

國立交通大學

網路工程研究所

碩士論文

多群轉乘的合夥用車系統

A Multi-Pool Transferring Carpooling System

研究生：彭宣翰

指導教授：陳 健 教授

中華民國一百年七月

多群轉乘的合夥用車系統

A Multi-Pool Transferring Carpooling System

研究生：彭宣翰

Student : Hsuan-Han Peng

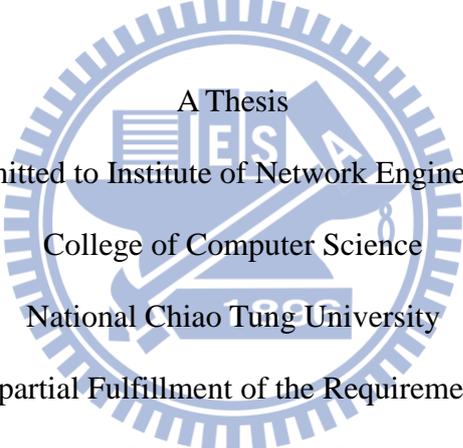
指導教授：陳 健

Advisor : Chien Chen

國立交通大學

網路工程研究所

碩士論文



A Thesis
Submitted to Institute of Network Engineering
College of Computer Science
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of
Master
in
Computer Science

July 2011

Hsinchu, Taiwan, Republic of China

中華民國一百年七月

多群轉乘的合夥用車系統

研究生：彭宣翰

指導教授：陳 健

國立交通大學網路工程研究所

中文摘要

交通運輸和上下班通勤已經成為我們日常生活中不可或缺的一部份。個人駕駛私人用車，由於時間容易掌握，且路線可由自己決定，是最常被人們使用的移動方式。也因為如此，車輛總數不斷上升，並讓交通問題日益嚴重。大眾運輸系統是一種解決上述問題的手段。但不幸的是，它並無法提供多樣化的路線自由和彈性，也沒辦法像私用車一樣舒服，甚至會有誤點的可能。合夥用車是一個創新的傳輸概念，利用駕駛和多個乘客分享私用車，減低車輛的使用數目，提高使用效率。它同時有著個人駕駛的路線彈性，又有大眾運輸系統解決交通管理問題的能力，故近幾年出現越來越多合夥用車的服務平台。但這些平台都有著根本上的一些限制，包括使用者要依據相同目的地分群、且用戶需要有駕照或者車子。這些限制使得某些用戶進入規劃系統後，沒有辦法被分配到已存在的合夥用車群集內，形成孤立用戶，必須自行開車。故在此篇論文中，我們針對未分到群集內的孤立用戶，提出一個進一步處理的方法，盡可能的讓所有使用者都能夠受益。作法是透過幾何上的費馬點問題，規劃出中繼點，並利用自訂的配對成本函式找出一個中繼群，中繼群將孤立用戶先載到此中繼點，再利用一個有相同目的地的目標群將此孤立點載到終點。透過效能分析跟實驗模擬，我們證明我們的方法能夠進一步的降低總運輸成本和車輛使用數量。最後，我們將此實作成一個 Web 服務系統，在此系統中，使用者可以依照喜好輸入起終點和出席日。系統接收到需求後，會到已建立的使用者和群集資料庫內搜尋是否有適合此用戶的合夥用車群集，並將使用旅行商問題演算法規劃後的路徑回傳給此用戶。

關鍵字：合夥用車、旅行商問題、費馬點、Web 服務系統

A Multi-Pool Transferring Carpooling System

Student: Hsuan-Han Peng

Advisor: Dr. Chien Chen

Institute of Network Engineering

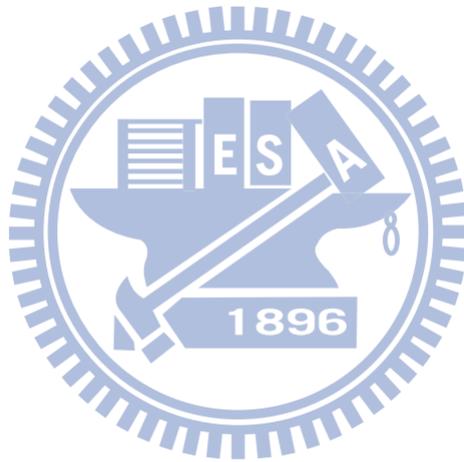
National Chiao Tung University

Abstract

Daily commute has become an integral part of our lives. Due to convenience and time saving, driving a personal car is the most common way people use to commute. Consequently, the number of vehicles rapidly increases, which create severe traffic congestion problem in major cities around the world. Public transportation system, however can ease the pain for the drivers stuck in the traffic. Nevertheless, it's not as timing, flexible and comfort as one's own car. Carpooling is an innovative transport services which reduce the number of vehicle usage by sharing use of a private car with other passengers. Carpooling not only has variety routes to choose for users, but also can relieve severe traffic congestion. As a result, more and more carpooling service platforms available recently. However, carpooling has some fundamental constraints, such as limit number of passengers, and carpooling cost function. These constraints cause some users could not join any carpooling group and need to drive alone. In this paper, we focus on these isolate users and try to design a further scheduling algorithm to reduce the number of isolate users. We propose that an isolate driver could moves to a destination by joining a carpool to a preselected transfer point, then taking another carpool to the destination. In other words, an isolate driver moves to destination by multi-pool transferring. We find a transfer point by mapping real world coordinate points to Fermat point problem, and find a transfer carpool and a target carpool by our cost function. We prove that our algorithm can reduce total traveling cost and the number of vehicles used through simulation. Finally, we implement our method into a Web-based carpool service system. Users can send their requests including their sources, destinations and schedules though our web interface. Our carpool system will check database to find a carpool, and return the routes

which are scheduled by well-known travelling salesman algorithm to user.

keywords: carpooling, travelling salesman problem, Fermat point, Web service system



誌謝

本篇論文的完成，我要感謝這兩年給予我協助和鼓勵的人。首先要感謝我的指導教授 陳健博士，雖然在這段期間，在研究上常常遇到挫折，但陳老師對我的指導跟教誨，都讓我在困難時找到解決方法，得以順利完成此篇文章。在其他方面，老師也給了許多不同的想法和概念，讓我有許多對人生的新想法，成長了很多，在此表達最誠摯的感謝。同時也感謝我的口試委員，王國禎教授、易志偉教授、朱煜煌博士，在口試時提出了許多寶貴意見，讓我受益良多。

感謝跟我一起努力的兩位同學，王柏翔和鄭元碩。不僅在修課階段大家彼此幫助，研究時若遇到瓶頸，常常也是三個人討論後，找到一些新的想法，讓彼此所作的研究更加完善。也要感謝實驗室的學長學弟們，陳盈羽、張哲維、黃鼎峰、孫冠羽、莊敬中、張大鈞等人，謝謝他們陪我度過兩年辛苦的研究生活，在我需要時不吝伸出援手幫忙。

特別感謝我的朋友，螢火蟲工作室的蟲兒們。每當研究累了，總是可以找大家出來聊個天，掃除心中的不愉快，甚至於得到一些新想法。這些好朋友們在我的研究生涯中，是一群不可或缺的推手，鼓勵並支持我一步一步的完成碩士學程。

最後，我要感謝家人對我的關懷與支持，父母含辛茹苦的栽培，讓我能夠無後顧之憂的專心於研究所的課程；還有姊姊在論文翻譯上的幫忙，在此要向他們致上最高的感謝。

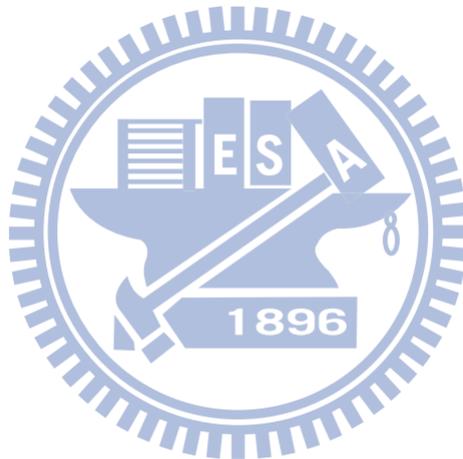
本文目錄

中文摘要.....	iii
Abstract.....	iv
誌謝.....	vi
本文目錄.....	vii
圖目錄.....	viii
表目錄.....	x
第一章 簡介.....	1
第二章 相關研究.....	4
第一節 合夥用車分群	5
第二節 合夥用車路徑規劃	6
第三節 合夥用車的公平性	6
第三章 多群轉乘的合夥用車系統.....	8
第一節 基本想法	8
第二節 問題描述	8
第三節 實際作法	10
第四章 系統架構.....	17
第一節 系統架構分析	17
第二節 實際操作介面	17
第五章 實驗結果.....	25
第六章 結論與未來展望.....	33

圖目錄

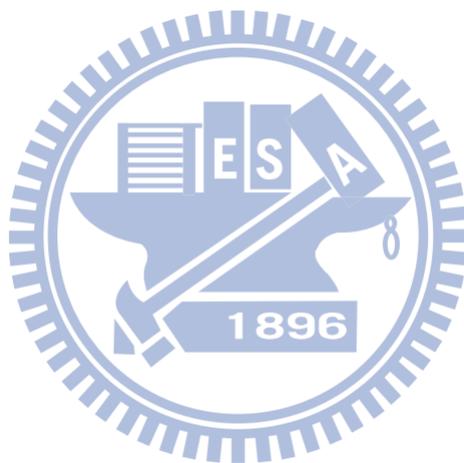
圖 一-1 桃北北宜基共乘網首頁.....	2
圖 一-2 TEXAS A&M UNIVERSITY CARPOOL 首頁.....	2
圖 一-3 megacarpool 首頁.....	2
圖 二-1 節約函數圖例.....	5
圖 二-2 合夥用車使用旅行商問題概念作路徑規劃圖例.....	6
圖 三-1 孤立點位置對節約函數的影響.....	9
圖 三-2 目標群、中繼點、中繼群、孤立點的關係.....	9
圖 三-3 傳統合夥用車移動情形.....	11
圖 三-4 中繼群乘載上限未滿的情況.....	12
圖 三-5 中繼群乘載上限已滿的情況.....	12
圖 三-6 加入中繼點後的路線變化.....	14
圖 三-7 三角形的費馬點尋找法.....	14
圖 三-8 凸、凹四邊形的費馬點.....	15
圖 三-9 實際地圖座標上尋找中繼點狀況.....	15
圖 四-1 系統架構.....	17
圖 四-2 網站介面.....	18
圖 四-3 網站介面 (衛星圖).....	18
圖 四-4 網站介面 (地形圖).....	19
圖 四-5 輸入欄位.....	19
圖 四-6 輸入起點後地圖上 (右下角) 出現定位圖標.....	20
圖 四-7 地圖上按右鍵設定起終點功能.....	20
圖 四-8 地圖上按右鍵設定起終點結果.....	20
圖 四-9 使用者輸入查詢資訊.....	21
圖 四-10 規劃結果 (使用者被規劃至已存在的單一群集).....	22
圖 四-11 使用者輸入查詢資訊.....	22
圖 四-12 規劃結果 (找不到適合此用戶的群集).....	23
圖 四-13 使用者輸入查詢資訊.....	23
圖 四-14 規劃結果 (使用者為孤立點，被中繼群跟目標群吸收).....	24
圖 五-1 使用者自行開車跟傳統合夥用車總移動距離比較.....	25
圖 五-2 使用者自行開車跟傳統合夥用車使用車輛比較.....	26
圖 五-3 傳統合夥用車和多群轉乘合夥用車總移動距離比較.....	26
圖 五-4 傳統合夥用車和多群轉乘合夥用車使用車輛數比較.....	27
圖 五-5 傳統合夥用車和多群轉乘合夥用車孤立點數量比較.....	27
圖 五-6 傳統合夥用車和多群合夥用車受影響群集總移動距離改善比例.....	27
圖 五-7 一百位用戶時傳統合夥用車和多群轉乘合夥用車總移動距離比較.....	28
圖 五-8 一百位用戶時傳統合夥用車和多群轉乘合夥用車使用車輛數比較.....	28

圖 五-9	一百位用戶時傳統合夥用車和多群轉乘合夥用車孤立點數量比較.....	29
圖 五-10	一百位用戶時傳統合夥用車和多群轉乘合夥用車受影響群集總移動距離改善比例.....	29
圖 五-11	不限制出席時傳統合夥用車和多群轉乘合夥用車總移動距離比較.....	30
圖 五-12	不限制出席時傳統合夥用車和多群轉乘合夥用車使用車輛數比較.....	30
圖 五-13	不限制出席時傳統合夥用車和多群轉乘合夥用車孤立點數量比較.....	30
圖 五-14	不限制出席時傳統合夥用車和多群轉乘合夥用車受影響群集總移動距離改善比例.....	31
圖 五-15	目的地數量五時傳統合夥用車和多群轉乘合夥用車總移動距離比較.....	31
圖 五-16	目的地數量五時傳統合夥用車和多群轉乘合夥用車使用車輛數比較.....	32
圖 五-17	目的地數量五時傳統合夥用車和多群轉乘合夥用車孤立點數量比較.....	32
圖 五-18	目的地數量五時傳統合夥用車和多群轉乘合夥用車受影響群集總移動距離改善比例.....	32



表目錄

表 二-1 決定駕駛用表格.....	7
表 三-1 假孤立點特定行程日被其他群集吸收後成為真孤立點.....	11



第一章 簡介

交通運輸和上下班通勤已經成為我們日常生活中不可或缺的一部份。個人駕駛私人用車，由於時間容易掌握，且路線可由自己決定，是最常被人們使用的移動方式。不僅如此，全世界車輛的擁有率依隨著人們的所得成長不停增加。根據西元兩千年的研究，全世界車輛的總數已超過了七億四千萬，這個現象大幅提高了交通問題的嚴重性。交通問題可能會引發道路壅塞、浪費車輛使用者的時間。對個人造成壓力、干擾，引發心血管疾病、聽力喪失等健康上的影響。車輛燃燒化石燃料所排放出來的二氧化碳和其他溫室氣體更進一步引起全球暖化、大氣污染。它引發的環境和社會成本太高，所以迫切需要一個有效的方法，在不影響便利性的前提下，來解決交通管理問題。

大眾運輸系統是一種解決的手段，大部分私人用車所造成的問題，像是通勤時間交通壅塞、車輛過多造成二氧化碳排放量過高等，都能夠有效的改善。而且採用大眾運輸系統，人們可以省下大量的車輛保養維修費，不需要花費大量時間找尋停車位，政府也不需使用太多空地來規劃停車空間。但不幸的是，它並無法提供多樣化的路線自由和彈性，也沒辦法像私用車一樣舒服，甚至會有誤點的可能。

