033				
	1703.400	446.039	5659.458	107.567				
	1773.250	550.416	5310.870	109.117				
總和	14455.975	3056.552	55698.760	1074.450				
平均	1445.598	305.655	5569.876	107.445				

附表九 蟻群演算法不同迭代數求解結果

迭代數	0.5N				N			
	路線長度	停等時間	運送時間	求解時間	路線長度	停等時間	運送時間	求解時間
	1302.968	156.572	5869.506	9.683	1364.54	174.677	5164.493	38.300
	1435.740	337.264	5565.946	10.483	1481.375	313.037	5009.259	36.400
	1527.186	362.102	5593.293	11.283	1488.957	355.183	5286.127	36.950
	1562.791	445.612	5637.061	10.383	1502.657	1147.701	5833.198	35.083
	1596.903	390.486	5490.178	11.850	1510.645	427.041	5373.192	36.467
	1606.008	401.266	6105.669	9.700	1526.433	342.249	5725.761	37.517
	1642.802	487.528	4995.102	10.717	1550.337	396.22	5477.794	37.433
	1665.433	520.573	5351.761	11.933	1572.821	386.25	6107.146	36.133
	1840.681	679.066	5928.917	10.167	1602.919	425.115	5222.339	38.700
	2033.390	727.522	6507.718	11.733	1876.054	688.687	5530.322	36.517
總和	16213.902	4507.991	57045.151	107.933	15476.738	4656.160	54729.631	369.500
平均	1621.390	450.799	5704.515	10.793	1547.674	465.616	5472.963	36.950
	2N							
	路線長度	停等時間	運送時間	求解時間				
	1209.304	364.918	5379.268	82.333				
	1245.288	364.494	5402.542	83.783				
	1251.804	357.815	5811.044	79.200				
	1288.771	231.187	5680.846	80.800				
	1354.502	212.947	5181.788	83.350				
	1388.334	375.943	5172.78	80.883				
	1497.101	356.916	5550.339	81.250				
	1613.767	371.15	5881.056	83.350				
	1615.101	366.105	5835.794	79.667				
	1681.086	502.127	5120.965	82.783				
總和	14145.058	3503.602	55016.422	817.400				
平均	1414.506	350.360	5501.642	81.740				

附表十 蟻群演算法不同費洛蒙起始濃度求解結果

濃度	10^{-1}				10^{-3}			
	路線長度	停等時間	運送時間	求解時間	路線長度	停等時間	運送時間	求解時間
	1302.968	156.572	5869.506	9.683	1192.121	350.847	5302.129	9.533
	1435.740	337.264	5565.946	10.483	1211.964	459.955	5729.821	11.683
	1527.186	362.102	5593.293	11.283	1221.975	461.95	5407.798	11.450
	1562.791	445.612	5637.061	10.383	1462.919	630.963	5468.548	11.283
	1596.903	390.486	5490.178	11.850	1464.808	1062.097	5957.865	10.700
	1606.008	401.266	6105.669	9.700	1661.758	509.745	5061.496	10.183
	1642.802	487.528	4995.102	10.717	1670.917	525.488	5127.885	11.183
	1665.433	520.573	5351.761	11.933	2069.934	841.9	5676.99	10.350
	1840.681	679.066	5928.917	10.167	2131.915	864.51	6059.974	10.983
	2033.390	727.522	6507.718	11.733	2248.922	1052.64	5579.948	9.817
總和	16213.902	4507.991	57045.151	107.933	16337.233	6760.095	55372.454	107.167
平均	1621.390	450.799	5704.515	10.793	1633.723	676.010	5537.245	10.717
濃度	10^{-5}							
	路線長度	停等時間	運送時間	求解時間				
	1486.972	360.747	5612.54	11.850				
	1506.326	339.662	5486.035	11.800				
	1585.969	371.033	5706.845	10.467				
	1603.413	395.519	5457.794	9.883				
	1646.695	442.488	5348.251	10.833				
	1700.561	503.045	5455.801	10.083				
	1720.855	626.441	5699.721	9.900				
	1759.047	576.674	5189.531	9.750				
	1806.817	576.085	5796.114	10.167				
	1930.08	716.142	5381.735	11.317				
總和	16746.735	4907.836	55134.367	106.050				
平均	1674.674	490.784	5513.437	10.605				

