

逢甲大學
交通工程與管理學系碩士班
碩士論文

動態運輸規劃模型之發展與研
究 — 大眾運輸路網之整合
Development of Dynamic
Transportation Planning Procedures –
Integration of Transit Assignment

指導教授：胡大瀛

研 究 生：張庭銀

中 華 民 國 九 十 五 年 七 月

誌謝

研究所生涯隨著本篇論文完成而將結束，回首修業期間點點滴滴，學生首先要感謝的是啟蒙恩師 胡大瀛教授，在恩師的悉心指導及不吝傳授所學下，充實學生在交通領域中的知識與技能，讓學生於學術研究的瀚海中獲益良多，無論在研究上的指導或是生活，老師總是不厭其煩的關懷與叮嚀，在此獻上學生最深的感謝之意。

學生於撰寫論文階段，承蒙所上 邱裕鈞博士細心審閱及指正，使學生論文順利完成，在此一併致上最誠摯之謝意。然於論文口試期間，同時承蒙中華大學 蘇昭銘博士於百忙之中撥冗細心審閱及指正，特此表達感謝之意。在此感謝系上諸多師長充實學生在交通專業上的知識與技能，以及三位助教在選課、學習、設備上的大力支持，讓論文能更加順利完成。

而在研究過程中，也要感謝Pluto Lab成員許多幫忙，感謝麗雯學姊、TT學長提供論文相關意見，百賢學長、文能學長於程式的大力指導，感謝一起在研究所努力打拼的同窗好友憲哥、鵬升、仁維、志豪、阿舜、欣翰、智超、堂展、耿楨、昱彰、家勝、智淮、雅博，有你們一起熬夜趕報告看日出、談天說地、一起慶生…，在我的回憶中烙下深刻回憶。同時要感謝大學同學貴楷、小許，因為你們充實了這段匆匆而紮實生活，特此深表謝意。

感謝奶奶、爸媽無時無刻、無微不至的在背後默默支持與照顧，使我能無後顧之憂，專心的從事研究，方能完成這篇論文；以及感謝哥哥對我在精神上的支持；最後要感謝的是陪伴在身邊的女友彥臻，一起度過這辛苦奮鬥的日子，使緊張刺激的生活中增添了歡笑愉快的色彩。感謝身邊關心我的每個人，因為有你們的支持才會有此成就。在此，僅將我的論文獻給你們，一起分享這份喜悅。

庭銀 謹誌于

逢甲大學丘逢甲紀念館 2006 年 7 月

摘要

目前世界各國都市交通擁擠與尖峰現象一直難以解決，為了因應日益增加之運輸需求，發展新的、高效率之大眾運輸系統並結合動態運輸規劃程序，為日後解決都市交通問題須努力的目標。在智慧型運輸系統架構下，傳統運輸規劃大多考量靜態運輸規劃模型，目前，考量時間因素之動態模型研究，尚無法滿足智慧型運輸系統架構下運輸規劃與交通控制需求。因此，動態運輸規劃之模式規劃應考量依時性旅次、即時性資訊與動態分析之能力，並整合大眾運輸系統的指派以進行交通系統之規劃與分析。

運輸規劃程序中，傳統的大眾運輸指派，以啟發式演算法為主。其中，有許多啟發式演算法修正了道路路網指派的過程，像最短路徑的全有全無指派，或合理路徑(reasonable paths)的多路徑指派，但在此階段並未考量依據旅運者行為決策的數學式；直到近15年，依據行為決策的數學模型才被發展。在一開始，啟發式演算法並沒有考量大眾運輸系統的擁擠影響，就是所謂的大眾運輸指派問題(transit-assignment problem TAP)。其假設所有的運輸路線都有無窮大的容量去提供任何的需求，但是一般道路都具有容量限制所以其不考量容量的影響會產生不合理的結果。所以之後的發展階段都考量擁擠的情況，其考量容量跟等車時間的影響，就是所謂的大眾運輸均衡指派問題(transit equilibrium assignment problem TEAP)。