合夥用車是一個創新的傳輸概念，和個人駕駛私人用車以及大眾運輸系統相比，合夥用車著重在增加乘載的彈性，提高車輛使用的效率，不只能夠減低車輛的使用數目，還能替人們節省開車的金錢消耗，例如油錢或通過收費站的支出，最重要的是，提供了大眾運輸沒有的路線跟時間彈性。它帶來了很大的利益。主要的作法是駕駛和多個乘客分享車子，通常使用於上下班。合夥用車往往是建立在用戶已有車輛的基礎上，參與者依照個人選擇條件，例如起終點、時間、日期等分群後，再使用群組內成員的車輛進行合夥用車。研究人員的測試結果顯示，合夥用車的作法除了大幅度的降低整體運輸成本，也能夠有效的解決交通管理問題。故近幾年國內外有越來越多的業者開始提供此類服務，例如：圖 一-1 的桃北北宜基共乘網 [1]、圖 一-2 的 Texas A&M university carpool [2]、圖 一-3 的 megacarpool [3] 等，用戶進入網站後，可以填寫資料成為會員，接著依照不同的目的地和出席日，系統會找出適合的合夥用車群集資訊並讓使用者確認。為了提倡合夥用車，某些國家的政府和公司也配合實施了一些提倡合夥用車的規定。例如：在一些大城市及周邊地區的高速公路上，常常會有 high occupancy vehicle 車道的縮寫



圖 一-1 桃北北宜基共乘網首頁



圖 一-2 TEXAS A&M UNIVERSITY CARPOOL 首頁



圖 一-3 megacarpool 首頁

HOV，也就是“高承載車”，這是交通管理當局為鼓勵人們合用汽車而專設的車道。或有些公司會規定，除非員工組成“合車組”，不然不可使用公司的停車位。

但合夥用車有著根本上的限制，譬如，當一位用戶周圍都沒有想跟他去相同目的地的人；或者，當所有跟該用戶前往相同目的地的人們，都已經找到自己的夥伴，而且他們的車子都已經坐滿，無法再容納其他人時，這個可憐的用戶最終還是只能開著自己的車出門，就如同系統沒有規劃過一樣。再讓我們想想合夥用車最基本的要素：須提供私人車輛跟駕駛，如果有一個沒有駕照，或是沒有買車的使用者，也想要節省支出時，上述的系統可能會不受理，因為他們連最基本的條件都不符合。我們發現，這些問題是因為傳統合夥用車規劃出來的結果，各個群體之間是彼此不相干的。這會導致某些參與此系統的用戶，依舊沒有辦法得到良好的規劃。

故本論文的目的是建立起群體間的關係，從原本獨立的群體，改變成多群一起規劃，藉此得到更好的結果。主要作法是找出初步規劃後無法被分配到群集的孤立點，利用統計各群集的出席狀況，試圖讓多個群集將孤立點的參與日分攤掉。若上個步驟結束後，依舊存在孤立點，則在符合車輛乘載的限制下，利用實際地圖的道路資料和座標點，對應到幾何上的費馬點問題後，找出合適的中繼點，並規劃出中繼群，將孤立點接送到中繼點，再讓擁有相同終點的目標群將它送到目的地，採取多個群之間做轉乘的方式，來降低整體的運輸成本。我們除了模擬測試這個想法的效能外，並將此實作成一個提高車輛使用率的規劃系統，盡可能的讓所有用戶都能夠受益。本系統鑑於現行路徑規劃網站(例如 google maps, Urmapp) 的運作方式，強調互動性和整合應用，採用 Web 2.0 的網站介面，系統主要使用一般大眾接受度高的 Google Maps API 建構，利用此工具不只可以得到 Google 提供的地圖資料，還有許多方便的功能可以使用。我們將它跟建立好的資料庫作整合應用，以提供用戶所需資訊。

本篇論文接下來的部分：第二章會介紹合夥用車問題的相關研究；第三章將會提出我們的多群轉乘合夥用車演算法；第四章則簡述我們系統的架構；第五章將經由模擬比較我們的方法以及傳統方法的效能；最後的第六章是對於此篇文章的結論以及展望。

第二章 相關研究

合夥用車的概念起源於 1970 年代中期，概念是兩個以上的用戶，當他們想去的目的地，在一條半共同路線上（此路線會經過起點跟終點）時，利用共享一台私人車輛達到目的。這個問題一般產生的原因是，許多的車輛在同一時間匯集到同一地點（生產設施，機場，體育中心等等），這種路線我們稱為半共同，因為就算終點相同，但每個人可能都從不同地方出發。通常來說，每個使用者從起點到終點可選擇的路徑都不同，只要是獨自移動，且沒有使用大眾運輸工具，路線上很難會有交集。

而合夥用車的任務是將想到達唯一目的地的使用者建立成一群集。在群集中的使用者根據「乘載容量」輪流提供車子，接著選擇一條路徑，此路徑會經過其他也在這個群集內的使用者所在地，再前往目的地。如果此群集是由五人組成，則對某個用戶來說，每五天就輪到他提供他的車（前往目的地）。

本篇文章所採用的合夥用車問題模組是由許多用戶組成，參考用戶所在位置和相同目的地分群，而用戶的需求是以最小的成本到達目的地。這取決於有多少車輛可用，以及它的覆蓋範圍。一般求解的演算法目的是，降低總體的運輸成本，這和已知的用戶群體資料有很大的關連性。

此研究的起源，是將目的地和路線建模成一個歐氏圖，由頂點跟邊的集合組成。為了簡單起見，我們將它都視為無向 (not oriented)；即對每條單一路徑來說，同時包含兩個方向。根據歐幾里得的定義，這個圖是 completely connected (or simply complete)，且符合三角定律。每個頂點都可從其他頂點過來；一邊連接兩個頂點時代表，這兩頂點中至少有一路徑存在。最後有一個假設是，我們能夠將一個真實存在的例子，轉成等價，且符合上述所說的歐氏幾何完全連通圖。並利用 Floyd-Warshall 演算法 (複雜度： $O(n^3)$)，解決圖中所有頂點對的最短路徑問題 (Shortest Path Problem)。雖然合夥用車的概念起源很早，但提出方法來解決此問題的論文產量並不多。本文主要參考一篇指標性的文章 [4]，整體架構的核心是先分群後規劃路線。首先，列出所有用戶的所在地跟目的地，跟道路資訊做結合後，轉成等價的歐氏幾何完全連通圖。接著我們分成兩個階段個別說明：

第一節 合夥用車分群

首先先定義節約函數，考慮用戶 i 和 j ，一個合夥用車問題的節約函數定義成：

$$S(i, j) = C(0, i) + C(0, j) - 2C(i, j)$$

其中， $C(i, j)$ 代表從用戶 i 移動到用戶 j 所需的成本， 0 則代表此兩點的共同目的地。這是一種在相同群集內，用戶配對後的優勢測量方法，只使用一台車前往它們共同的目的地，而不是分別使用自己的車。圖 二-1 顯示了兩個用戶的移動狀況和成本。 $S(i, j)$ 是一個測量的節約值，從兩次經過 0 點或從 0 點回去的旅程計算獲得。還有另一個整體節約成本，就是每天只使用一台車，而非兩台。因為這個關係，在圖 二-1 的第一部分，考慮兩個工作天的情況下，每點須要兩台車子作移動，而第二部份整體只需兩台車子。

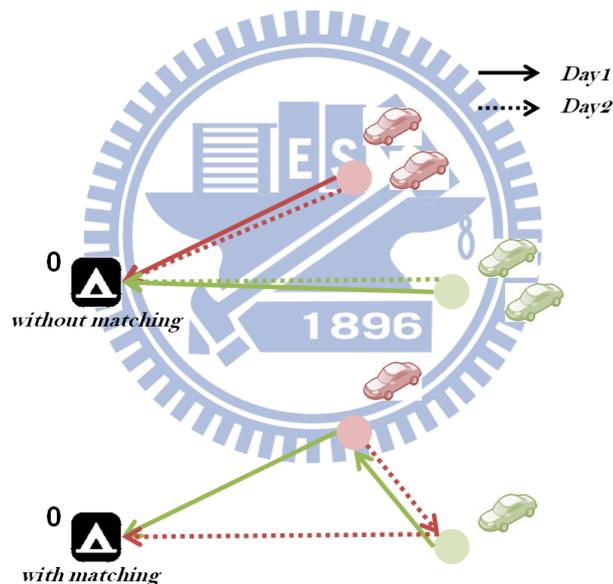


圖 二-1 節約函數圖例

合夥用車的分群階段，許多作法都是以貪婪為主，主要概念都是盡可能符合承載限制，快速的增加用戶來建立叢集。所有 $S(i, j)$ 值依照降序排列後，只要它的值大於零，即可生成連結。故當一個用戶新增到系統後，只要此用戶和候選群集內所有人的 $S(i, j)$ 值皆大於零，且滿足乘載限制，則可加入到此候選群集內，否則考慮下一個候選群；若皆不滿足則自己新增一群集。在我們的系統中，為了防止接乘客而繞路過遠的情況，加入了一個額外條件：若合夥用車所需時間跟原來行駛時間相比增加太多，則考慮下一個候選群。

第二節 合夥用車路徑規劃

著名的旅行商問題 (Travelling Salesman Problem)，目的是找出最短的漢彌爾頓迴圈 (Hamiltonian cycle)，可想成一個特殊的合夥用車例子。旅行商問題是一個多局部最優的最優化問題：有 n 個城市，一個推銷員要從其中某一個城市出發，唯一走遍所有的城市，再回到他出發的城市，求最短的路線。最明顯的演算法就是窮舉法，即尋找一切組合併取其最短。這種演算法的排列數為 $n!$ (其中 n 為節點個數)。用動態規劃技術，我們可以在 $O(n^2 2^N)$ 時間內解決此問題。雖然這仍然是指數級的，但要比 $O(n!)$ 快得多。對應到合夥用車上，我們知道，一個駕駛開車出門，經過每個使用者所在地後，再前往共同的目的地，最後是不需要回到路徑起點 (駕駛出發處) 的。所以我們可以將目的地到路徑起點的邊成本假定為零，然後利用動態規劃等方法，如同求旅行商問題的近似最佳解即可，如圖 二-2。

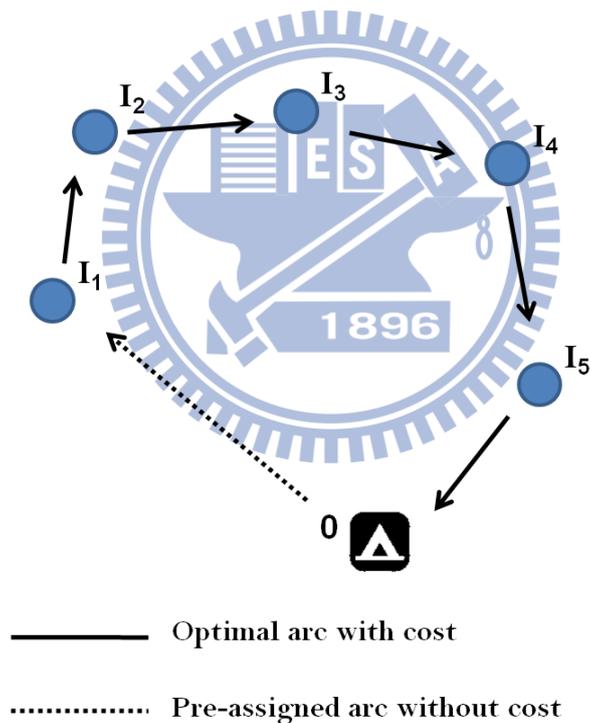


圖 二-2 合夥用車使用旅行商問題概念作路徑規劃圖例

第三節 合夥用車的公平性

由於合夥用車的運輸成本，會落到當日的駕駛身上，故直觀來說，我們需要一個公平的方式，每天決定不同的用戶當作駕駛，以平均整個群體的總運輸成本。一般想到的

方式是採用輪流的，比如使用姓名筆劃排列等，是最簡單的方案，也是最經常使用的方法。因此，若一個合夥用車群集內有 N 個成員，則對 i 這個人來說，每一次 N 天的週期，他需要在第 i 天當駕駛。這個方法的優點在於簡單易用，容易決定下一個駕駛是誰。但問題出在若指定的駕駛突然有事，就必須和其他成員交換順序。這種狀況發生多次之後，可能會難以判斷下一個該誰當駕駛。還有另一個問題是，若非駕駛的乘客有幾天未參與到，那是否還是要照著輪班表輪值？此乘客可能會感受到不公平，而取消此服務。

由於上述的問題，我們採用 [5] 這篇文章的想法來解決。首先定義一個值 U ，代表旅行的總成本。為方便起見， U 設為 $1, \dots, m$ 的最小公倍數， m 為同一群集中一日出席的最大人數。我們假設 m 為群集內的參與者總數 N (即 4)。所以 U 為 $1, \dots, 4$ 的最小公倍數 12。則可建出一個表格，包含了日期跟參與者，每一天則新增一列到裡面，而當天的開車者分數增加 $U(k-1)/k$ ，搭乘客則減 U/k ， k 為當天參與人數。選擇駕駛的依據一樣是輪流，以分數最低的為主。如表 二-1，起始所有用戶的分數皆為零，第一天，有 John, Phyllis, Ron 三人出席，因分數皆同，系統隨機挑選一人出來當作駕駛，當天駕駛所加的分數是 $12*(3-1)/3 = 8$ ，乘客則減 $12/3 = 4$ 。第二天四人都出席，其中 Phyllis, Ron 兩人分數相同，系統隨機挑選一人出來當作駕駛，當天駕駛所加的分數是 $12*(4-1)/4 = 9$ ，乘客則減 $12/4 = 3$ ，以此類推。

Date	Don	John	Phyllis	Ron
Origin	0	0	0	0
Day 1	0	8	-4	-4
Day 2	-3	5	-7	5
Day 3	-9	5	-1	5

表 二-1 決定駕駛用表格

因為 U 取了最小公倍數，所以每個欄位都為整數。用直覺的觀點來想這個演算法，駕駛的「成本」可以想成 U 個單位。而成本要照參與人數平均分配，所以若有 k 個參與者，每個人的成本為 U/k 。因此對每個乘客來說，都須「付給」 U/k 個單位給駕駛。