附表十一 不同車輛最大路線長度違反值自我調整懲罰成本起始值求解結果

起始值	1				5			
	路線長度	停等時間	運送時間	違反量	路線長度	停等時間	運送時間	違反量
	1226.575	170.691	1007.12	0	1158.469	206.359	1702.225	29.382
	1273.707	265.038	1162.565	34.205	1221.23	164.268	1742.838	83.993
	1301.626	303.21	1869.164	105.539	1330.116	264.549	1001.078	0
	1372.193	225.938	1663.269	35.094	1363.936	167.543	1423.479	0
	1385.593	225.942	1788.615	58.266	1371.796	266.64	1378.325	0
	1390.865	150.967	1843.644	84.754	1385.895	184.109	1177.661	0
	1490.588	228.135	1629.748	36.080	1426.73	152.829	1627.912	0
	1490.59	298.549	1785.721	0	1511.11	157.056	1471.439	7.547
	1572.826	221.798	1920.028	0	1518.398	281.435	963.472	0
	1615.031	195.058	1818.69	12.102	1810.287	120.719	1692.098	14.501
總和	14119.59	2285.326	16488.56	366.040	14097.97	1965.507	14180.53	135.423
平均	1411.959	228.533	1648.856	36.604	1409.797	196.5507	1418.053	13.542
起始值	10							
	路線長度	停等時間	運送時間	違反量				
	1430.541	250.517	1071.189	0				
	1441.744	193.433	1126.566	0				
	1458.004	269.635	1080.945	0				
	1467.701	208.815	1152.381	0				
	1470.734	249.39	1240.414	0				
	1484.431	109.464	866.719	0				
	1501.626	203.21	1869.164	0				
	1524.828	179.966	1611.256	0				
	1529.868	218.779	1101.913	0				
	1572.826	181.798	920.028	0				
總和	14882.3	2065.007	12040.58	0				
平均	1488.23	206.501	1204.058	0				

附表十二 不同乘客最大乘車時間違反值自我調整懲罰成本起始值求解結果

起始值	1				5			
	路線長度	停等時間	運送時間	違反量	路線長度	停等時間	運送時間	違反量
	1366.619	137.991	1653.384	197.5742	1303.422	197.517	892.501	39.01985
	1369.232	148.866	1533.598	102.721	1320.179	353.74	1901.153	52.20306
	1339.882	147.363	1312.559	25.76036	1348.474	234.824	1260.558	0
	1441.347	172.448	1647.971	41.30698	1369.055	205.991	1390.456	51.04606
	1595.601	128.595	2426.429	167.1754	1403.834	235.161	1402.69	146.9515
	1614.947	434.965	1932.245	25.68974	1432.299	153.622	1720.254	48.87578
	1416.411	244.603	1903.095	118.274	1445.816	353.74	1601.153	146.314
	1443.98	220.784	1374.711	0	1456.872	161.105	1745.721	89.75881
	1628.997	216.271	1437.528	54.32963	1527.583	240.582	1820.689	50.96533
	1320.431	328.918	2029.102	77.27123	1563.235	132.008	1600.101	164.9812
總和	14537.45	2180.804	17250.62	810.1024	14170.77	2268.29	15335.28	790.1156
平均	1453.745	218.0804	1725.062	81.01024	1417.077	226.829	1533.528	79.01156
起始值	10				20			
	路線長度	停等時間	運送時間	違反量	路線長度	停等時間	運送時間	違反量
	1470.206	213.346	1413.505	0	1447.668	182.779	1234.549	0
	1490.411	190.142	1386.012	0	1495.833	235.046	1037.769	0
	1505.289	247.619	1418.991	0	1517.129	236.115	1121.776	0
	1513.756	164.256	1322.547	3.516855	1525.996	196.053	1673.951	0
	1535.078	244.983	1765.524	10.47379	1531.043	204.01	1238.06	0
	1538.866	230.409	1204.127	52.36437	1545.662	130.781	1444.751	0
	1558.132	250.653	1181.004	0	1560.819	187.328	1294.399	0
	1602.176	152.872	1866.793	0	1566.763	203.695	1365.592	0
	1620.485	229.213	1045.422	0	1571.833	167.362	1711.636	0
	1694.337	182.931	1298.321	28.57294	1588.588	264.785	1170.706	0
總和	15528.74	2106.424	13902.25	94.92795	15351.33	2007.954	13293.19	0
平均	1552.874	210.6424	1390.225	9.492795	1535.133	200.7954	1329.319	0