本研究主要探討在動態運輸規劃程序下，公路路網模型中加入大眾運輸路網，進行大眾運輸指派，之後並探討大眾運輸路網加入後產生動態運輸規劃模型相對應之結構與程序，並將此一程序應用於公車捷運化規劃之指派結果探討，本研究將以構建完成之大眾運輸路網作為實作案例。本研究使用動態交通指派程式(DynaTAIWAN)的動態運輸規劃模型，進行大眾運輸路網模擬，DynaTAIWAN系統可考量不同車種於路網之表現。車種包含小型車、大型車以及公車。整個系統適用都市地區路網。目前系統的

輸出包含小型車及大型車的個體屬性資料、總體性統計資料以及路段屬性資料；而在公車方面並未發展至此，僅止於模擬階段，未來將針對此部分進行深入探討。而使用者可透過DynaTAIWAN系統進行路網之車流模擬，並可進一步以此系統進行相關管理應變措施的模擬評估，以協助擬定方案，並檢視系統各單元與參數之合理性，及分析公車班次與班表變動之依據，以期其應用於智慧型運輸系統的路網動態分析。

關鍵詞：動態運輸規劃、大眾運輸均衡指派、DynaTAIWAN



Abstract

In order to enhance the efficiency of transportation systems, dynamic transportation planning procedure could improve the planning of transportation systems in the context of Intelligent Transportation Systems (ITS). Dynamic models have been developed to utilize ITS technologies; however, results from the dynamic models have not been able to integrate with traditional planning procedures in order to evaluate future traffic conditions.

Traditional transportation planning procedures, due to the static essence, could only be applied in long-term transportation projects. Under ITS, real-time or daily operations of traffic measures depend on long-term planning results; however, current planning procedure could not provide fundamental functions for dynamic analysis. In order to capture dynamic traffic characteristics, transportation planning models should play an important role to integrate basic data with real-time traffic management and control.

In transportation planning procedures, heuristic algorithms have been developed to manage transit assignment. None of the heuristic algorithms proposed considers congestion effects over the transit system. The problem is defined as transit-assignment problems (TAP). The assumption is made that all transit lines have unlimited capacity to accommodate any amount of demand that they could face. If vehicle capacities and waiting times are taken into account, the problem is called transit equilibrium assignment problems (TEAP).

This research is to integrate transit assignment into dynamic transportation planning procedures. The simulation-assignment model of DynaTAIWAN is applied to simulate traffic flows in both road and transit networks. Furthermore, user can simulate the traffic flow in the network through DynaTAIWAN, and can assess the relevant management

measures with this system, to assist in drafting the scheme, and inspect the rationality of every unit and parameter, and analyze the headway of bus and the basic variations of schedule that is applied to dynamic analysis of network of the intelligent transportation system.

The overall system is illustrated through numerical experiments, based on a test network, and dynamic variations in traffic flow patterns and transit operations are observed and compared under different assumptions.

Keywords : dynamic transportation planning 、transit equilibrium assignment 、DynaTAIWAN



目 錄

誌 謝	
摘 要	i
Abstract.....	iii
目 錄	v
圖 目 錄	vii
表 目 錄	ix
第一章 緒 論	1
1.1 研究背景與動機	1
1.2 研究目的	2
1.3 研究方法與流程	2
第二章 文獻回顧	5
2.1 傳統運輸規劃發展	5
2.2 大眾運輸指派原則與方法	7
2.2.1 大眾運輸指派基本概念	7
2.2.2 國內外大眾運輸指派方法	9
2.2.3 大眾運輸指派和公路指派之異同	13
2.3 大眾運輸均衡指派問題	14
2.3.1 大眾運輸指派問題的公式化	15
2.4 運輸規劃軟體之回顧	18
2.5 DynaTAIWAN 系統架構	27
第三章 大眾運輸指派模型架構之建立	31
3.1 動態運輸規劃架構	31
3.2 依時性大眾運輸指派模型運作流程	32
3.3 大眾運輸旅客指派	34
3.3.1 不考量擁擠影響之大眾運輸指派	35
3.3.2 考量擁擠影響之大眾運輸指派	35
3.3.3 不考量大眾運輸容量限制之大眾運輸指派	36