第三章 多群轉乘的合夥用車系統

第一節 基本想法

在前面的章節有提到，傳統的合夥用車概念，有一些根本上的限制，例如需要相同目的地的條件分群、車輛乘載的容量等，而且我們將傳統合夥用車的結構分析後發現，分群後規劃出來的結果，各個群體之間是彼此獨立不相干的。也意謂著，一開始雖然是全部的使用者進入系統，但分完群之後，每群合夥用車的路徑規劃是各自考慮計算，再加上 [4] 所提到的**節約函數**（一個判斷兩用戶若組成合夥用車是否會有助益的算式）額外的分群限制，可能會使**孤立點/用戶** (isolate node/user) 的數量上升。換句話說，這些限制會導致某些參與此系統的用戶，依舊沒有辦法得到良好的規劃。

為了解決上述所提到的問題，我們提出一個想法，希望可以在分群後得到的合夥用車群集集合間，增加群與群的處理，利用孤立點在群集之內換車轉乘的方式，讓他們也能夠參與合夥用車，進一步的降低總體的運輸成本，以及減少車輛使用量。

第二節 問題描述

我們發現孤立點的存在表示兩個原因：和相同目的地的其他群集作節約函數的結果對於總運輸距離皆無助益，或是孤立點的出席日當天相同目的地的其他群集皆已達到乘載上限。這兩個原因中，我們將針對前者做研究，而後者則視為系統的部分飽和，不做更進一步的處理。

分析節約函數後，可以發現孤立點和相同目的地的其他群集作節約函數結果對總運輸距離皆無助益這個問題，較大的起因是由於參與節約函數計算的兩位用戶所在地之間距離過遠，導致求出的值為負，也就是對總運輸成本的降低沒有助益。例如圖 三-1，假設用戶 A 和用戶 B 要前往相同目的地 D，B 從一開始的位置，和 A 作節約函數後，得到的值為 $50 + 50 - 70 * 2 = -40$ ，結果為負，表示 B 和 A 合夥用車是沒有助益的。但如果 B 能夠移動到星星圖示位置 C，此時節約函數計算結果的值為 $50 + 60 - 35 * 2 = 40$ 。表示 A 若和在位置 C 上的 B 組成一個合夥群集，便可以降低車輛使用數跟運輸成本。同樣的想法，若我們能讓孤立點移動到離相同目的地群集近一點的地方，得到節約函數為正，即有助益的結果，孤立點就可進一步被其他群集吸收，並改善總運輸成本。

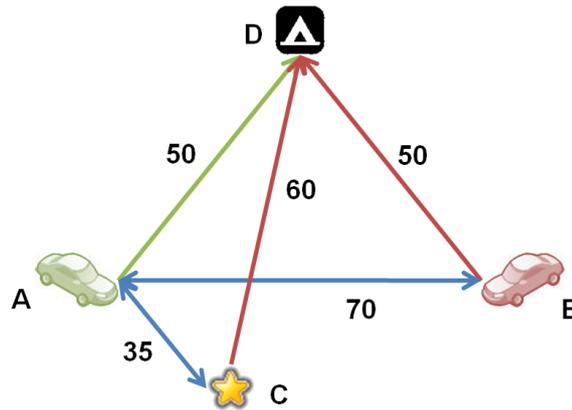


圖 三-1 孤立點位置對節約函數的影響

由於和孤立點有相同目的地的群集可能已經滿載或者距離過遠，若此時相同目的地的群集和此孤立點合夥用車，雖然使用的車輛數減少，但反而會增加總體運輸距離成本。因此我們提出利用中繼點的概念，並利用兩個合夥用車群集，讓孤立點不需自行開車，達到減少車輛使用，又能夠減少總體運輸距離的目標。如圖 三-2，對於一個孤立點，它分別從相同目的地跟相異目的地的群集中選出一個候選群，前者我們稱為**目標群** (target carpool)，即圖中綠色的群集；後者我們稱為**中繼群** (transfer carpool)，即圖中紅色的群集。我們的想法是針對一個孤立點，如圖中的小人，在它的出席日，系統會選擇並指派一目標群和一中繼群。孤立點先搭乘中繼群駕駛的車到系統指派的中繼點 (transfer point) 後下車，如圖中的藍色 P 圖示，再由目標群駕駛接送到目的地。

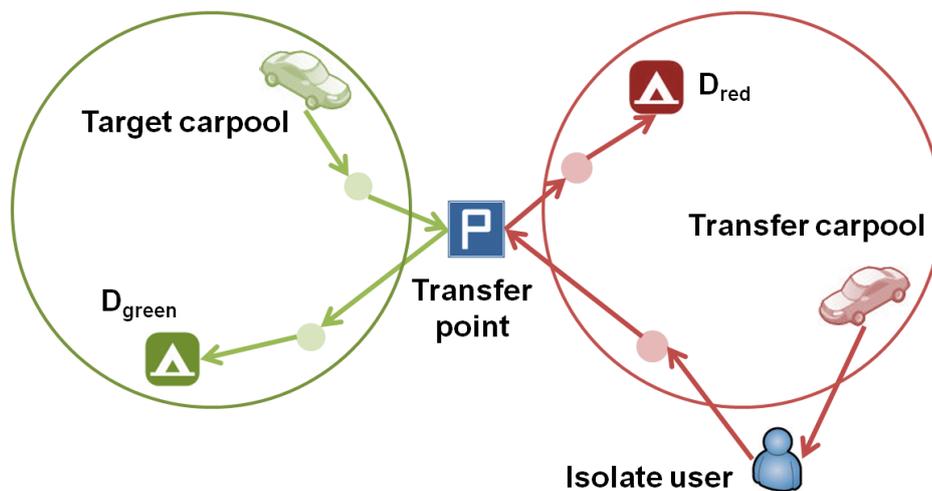


圖 三-2 目標群、中繼點、中繼群、孤立點的關係

需注意此處所選出的目標群，不可達到乘載上限，不然孤立點會無法上車。中繼群並沒有這個限制，若中繼群剛好達到乘載限制，只須先讓孤立點下車後，再將其餘參與者載上車前往目的地即可。現在我們的問題是，如何選擇一個目標群和一個中繼群以及中繼點，使得目標群和中繼群在增加了一個需要經過的新地點讓孤立點轉乘的狀況下，能夠吸收掉孤立點整段旅途的成本，讓總運輸成本仍可下降，更加提高合夥用車效益。

第三節 實際作法

我們的方法利用 [4], [5] 所描述的演算法為基礎，並利用 [4] 中所提到的節約函數，來處理分群的步驟。為了符合車輛乘載限制，我們假定一台小客車最多乘載人數為五人(含駕駛)，所以一個群集初步只可容許五人組成。換句話說，若參與此群集的人數到達上限，相同目的地的新用戶將不會考慮此群集為候選群。當分群完之後，我們會得到一個合夥用車群集的集合。因為上述條件，在此集合中，群集的大小最大為五，最小為一。此時將大小為一的群集稱為**假孤立點/用戶** (pseudo isolate node/user)。

接著將假孤立點從合夥用車群集的集合內取出，得到一個子集合。因為我們的系統會要求使用者輸入各自的出席日，並記錄在資料結構內，所以可以針對非孤立點，統計各群集每天的出席人數。若某個群集某天的出席人數不到乘載上限，則在節約函數大於零且目的地相同的前提下，這個群集可以吸收假孤立點成為當天的合夥用車用戶，如表三-1。在此表中，用戶 1, 3, 5, 8, 9 因為目的地相同，被分配至同一群集，即 1 號群集，而用戶 10 雖然跟他們有相同目的地，不過 1 號群集已滿，只能成為假孤立點。但當我們統計 1 號群集的出席狀況後，發現星期二跟四事實上是沒有達到乘載上限的，故這兩天用戶 10 可以加入 1 號群集，在特定日期出席。透過此步驟處理後，若依舊存在大小為一的群集時，我們將這些群集稱為**真孤立點/用戶** (real isolate node/user)。表三-1 的用戶 10 在星期一、三、五依舊找不到其他群集可以參與，故用戶 10 在這三天成為真孤立點。由於多群轉乘會增加額外繞路路徑，因此若不對假孤立點作判斷，而直接拿來考慮多群轉乘，可能導致整體運輸成本不減反增。

用戶 id	目的地	一	二	三	四	五	所屬群
1	NCTU	V	V	V	V	V	1
3	NCTU	V	V	V	V	V	1
5	NCTU	V		V	V	V	1
8	NCTU	V	V	V	V	V	1
9	NCTU	V	V	V		V	1
10	NCTU	V	V	V	V	V	3
人數統計		6	5	6	5	6	

表 三-1 假孤立點特定行程日被其他群集吸收後成為真孤立點

對於孤立點來說，接下來的問題是如何選出適合的目標群和中繼群。若挑到不好的組合，可能導致成本不減反增，因此我們制訂了一個配對影響值，用來決定每種組合的優劣，以方便每個孤立點選擇候選群。我們先假定同一群集裡面的使用者彼此之間的距離都相當短，以至於我們能夠將一個群集 I 縮小到內含所有使用者的二維座標重心 M_I (見圖 三-3,圖 三-4,圖 三-5)，得到簡化的示意圖，並利用這些圖來計算配對影響值。

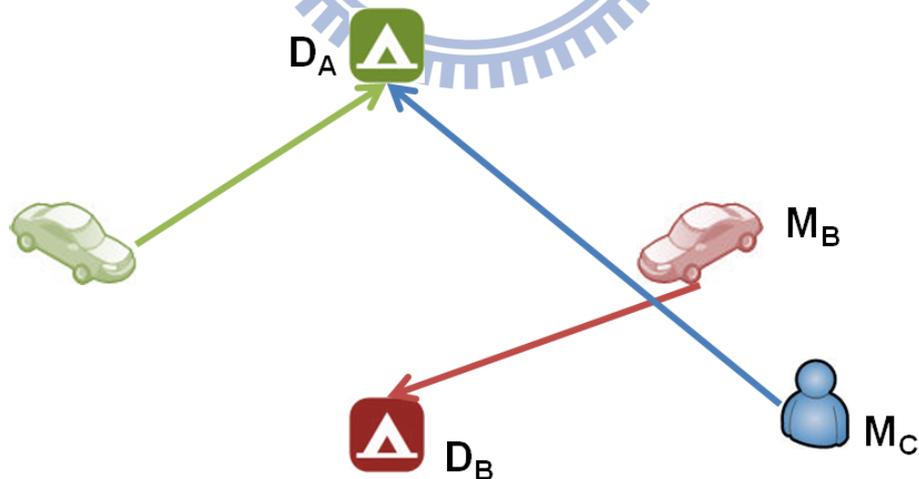


圖 三-3 傳統合夥用車移動情形

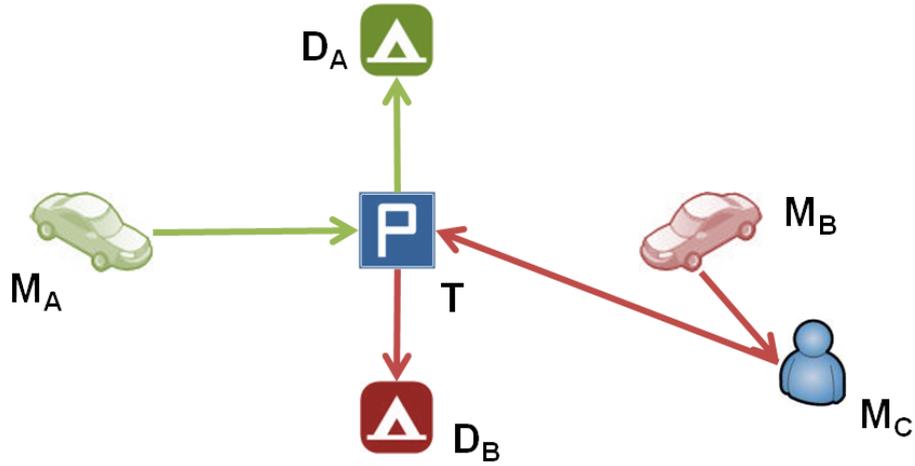


圖 三-4 中繼群乘載上限未滿的情況

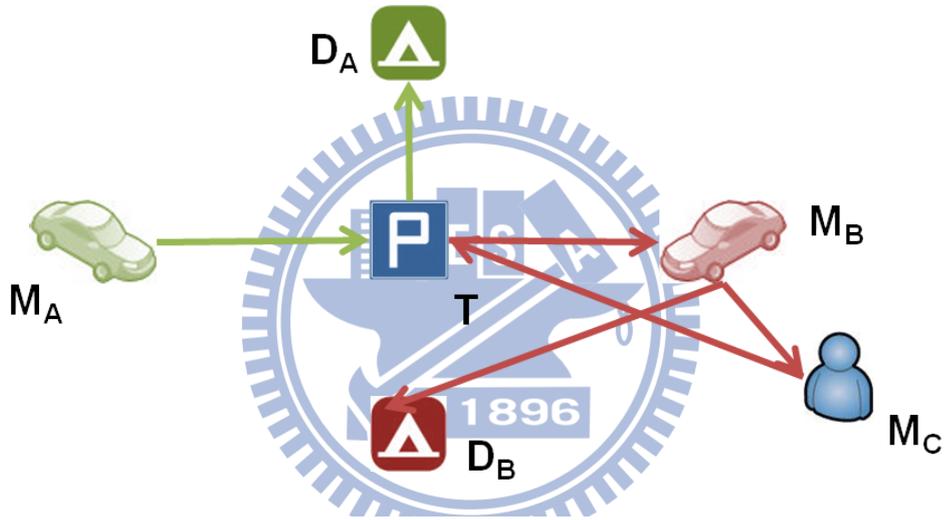


圖 三-5 中繼群乘載上限已滿的情況

若我們有一個目標群 i ，一個中繼群 j ，一個孤立點 k ，則我們假設原始的運輸成本為：

$$X(i, j, k) = \overline{M_i D_i} + \overline{M_j D_j} + \overline{M_k D_i} \quad (1)$$

而當我們考慮多群轉乘時，主要分為兩種情況。第一種是中繼群當天的出席人數未到達車輛乘載上限，表示中繼群就算加上孤立用戶，車子也能夠坐的下，因此不需要考慮特殊狀況，可以將孤立點囊括進來一起進行路徑規劃，則我們可以假設運輸成本為：

$$X(i, j, k) = \overline{M_i T} + \overline{T D_i} + \overline{M_j M_k} + \overline{M_k T} + \overline{T D_j} \quad (2)$$

其中 T 是當選到 (i, j) 這個目標群跟中繼群組合所求出的中繼點。

否則，中繼群須要先將孤立點送到中繼點，回頭把剩餘用戶接上車後，再前往目的地，此時運輸成本變成：

$$X(i, j, k) = \overline{M_i T} + \overline{T D_i} + \overline{M_j M_k} + \overline{M_k T} + \overline{T M_j} + \overline{M_j D_j} \quad (3)$$

如圖 三-3，一個傳統合夥用車分群的例子，它的運輸成本應為 $\overline{M_A D_A} + \overline{M_B D_B} + \overline{M_C D_A}$ 。當考慮多群轉乘時，如圖 三-4，中繼群未達到乘載上限，此時運輸成本可寫成 $\overline{M_A T} + \overline{T D_A} + \overline{M_B M_C} + \overline{M_C T} + \overline{T D_B}$ 。否則，如圖 三-5，運輸成本需寫成 $\overline{M_A T} + \overline{T D_A} + \overline{M_B M_C} + \overline{M_C T} + \overline{T M_B} + \overline{M_B D_B}$ 。

我們把後兩者減掉前者的值稱為**配對影響值**，顯而易見的，若配對影響值為負，表示 (A, B, C, T) 這個組合可以讓總體的運輸成本進一步下降，且由於孤立點被中繼群和目標群接送，可以減少一台車輛使用。當我們選到 (A, B) 組合時，需依據兩個群集的相對位置計算中繼點 T 的位置。由上述的算式可以發現， T 的位置會大幅影響到配對影響值，故我們的下一步，是研究如何找出一個點，能達到目標群和中繼群成本上升最少的目標。

從幾何的角度而言，以圖 三-6 左邊的部分為例，在不考慮中繼點時，存在兩個群集，左邊綠色的目標群跟右邊紅色的中繼群，圖中的節點皆為合夥用車用戶。假設某天此兩群集規劃出的路線分別為 $X \rightarrow A \rightarrow B$ 和 $Y \rightarrow C \rightarrow D$ 。當目標群和中繼群加入一中繼點並繞路後，則變成了圖 三-6 右邊部分的狀態，可以發現路線分別變為 $X \rightarrow A \rightarrow T \rightarrow B$ 以及 $Y \rightarrow C \rightarrow T \rightarrow D$ ，即原本 $\overline{AB} + \overline{CD}$ 所需的運輸成本變為 $\overline{AT} + \overline{TB} + \overline{CT} + \overline{TD}$ ，我們將 A, B, C, D 四個用戶稱為**參與繞路用戶**，而這個變化就是加入中繼點後目標群跟中繼群因繞路運輸成本增加的部分。直觀來說，要使這個值增加的越少，主要有兩個要素：一個是參與繞路的四個用戶距離遠近，另一個則是 T 的位置。針對前者，我們所做的處理是，在加入中繼點前先分別對目標群跟中繼群作一次合夥用車的路線規劃，得到當天所走的路線和用戶出席狀況，並利用這個資訊，從中繼群跟目標群各找出一位當天有出席，且這兩人之間距離為最小的配對，例如找到圖 三-6 的 A, C 兩點。找到此配對後，再分別使用這兩點和對方的前後用戶作距離判斷，意謂計算 $\overline{AY}, \overline{AD}$ 以及 $\overline{CX}, \overline{CB}$ ，並分別找出較小者，例如找到 \overline{AD} 和 \overline{CB} ，此時得到 A, B, C, D 四個參與繞路用

戶，同時將 \overline{AB} 和 \overline{CD} 分別當作為兩群集中最近的兩線段，利用從這四個位置作繞路，讓增加的運輸成本能夠較小。

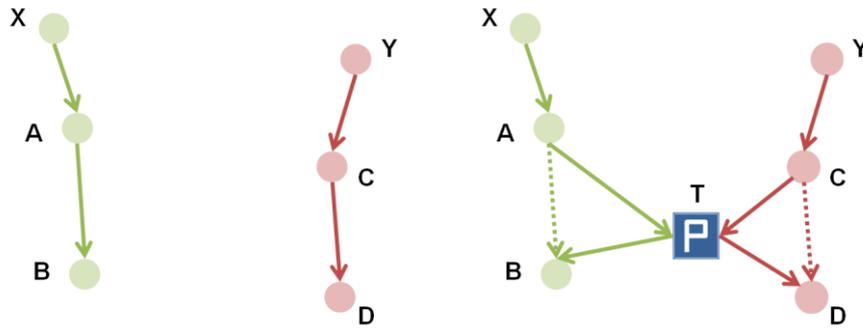


圖 三-6 加入中繼點後的路線變化

至於後者，意味著需要找出一個 T ，使得 $\overline{AT} + \overline{TB} + \overline{CT} + \overline{TD}$ 為最小。當 \overline{AB} 和 \overline{CD} 兩線段有交點時，取交點可以讓此值最小；當沒有交點時，我們將這個問題對應到幾何上的費馬點問題來作處理。在幾何學中，費馬點是位於三角形內的一個點，給定一個三角形 $\triangle ABC$ ，從這個三角形的費馬點 P 到三角形的三個頂點 A 、 B 、 C 的距離之和 $\overline{PA} + \overline{PB} + \overline{PC}$ 比從其它點算起的都要小。這個特殊點對於每個給定的三角形都只有一個。費馬點問題最早是由法國數學家費馬在一封寫給義大利數學家托里切利的信中提出的。托里切利最早解決了這個問題，而 19 世紀的數學家斯坦納重新發現了這個問題，並系統地進行了推廣，因此這個點也稱為托里切利點或斯坦納點。

如圖 三-7，三角形的費馬點找法是分別以每個邊往外作正三角形，然後將外部頂點和此邊對面的三角形頂點作連線，最後三線的交點即為費馬點。

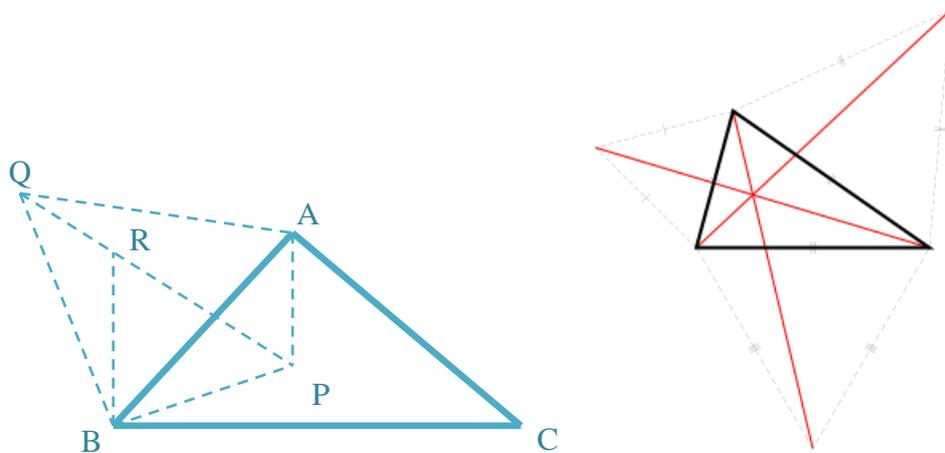


圖 三-7 三角形的費馬點尋找法

故中繼點 T 只要取四邊形的費馬點，即可讓此值相加後為最小。但傳統的費馬點只對於三角形作處理，我們利用 [6] 這篇文章內的證明，找出四邊形費馬點。如圖 三-8 所示，對於凸四邊形來說，費馬點是對角線交叉點 P；凹四邊形是該四邊形的凹點 D。

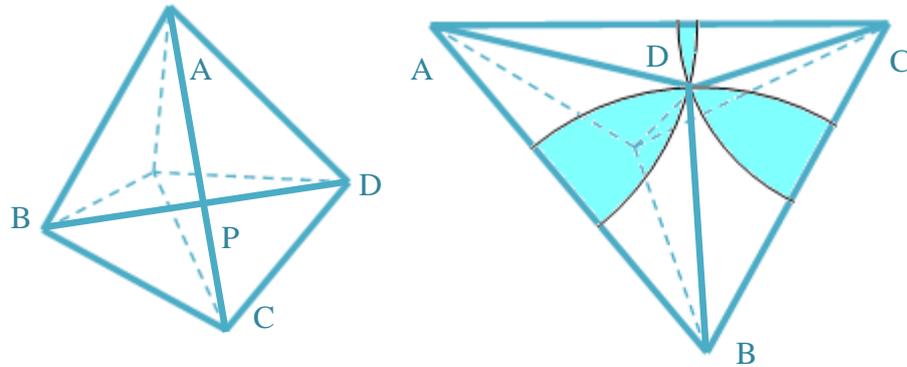


圖 三-8 凸、凹四邊形的費馬點

然而我們知道，在實際的情況下，車輛移動會受到道路的限制，兩點之間的路線並不一定為直線，所以要在地圖上找出最佳參與繞路節點，需要使用實際地圖圖資的交叉路口、道路等定位座標點，才能得到較精確的結果。但若把整條路徑上的圖資定位座標點都考慮進來，計算量會相當龐大，故我們利用上述方法，先找出目標群和中繼群中參與繞路用戶 A, B, C, D 的位置，如圖 三-9，以此降低需要考慮的座標點數量，再把使用者 A 點和 B 點的所在位置傳到 Google maps 作路徑規劃，C 點和 D 點亦同，最後利用回傳的道路定位座標，使用找出 A, B, C, D 的相同方法，找出地圖上真正最近的 A', B', C', D' 當作最佳參與繞路節點，再利用此四點找尋費馬點作為我們的中繼點。

找到中繼點後，將此點資訊提供給相關聯的目標群、中繼群、孤立點。全部的孤立點都處理結束後，利用各群資訊作合夥用車路徑規劃。並將處理結果回傳給使用者。

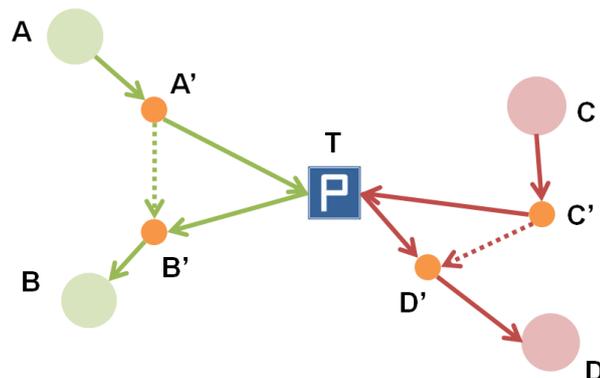
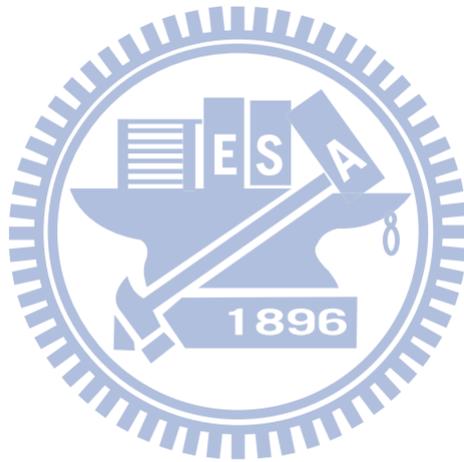


圖 三-9 實際地圖座標上尋找中繼點狀況

若對孤立點的群集作上述處理後，沒有辦法找到任何能夠提升效能的多群轉乘解時，我們的系統會採用原本傳統合夥用車的規劃結果，即傳統合夥用車的解是我們的最差情況。



第四章 系統架構

由於本系統希望能在現實生活實際應用，考慮到方便性以及普遍性，採用 Java 為我們開發的程式語言，使用 Java 撰寫的另一個好處是較容易跟 Java Servlet 和 JSP 等相關應用接軌，使得跟用戶端聯絡更為方便。我們規劃並設計了一個 Java Servlet Web Server，讓使用者可以以瀏覽器連至入口網站，透過地圖介面自行設定起終點跟出席狀況，並得到處理後的結果。

第一節 系統架構分析

我們的系統主要分為三層，分別為介面層、應用層、資料層。架構圖如圖 四-1 所示，介面層主要提供給使用者閱覽並使用，採取 Web 2.0 的網站介面，以一般大眾接受度高的 Google Maps API 建構。使用者輸入需求以後，資料會送到我們所架設 Apache 伺服器上的應用層，利用 Java Servlet 的方式在伺服器上進行多群轉乘合夥用車程序，規劃完後將結果回傳至用戶端顯示。為了畫面美觀跟容易撰寫，我們採用 JavaServer Pages (JSP) 來處理動態網頁的部分。資料層我們使用 PostgreSQL 這套軟體的 8.2 版建構，存放已登入系統內使用者的各項資訊，譬如起終點、出席天數。

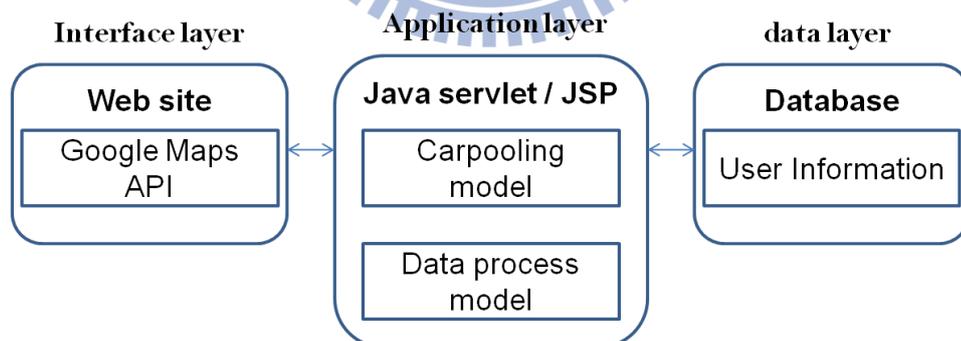


圖 四-1 系統架構

第二節 實際操作介面

我們主要使用 Google maps API 來設計介面，用戶只需要簡單的連上網頁，並輸入自己的需求，包括所在地、目的地、以及出席日，系統會自動連至資料庫，搜尋是否有合適的群集，並在規劃後，將每日行程的路線顯示在網頁上，讓使用者作確認。每個用

戶只看的到自己群集的規劃結果，整體資料管理則是由我們的伺服器負責。介面主要以高互動性為原則，使用者可以用多種方法輸入資訊，包括鍵盤輸入或是使用滑鼠點選地圖。系統會適時的給予使用者一些提醒或幫助，讓操作更加簡單。