3.3.4 考量大眾運輸容量限制之大眾運輸指派	36
3.4 均衡指派模式	36
第四章 大眾運輸指派流程建立	39
4.1 大眾運輸路網構建	39
4.2 大眾運輸指派流程	43
4.3 程式架構	45
4.4 公車模擬功能測試	48
第五章 數值實驗	52
5.1 50 節點中型路網	52
5.2 初步數值實驗規劃	55
5.2.1 不考量擁擠影響及不考量大眾運輸容量限制	55
5.2.2 不考量擁擠影響及考量大眾運輸容量限制	58
5.2.3 考量擁擠影響及考量大眾運輸容量限制	61
5.3 敏感度測試	62
5.3.1 不考量擁擠影響及不考量大眾運輸容量限制	63
5.3.2 不考量擁擠影響及考量大眾運輸容量限制	64
5.3.3 考量擁擠影響及考量大眾運輸容量限制	65
5.3.4 不同 OD 需求測試	67
5.4 均衡指派測試	68
5.4.1 小結	73
第六章 結論與建議	75
6.1 結論	75
6.2 建議	75
參考文獻	77
附錄一 輸入資料說明	79
附錄二 程式碼	84

圖目錄

圖 1.1 研究流程圖	4
圖 2.1 大眾運輸路網(一)	7
圖 2.2 大眾運輸路網(二)	8
圖 2.3 分區旅次指派到該分區之節點	10
圖 2.4 不對稱成本函數範例	17
圖 2.5 DynaTAIWAN 系統架構圖	28
圖 2.6 系統模擬進行流程	29
圖 2.7 DynaTAIWAN 系統架構之模擬層	30
圖 3.1 動態運輸規劃架構圖	32
圖 3.2 依時性大眾運輸指派模型架構	34
圖 4.1 大眾運輸路網圖	41
圖 4.2 停靠站與道路路網之關係圖	42
圖 4.3 大眾運輸路網轉車圖	43
圖 4.4 大眾運輸指派之流程圖	45
圖 4.5 程式架構圖	47
圖 4.6 大眾運輸車輛指派程式執行之畫面	47
圖 4.7 大眾運輸車輛於班距 5 分鐘之時空圖	50
圖 4.8 大眾運輸車輛於班距 10 分鐘之時空圖	50
圖 4.9 大眾運輸車輛於班距 15 分鐘之時空圖	51
圖 5.1 車輛需求產生型態	53
圖 5.2 測試流程圖	54
圖 5.3 班距 5 分鐘各分區間大眾運輸成本	57
圖 5.4 班距 15 分鐘各分區間大眾運輸成本	57
圖 5.5 班距 30 分鐘各分區間大眾運輸成本	57
圖 5.6 班距 5 分鐘各分區間大眾運輸成本	61

圖 5.7 班距 15 分鐘各分區間大眾運輸成本	62
圖 5.8 班距 30 分鐘各分區間大眾運輸成本	62
圖 5.9 班距 5 分鐘各分區間大眾運輸成本	63
圖 5.10 班距 15 分鐘各分區間大眾運輸成本	63
圖 5.11 班距 30 分鐘各分區間大眾運輸成本	64
圖 5.12 班距 5 分鐘各分區間大眾運輸成本	64
圖 5.13 班距 15 分鐘各分區間大眾運輸成本	65
圖 5.14 班距 30 分鐘各分區間大眾運輸成本	65
圖 5.15 班距 5 分鐘各分區間大眾運輸成本	66
圖 5.16 班距 15 分鐘各分區間大眾運輸成本	66
圖 5.17 班距 30 分鐘各分區間大眾運輸成本	67
圖 5.18 公車路線 2 於班距 5 分鐘下累積人數	67
圖 5.19 公車路線 2 於班距 5 分鐘下累積人數	68
圖 5.20 班距 5 分鐘各分區間大眾運輸成本	69
圖 5.21 班距 15 分鐘各分區間大眾運輸成本	69
圖 5.22 班距 30 分鐘各分區間大眾運輸成本	70
圖 5.23 公車路線 2 於班距 15 分鐘下累積人數	70
圖 5.24 考量均衡路線 2 於班距 15 分鐘下累積人數	71
圖 5.25 公車路線 3 於班距 15 分鐘下累積人數	71
圖 5.26 考量均衡路線 3 於班距 15 分鐘下累積人數	72
圖 5.27 班距 15 分鐘，分區一至分區六乘客搭乘方式	72
圖 5.28 考量均衡班距 15 分鐘分區一至六乘客搭乘方式 ...	73