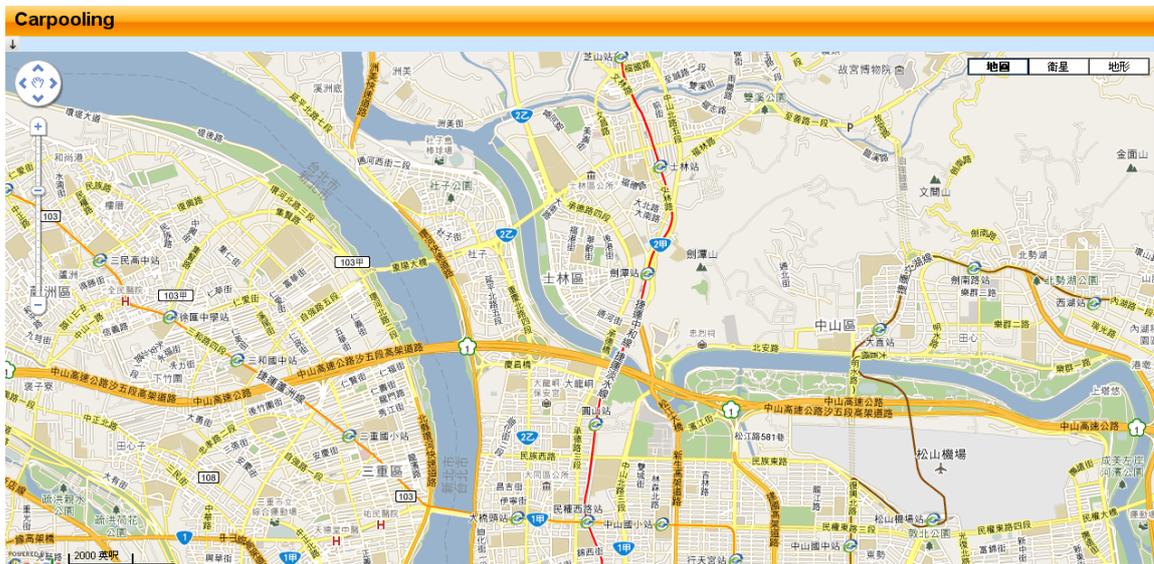


圖 四-2 網站介面

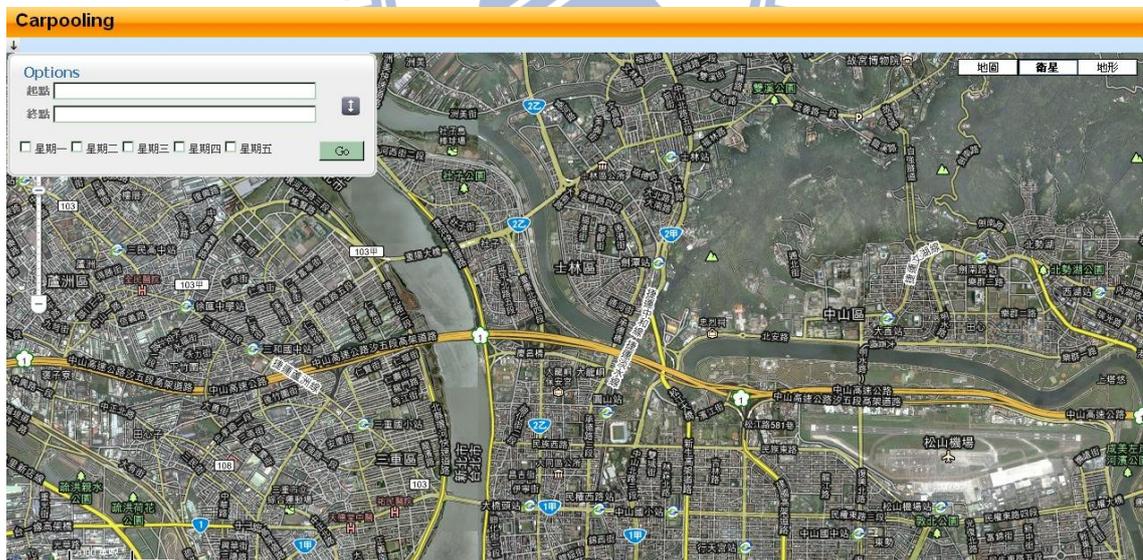


圖 四-3 網站介面（衛星圖）

圖 四-2 是我們的網站介面，採用一般大眾普遍接受度高的 Google maps 來架設此介面，利用 Google 提供的 API，可以輕易的取得地圖圖資。若使用者有需要，在右上角提供了衛星圖和地形圖的轉換，也可以使用滾輪或是畫面左邊的拉桿來調節地圖大小，

如圖 四-3,圖 四-4。

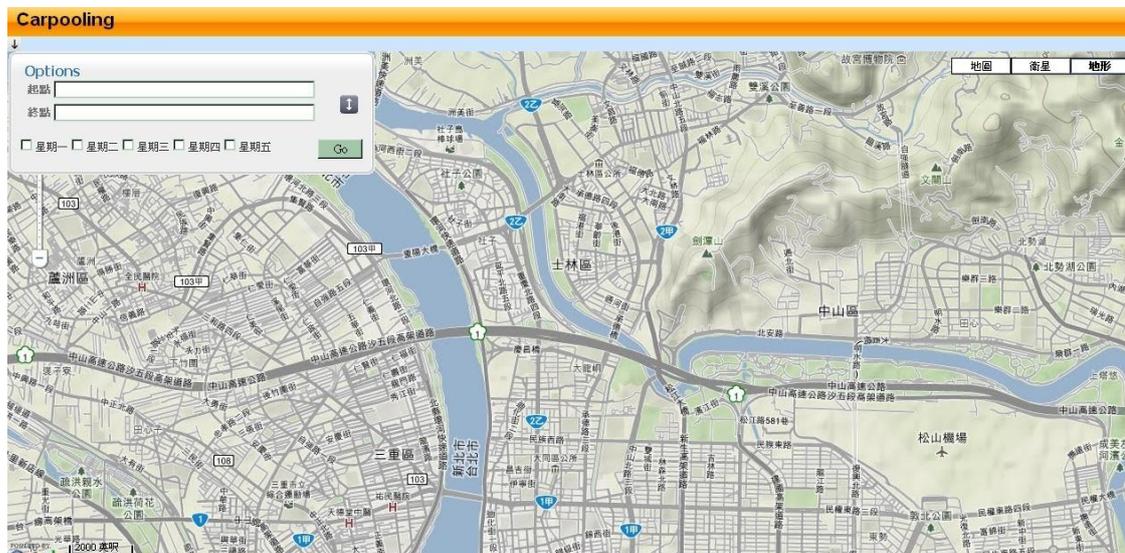


圖 四-4 網站介面 (地形圖)

對於一個想要加入合夥用車的使用者來說，只需在圖 四-5 中的欄位，輸入自己的起點跟想去的目的地以及出席日，系統會在地圖上出現定位圖標並移動讓使用者確認，如圖 四-6。當用戶確認資料無誤送出後，系統會進入資料庫尋找是否已存在適合此使用者的合夥用車群集，在符合車輛乘載限制下，檢查是否可讓此用戶加入。如圖 四-7，圖 四-8，我們的系統也提供了在地圖上按右鍵點選設定起終點的功能，讓使用者不需填寫繁複的地址，輕鬆的輸入需求。當點選右鍵設定起終點時，地圖上也同樣會出現定位圖標，且起終點欄位出現點選處的住址。

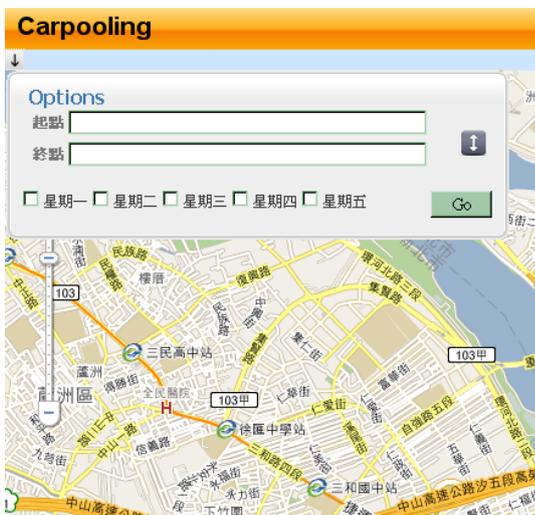


圖 四-5 輸入欄位

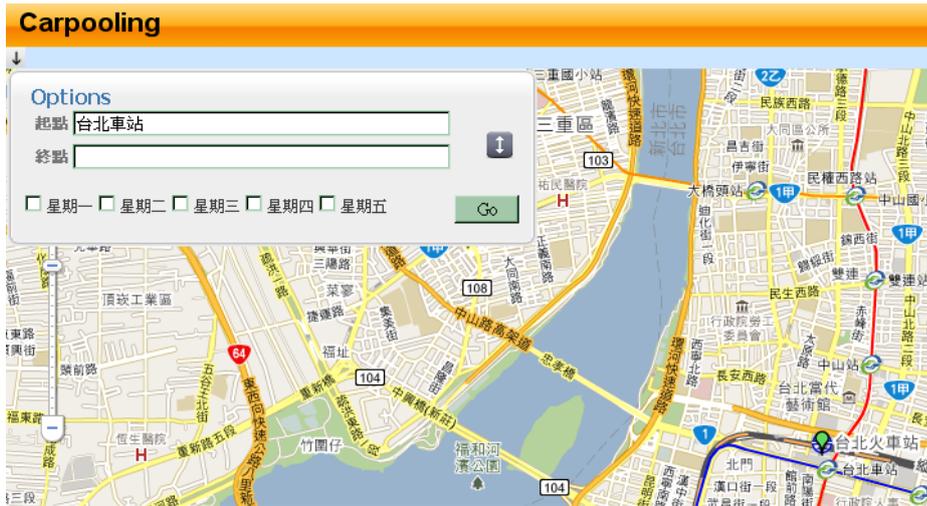


圖 四-6 輸入起點後地圖上（右下角）出現定位圖標

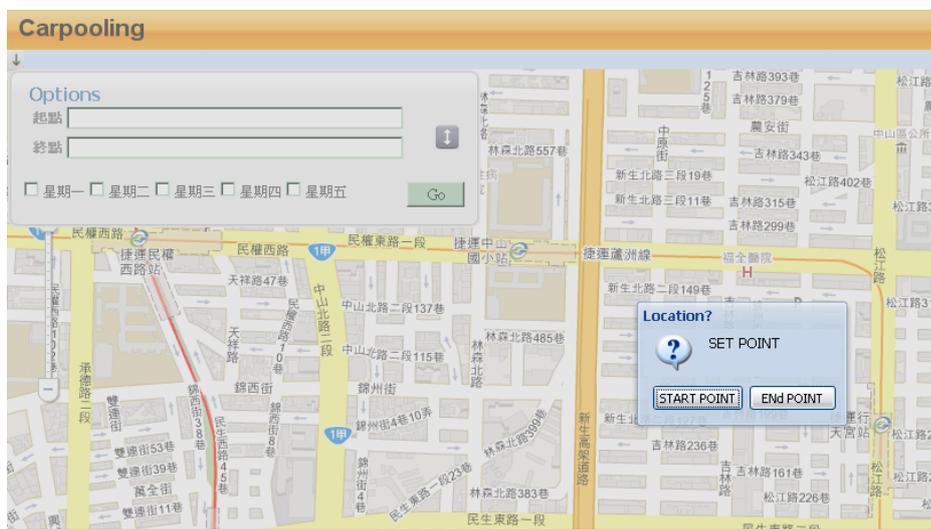


圖 四-7 地圖上按右鍵設定起終點功能

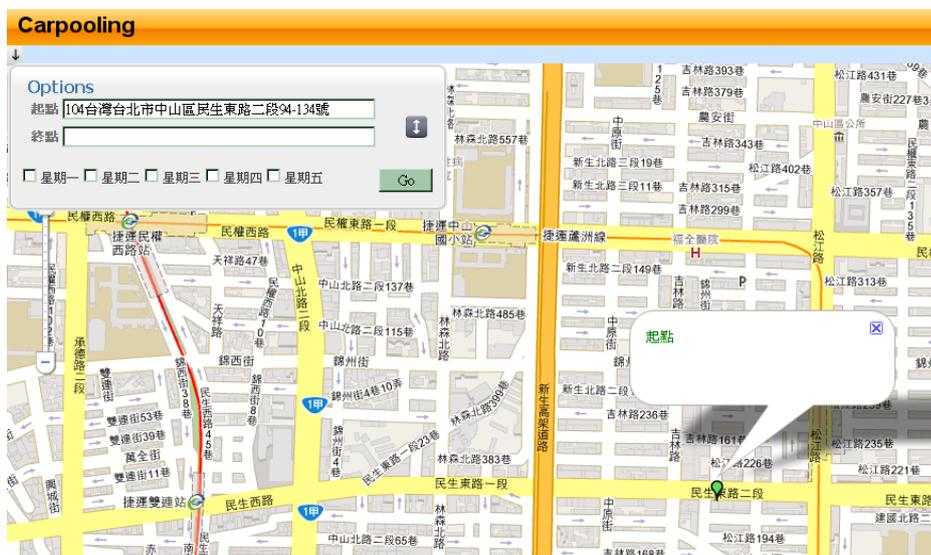


圖 四-8 地圖上按右鍵設定起終點結果

接著我們來看一些系統運作的實際例子，圖 四-9 是一位想去士林夜市的用戶，但因不清楚現在所在位置，所以起點的部分他在地圖上點選，終點輸入了士林夜市，我們可以發現圖上出現了綠色的定位圖標，表示了起終點，而出席日此用戶勾選了星期一、二、四。送出需求後，系統在資料庫內搜尋到一個適合此用戶的合夥用車群集，並將路徑回傳給該使用者。當天的行程是從車子圖樣處，為駕駛的 0 號用戶出發，並依照紅色路線，經過 1 號、2 號、3 號，然後到達房子圖示的使用者所在地接用戶上車後，最後前往終點（紅色驚嘆號處），總路程約花費 21 分鐘，如圖 四-10。由於合夥用車的路徑可能會有交錯，為了避免閱讀困難，當游標移至用戶圖標上時，畫面上會顯示出該用戶的編號，利用此功能和路徑說明對照，提供較明確的資訊讓使用者較容易了解。

下一個例子如圖 四-11，圖 四-12，一個在松山機場附近想前往台北車站的用戶，在輸入了個人資訊後，系統並無法找到適合他的合夥用車群集，也無法找到能夠降低總運輸距離的多群轉乘解，故系統顯示出起點到終點的路線，並請此使用者自行開車前往目的地。



圖 四-9 使用者輸入查詢資訊



圖 四-10 規劃結果（使用者被規劃至已存在的單一群集）

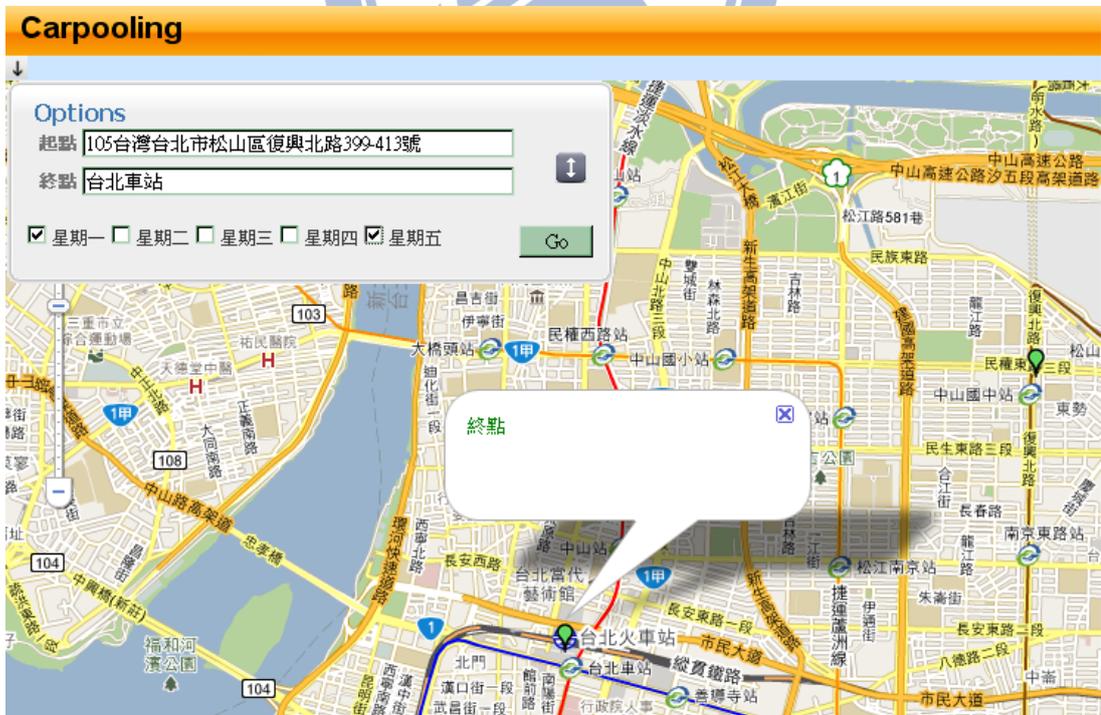


圖 四-11 使用者輸入查詢資訊

Carpooling

Options

起點 105台灣台北市松山區復興北路399-413號

終點 台北車站

星期一 星期二 星期三 星期四 星期五

Go

 105台灣台北市松山區復興北路399-413號

未分群至任何 carpool 或本日無其他用戶出席
請自行開車 5722.0 公尺到目的地
總行程約花費 5 分鐘



圖 四-12 規劃結果 (找不到適合此用戶的群集)

Carpooling

Options

起點 台北醫學大學

終點 士林夜市

星期一 星期二 星期三 星期四 星期五

Go

終點

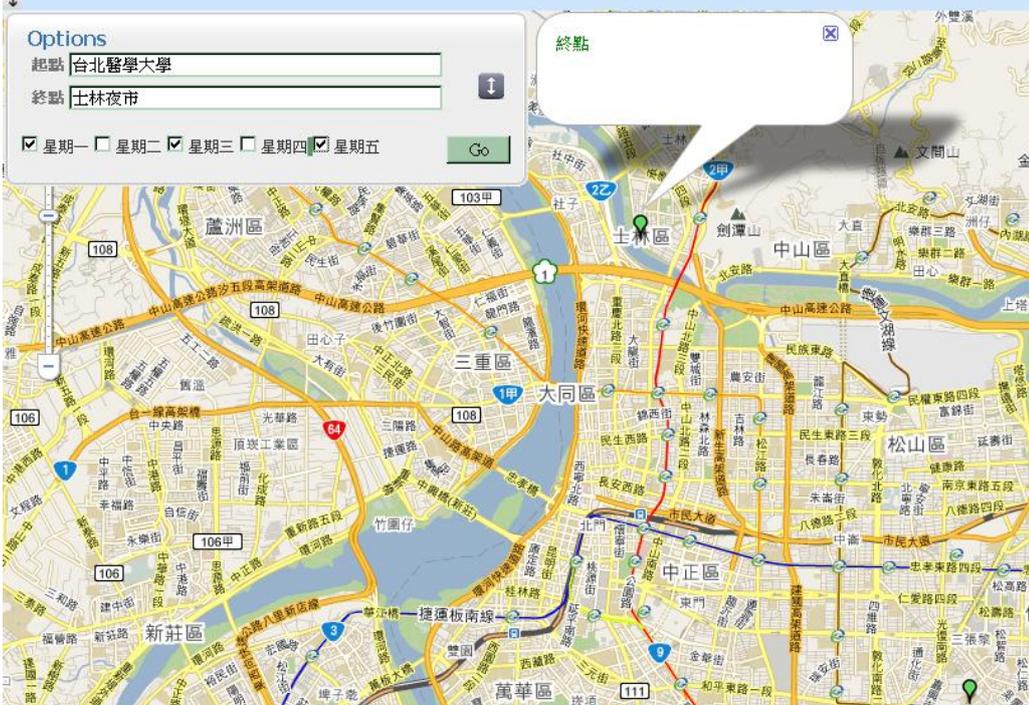


圖 四-13 使用者輸入查詢資訊

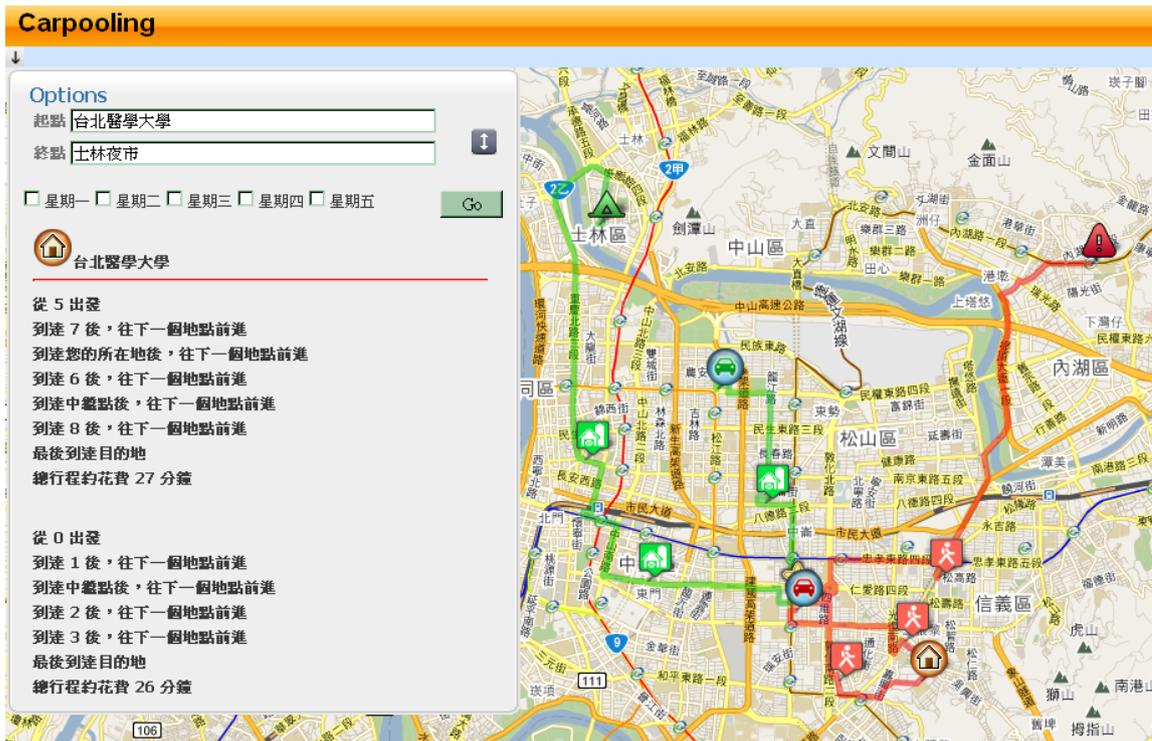


圖 四-14 規劃結果（使用者為孤立點，被中繼群跟目標群吸收）

最後一個是成功多群轉乘的例子。一位用戶從台北醫學大學出發，想要前往士林夜市，如圖 四-13。在他輸入需求後，系統沒有辦法在附近搜尋到一個目的地一樣為士林夜市的合夥用車群集，但卻可以找到降低總運輸距離的多群轉乘解。如圖 四-14，系統規劃的結果是：以紅色為代表的中繼群，行程從紅色車輛圖標處的 5 號用戶為駕駛出發，依照 7 號用戶、該使用者、6 號用戶、中繼點、8 號用戶然後前往目的地的順序作移動，總花費約 27 分鐘。目標群則以綠色為代表，從綠色車輛圖標處的 0 號用戶為駕駛，依照 1 號用戶、中繼點、2 號用戶、3 號用戶然後前往目的地的順序作移動，總花費約為 26 分，其中中繼點以星星圖示代表。系統規劃採取多群轉乘的方式，讓此用戶能夠順利到達目的地，左邊的說明欄位也清楚的描述兩個群的移動順序。

第五章 實驗結果

在此章中，我們針對不同的狀況，作合夥用車的模擬。採用的測試環境是 Java，結合 PostgreSQL 的資料庫。測試的地圖圖資主要是台北市松山區、中正區、萬華區、信義區，還有周邊新北市永和區、中和區、板橋區等等。

我們從這幾個區域內隨機選出使用者的目的地跟出席狀況，目的地則由系統提供的幾個熱門地點（包括台北車站、板橋高鐵站、松山機場、士林夜市等等）擇一。每次系統處理週期為一星期（一次處理五天），隨機測試十次。根據以前合夥用車的研究，大多數都著重在於比較總體運輸距離（成本）上，故我們也專注在這方面，再加上統計使用的車輛數目，以確定我們方法的效能。

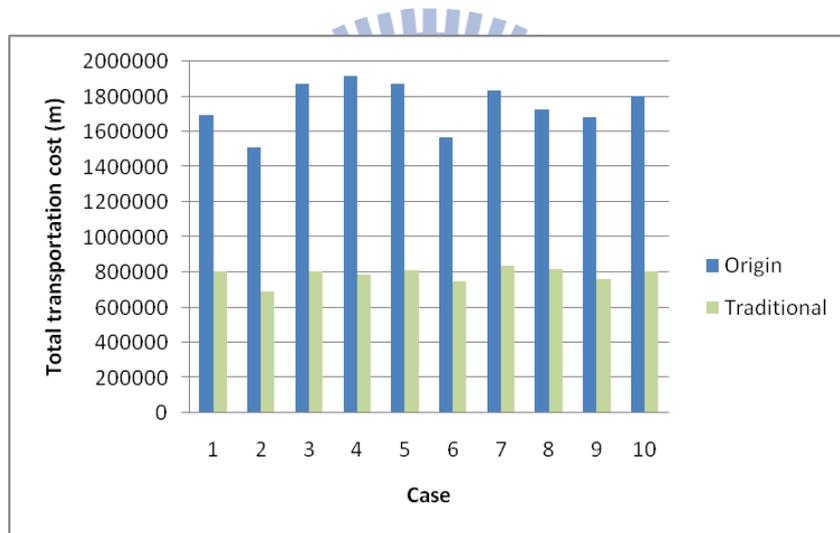


圖 五-1 使用者自行開車跟傳統合夥用車總移動距離比較

首先我們先測試有無合夥用車上的差異，我們假設有五十位隨機產生用戶，三個固定目的地，且所有用戶的出席狀況皆為三天以上（若無特別說明，後面測試都以此為對照組）。由圖 五-1 我們可以發現，使用者合夥用車之後，跟自行開車相比，所省下的整體運輸總距離約可到達一半。從圖 五-2 則可知道使用車輛數的差別，約可省下四分之三的使用車輛。總體得到的效益非常的高。也因為兩者結果差異過大，後面比較我們將不再和自行開車相比，只單純用我們的方法跟傳統合夥用車比較。

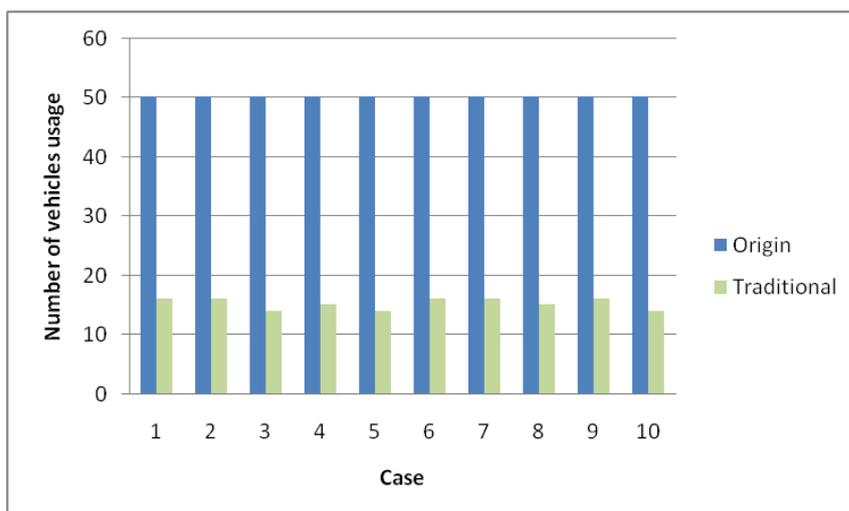


圖 五-2 使用者自行開車跟傳統合夥用車使用車輛比較

接著在同樣條件下，來比較合夥用車和我們的方法。如圖 五-3,圖 五-4 可發現，採用我們的方法後，可以讓總移動距離小幅度再減少，並減少車輛使用數目。由於我們的方法目的是在於減低總移動距離（運輸成本），所以傳統合夥用車的成本可以當作上限值，在最差的情況下，我們的規劃結果會與傳統方法相同。



圖 五-3 傳統合夥用車和多群轉乘合夥用車總移動距離比較

接著測試我們的方法，在不同情境跟條件下的表現。圖 五-7 至圖 五-10 是原用戶不動，再增加五十位隨機使用者，將用戶數提升到一百人的情形，圖 五-11 至圖 五-14 是不設定用戶出席限制的情況，圖 五-15 至圖 五-18 是將可選目的地加至五個的情況。

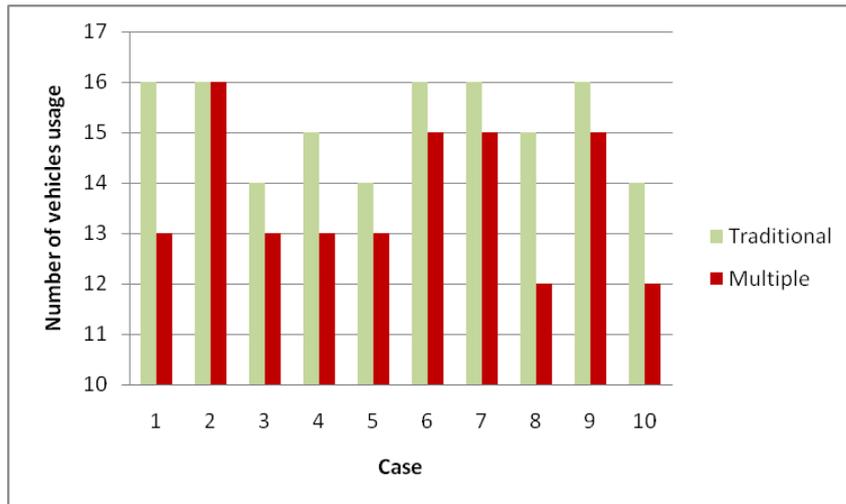


圖 五-4 傳統合夥用車和多群轉乘合夥用車使用車輛數比較

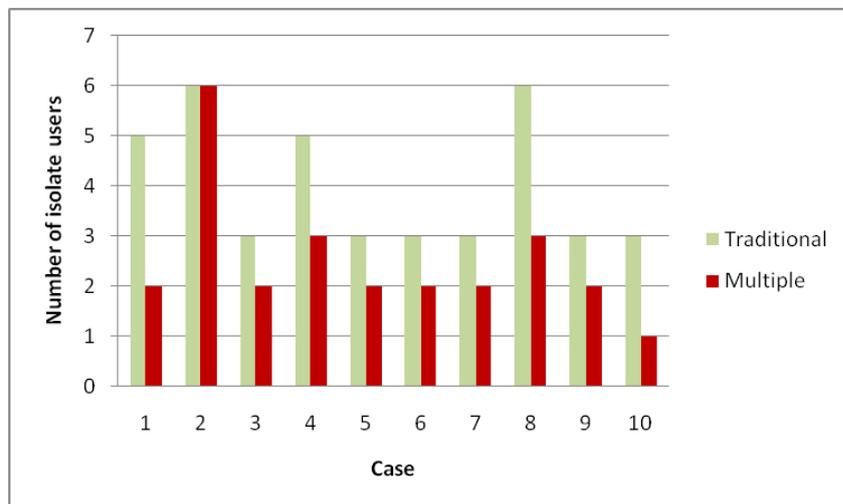


圖 五-5 傳統合夥用車和多群轉乘合夥用車孤立點數量比較

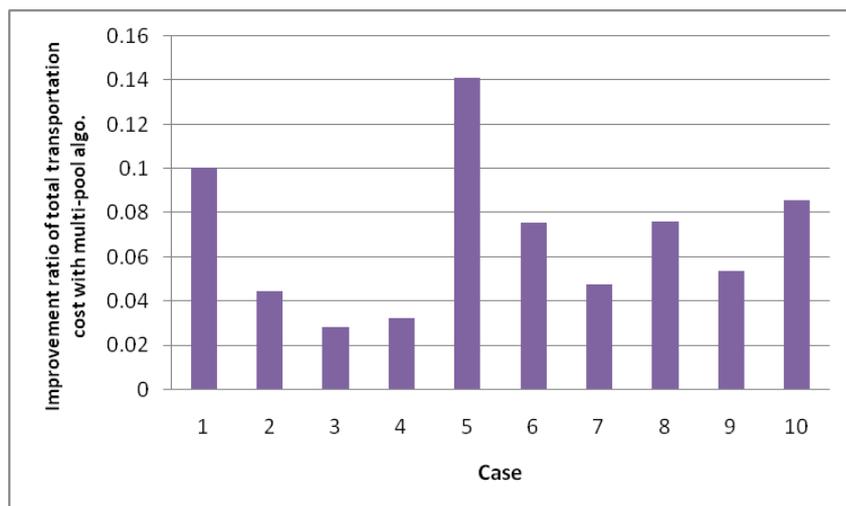


圖 五-6 傳統合夥用車和多群合夥用車受影響群集總移動距離改善比例

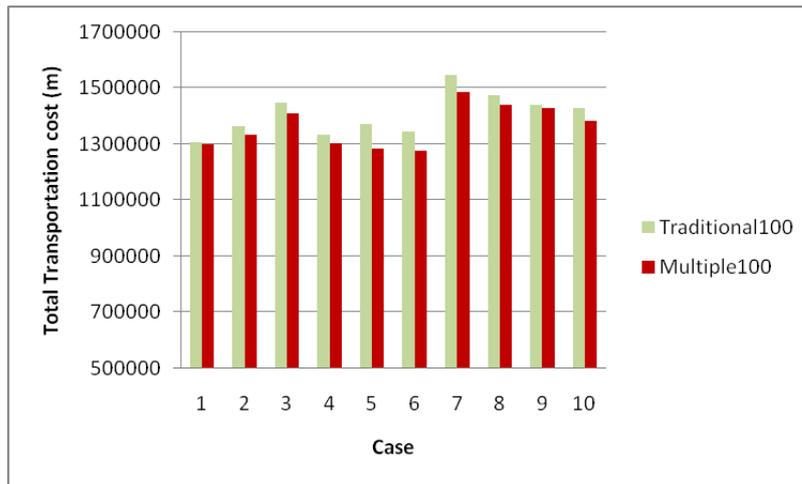


圖 五-7 一百位用戶時傳統合夥用車和多群轉乘合夥用車總移動距離比較

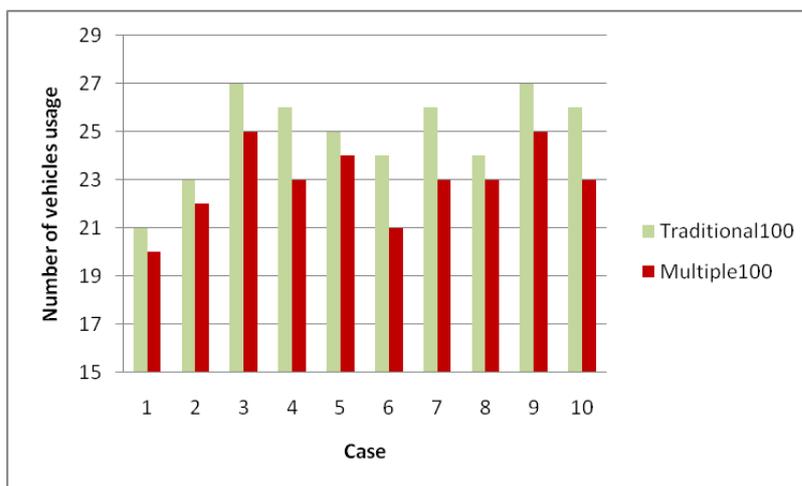


圖 五-8 一百位用戶時傳統合夥用車和多群轉乘合夥用車使用車輛數比較

我們觀察當總用戶數達到一百人時的狀況，如圖 五-7，明顯的十次測試中，車輛總移動距離皆上升。由於參與者變多，車輛的乘載上限不變，相對合夥用車的分群的數量會增加，使用的車輛數也增加，如圖 五-8。直觀來說，孤立點的數量也會上升，但由於群集變多，設定的目的地數量未增加，故新進用戶比較容易找到和它相同目的地的群集進行合夥用車；群集的數量增加，使得真孤立點變為假孤立點的機會也提高了。兩者消長之下，孤立點的數量沒有明顯改變，系統能夠進一步規劃的選擇較少，多群轉乘的效益增加較低，如圖 五-9, 圖 五-10。

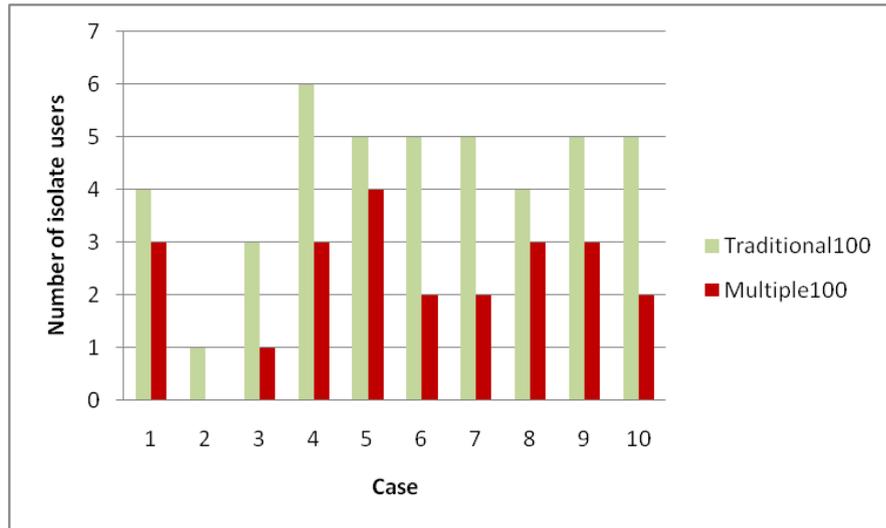


圖 五-9 一百位用戶時傳統合夥用車和多群轉乘合夥用車孤立點數量比較

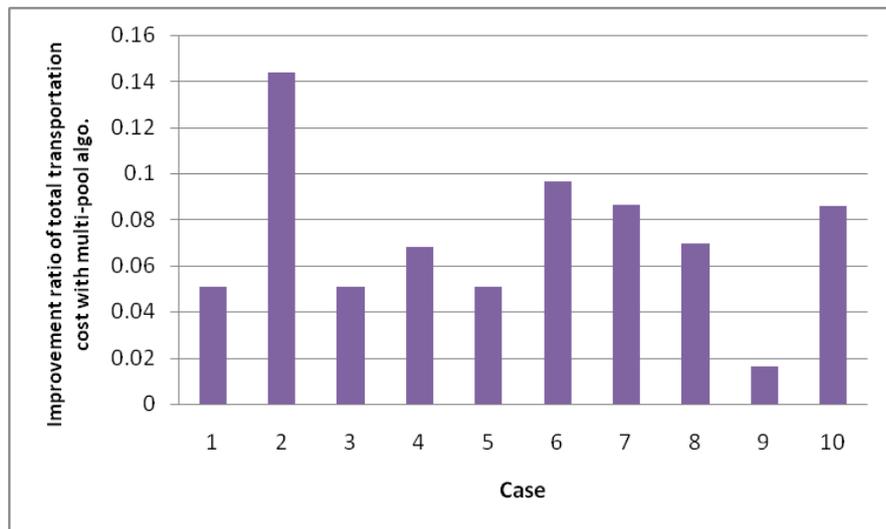


圖 五-10 一百位用戶時傳統合夥用車和多群轉乘合夥用車
受影響群集總移動距離改善比例

接著觀察不限制出席時的情形 (圖 五-11 至圖 五-14)。由於用戶數目、起始位置、跟目的地不變，所以分群狀態的結構上是相同的。原本出席三天以上的用戶，現在可能會變成出席一天或兩天，整體來說總運輸成本會因出席天數減少而下降，可以在圖中明顯的發現，由於出席狀況沒有限制，讓使用者的彈性較大，所以一個群體內包含的用戶數增加，相對地總群體數少，且每天到達乘載上限值的機率較高。在此情況下，孤立點的數量沒有一個明顯的趨勢，當出席天數較少，可能直接被其他相同目的地的群集吸收，導致數量減少；本來參與人數大於二的群集，可能由於其他用戶提前被另外的群集吸收，導致某用戶轉變為孤立點，孤立點數量增加。

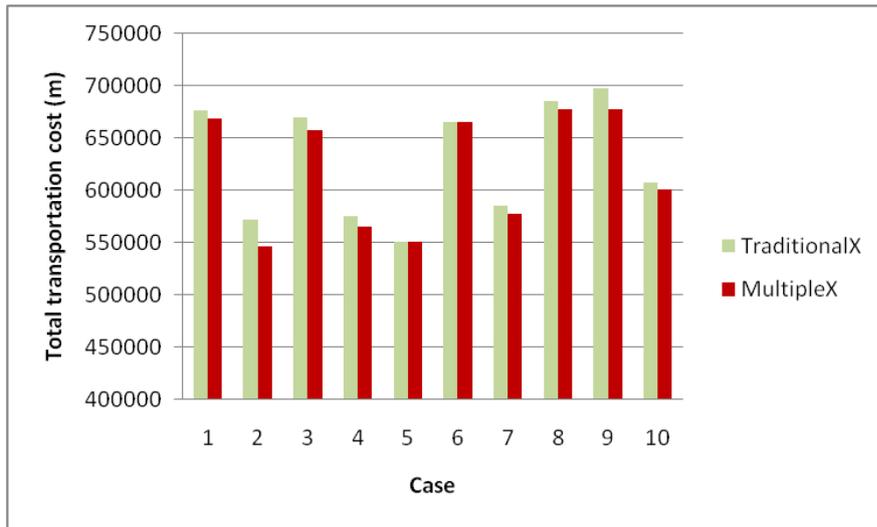


圖 五-11 不限制出席時傳統合夥用車和多群轉乘合夥用車總移動距離比較

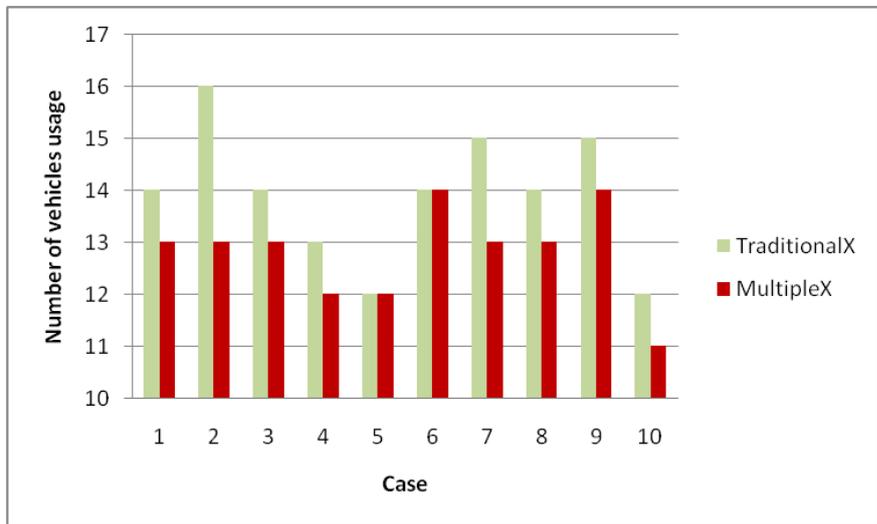


圖 五-12 不限制出席時傳統合夥用車和多群轉乘合夥用車使用車輛數比較

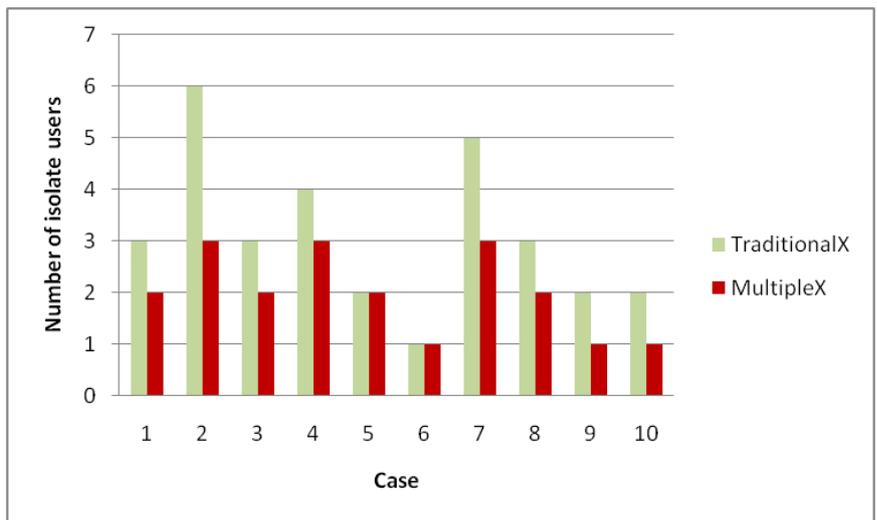


圖 五-13 不限制出席時傳統合夥用車和多群轉乘合夥用車孤立點數量比較

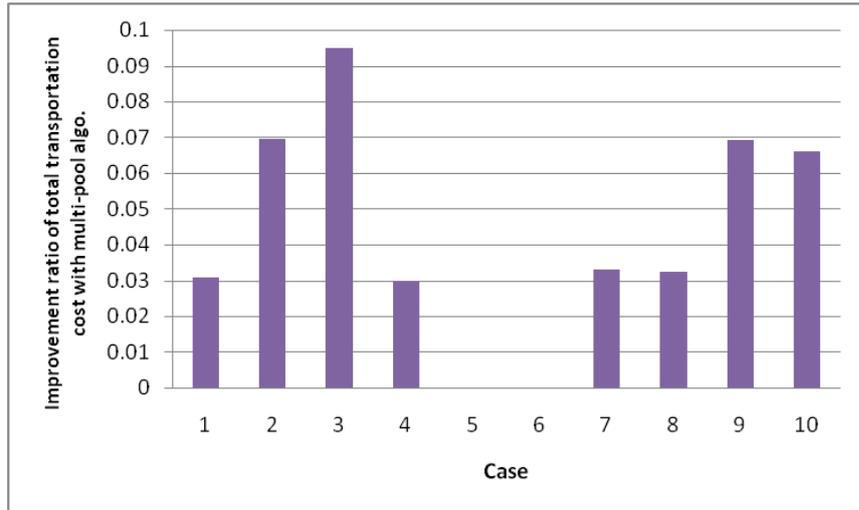


圖 五-14 不限制出席時傳統合夥用車和多群轉乘合夥用車
受影響群集總移動距離改善比例

最後我們觀察當用戶狀況跟出席狀況不變，調整可選擇目的地數量至五個的狀況（圖 五-15 至圖 五-18）。由於目的地數量上升，總人數沒有變化，就平均機率而言，要前往相同目的地的群集數會減少，表示用戶不容易找到要前往相同目的地的同伴。故在此情況下，容易出現孤立點大幅增加的現象。也因為如此，我們可規劃處理的孤立點變多，較高機率找出能夠大幅減少運輸成本的孤立點、目標群、中繼群、中繼點配對，故相對改善效能的程度也較高。

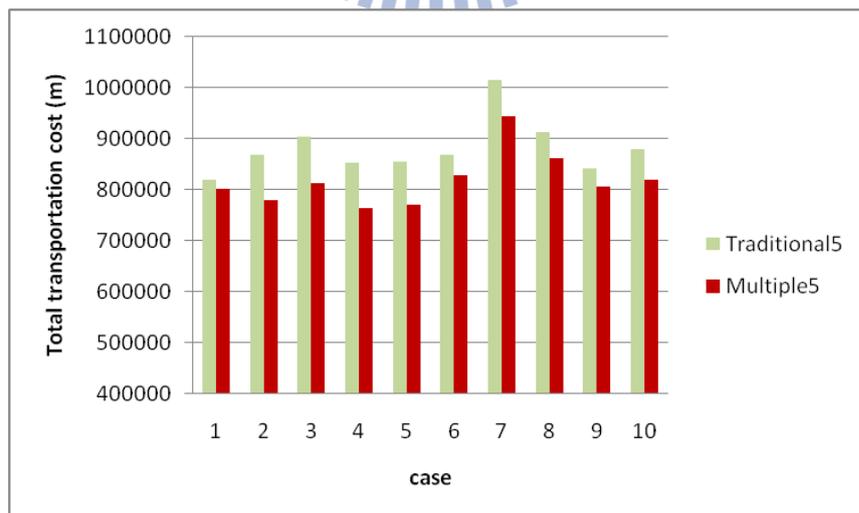


圖 五-15 目的地數量五時傳統合夥用車和多群轉乘合夥用車總移動距離比較

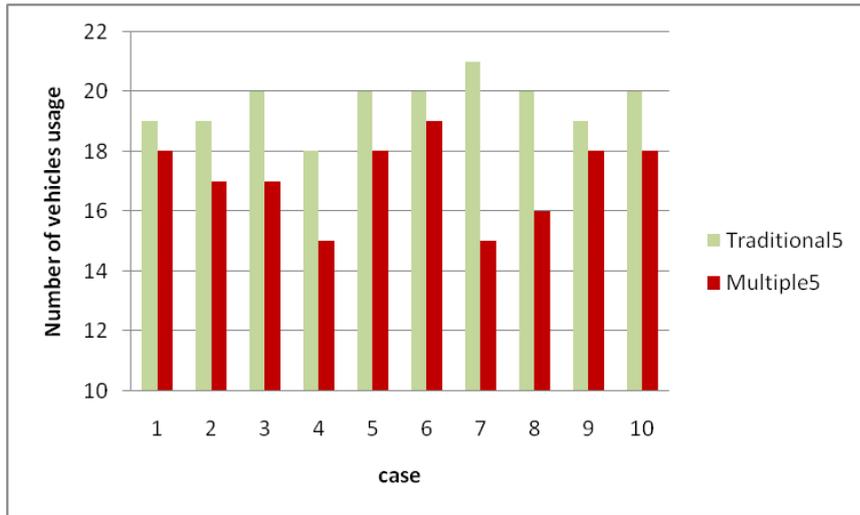


圖 五-16 目的地數量五時傳統合夥用車和多群轉乘合夥用車使用車輛數比較

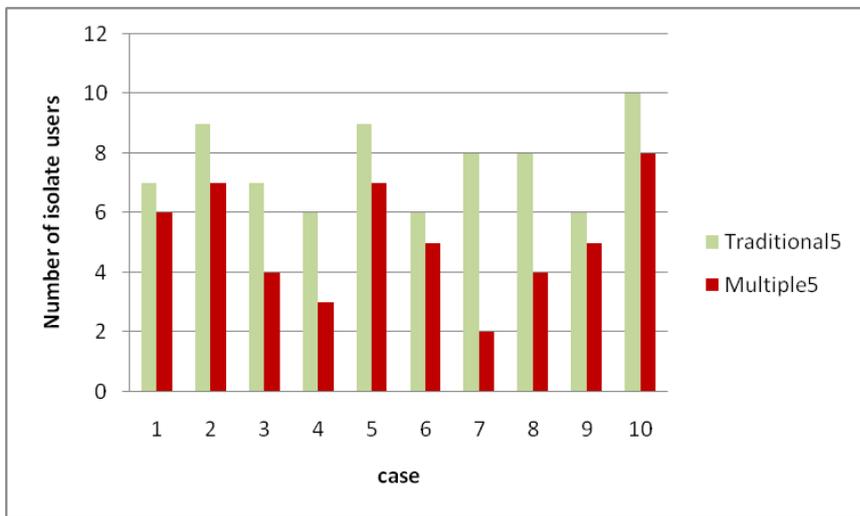


圖 五-17 目的地數量五時傳統合夥用車和多群轉乘合夥用車孤立點數量比較

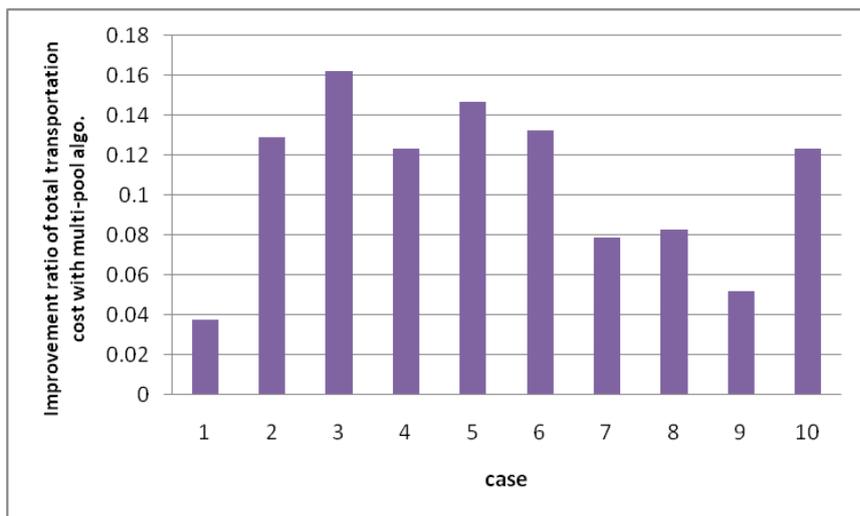


圖 五-18 目的地數量五時傳統合夥用車和多群轉乘合夥用車
受影響群集總移動距離改善比例

第六章 結論與未來展望

因為合夥用車根本上有車輛乘載跟相同群集須相同目的地的限制，故參與現行合夥用車的用戶，不一定能夠得到適合的規劃。所以我們提出了一個改善現有合夥用車的想法，從群體各自運作轉變為多群共同考慮並規劃，在此方法下，可以進一步的減少運輸距離，並減少車輛的使用數目，增加單一車輛的使用率。

我們的方法隱含有公平的觀念在裡面，會由系統計算過後，決定每一天的駕駛，就長久的推算而言，每個用戶駕駛的次數應該趨近於相同。但孤立點有不開車轉乘的這個限制，故只能一直為乘客的角色，因此會使群體變為不公平的狀態。為了解決這個問題，可以試著設計額外的折抵方式來處理。例如，當一個用戶是未成年、本身沒有車輛、或是孤立點等情況不能駕駛時，則他們可以向營運廠商購買代幣。當被其他用戶接送時，則以這些代幣作為代價支付。

合夥用車尚有一個根本的問題，在本文章內沒有特別提出，也就是安全性以及和陌生人一起搭車的問題。由於不知道其他人的來歷，危險性會非常的高。但我們可以利用近幾年熱門的社群網站概念來解決。例如，當一個用戶想要新創一個合夥用車群體時，系統可以將此訊息發送給此用戶的朋友，或者是朋友的朋友，檢查是否有人想前往相同目的地。如此一來，就可以建立起有信任基礎且有安全性的群體。系統也可以提供一些額外的選項，例如興趣選項給使用者挑選，這可以讓有相同喜好的使用者分至同一群體，讓車上的氣氛更好，甚至達到交友的功能。

參考文獻

- [1] 桃北北宜基共乘網, <http://carpool.tpc.gov.tw/carpool/>
- [2] TEXAS A&M UNIVERSITY CARPOOL, <http://carpool.tamu.edu/>
- [3] Megacarpool, <http://www.megacarpool.com/>
- [4] Emilio Ferrari, Riccardo Manzini, Arrigo Pareschi, Alberto Regattier, and Alessandro Persona, “The Car Pooling Problem: Heuristic Algorithms Based On Savings Functions” in *Journal of Advanced Transportation*, Vol.37 No.3, pp.243-272, Autumn 2003.
- [5] Ronald Fagin and John H. Williams, “A Fair Carpool Scheduling Algorithm” in *IBM Journal of Research and Development*, Vol.27 No.2, pp.133-139, March 1983.
- [6] 李兆甯, 許瑋婷, “總站該設在哪裡?—另類的費馬點研究” in 臺灣二〇〇五年國際科學展覽會.
- [7] Philip A. Gruebele, “Interactive System for Real Time Dynamic Multi-hop Carpooling” in *Global Transport Knowledge Partnership*, September 2008.
- [8] Wesam Herbawi and Michael Weber, “Evolutionary Multiobjective Route Planning in Dynamic Multi-hop Ridesharing” in *Evolutionary Computation in Combinatorial Optimization Lecture Notes in Computer Science*, Volume 6622/2011, 84-95, 2011.
- [9] Manel Sghaier, *et al.*, “A Distributed Dijkstra's Algorithm For The Implementation Of A Real Time Carpooling Service With An Optimized Aspect On Siblings” in *13th International IEEE Annual Conference on Intelligent Transportation Systems*, September 2010, pp.795-800.
- [10] Petros Lalos, *et al.*, “A Framework for dynamic car and taxi pools with the use of Positioning Systems” in *Computation World: Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns*, November 2009, pp.385-391.
- [11] K. H. Yew, *et al.*, “Pervasive Car Pooling System using Push Strategy” in *Information Technology, 2008. ITSIM 2008*, August 2008, Vol.1, pp.1-6
- [12] Ted Selker, Paula Helen Saphir, “TravelRole: A Carpooling / Physical Social Network Creator” in *Collaborative Technologies and Systems (CTS)*, May 2010, pp. 629-634.
- [13] Vineeta Chaube, *et al.*, “Leveraging Social Networks to Embed Trust in Rideshare

Programs” in *43rd Hawaii International Conference on System Sciences*, January 2010, pp.1-8

- [14] Roger F. Teal, “Carpooling: who, how and why” in *Transportation Research Part A: Policy and Practice*, Vol.21 No.3, pp.203-214, May 1987.
- [15] Moni Naor, “On fairness in the carpool problem” in *Journal of Algorithms*, Vol.55 No.1, pp. 93-98, April 2005.