表目錄

表 2.1 常用運輸規劃軟體整理表	26
表 4.1 時間權重整理表	44
表 4.2 大眾運輸車輛輸入檔	48
表 4.3 大眾運輸車輛追蹤檔	49
表 5.1 50 節點路網屬性設定值	52
表 5.2 不同班距下各路線統計值	56
表 5.3 不同班距下各路線統計值	59
表 5.4 不同班距下各路線統計值(續)	60
表 5.5 不同班距下各路線人數移轉百分比	74
表 5.6 不同班距下各路線人數移轉百分比	74



第一章 緒論

1.1 研究背景與動機

世界各國都市交通擁擠與尖峰現象一直難以解決，為了因應日益增加之運輸需求，利用動態運輸規劃程序下，發展高效率之大眾運輸系統，為日後解決都市交通問題的關鍵。而傳統運輸規劃大多考量靜態特性，考量時間因素之動態模型研究，尚無法滿足智慧型運輸系統架構下運輸規劃與交通控制需求。因此，動態運輸規劃之模式規劃應考量依時性旅次、即時性資訊與動態分析之能力，並整合大眾運輸系統的指派以進行交通系統之規劃與分析。

運輸規劃程序中，傳統的大眾運輸指派，以啟發式演算法為主。這些啟發式演算法修正了道路路網指派的過程，像最短路徑的全有全無指派，或合理路徑(reasonable paths)的多路徑指派，但在大眾運輸指派一開始發展階段尚未考慮旅運者行為決策；直到近20年，依據行為決策才被納入。在傳統大眾運輸指派一開始的發展，所提出之啟發式演算法並沒有考量大眾運輸系統的擁擠影響，就是所謂的大眾運輸指派問題(Transit-Assignment Problems, TAP)。其假設所有的大眾運輸車輛都有無窮大的容量去提供任何的需求，但是一般大眾運輸車輛都具有容量限制故不考量容量的影響會產生不合理的結果。所以之後的發展階段都考量擁擠的情況，其考量大眾運輸容量跟等車時間的影響，就是所謂的大眾運輸均衡指派問題(Transit Equilibrium Assignment Problems, TEAP)。

而大眾運輸指派模式的發展過程中，在1980年之前，許多學者即開始處理大眾運輸指派問題，但以往的研究者往往基於運算效率的考量而對乘客的選擇行為採取簡化的假設。直至目前已經提出了許多的演算法，而且這些演算法已經被使用去分析現實的大眾運輸路網。有鑑於此，本研究希望藉由蒐集國內外相關的研

究成果，於不同假設條件下加以深入的分析與比較，試著討論大眾運輸容量限制與等車時間之關係，並建立合理的依時性大眾運輸均衡指派模式。

本研究使用動態交通模擬指派程式(DynaTAIWAN)，進行大眾運輸路網模擬，DynaTAIWAN系統可考量不同車種於路網之表現，而使用者可透過DynaTAIWAN系統進行路網之車流模擬，並可進一步以此系統進行相關管理應變措施的模擬評估，以協助擬定方案，並檢視系統各單元與參數之合理性，及分析公車班次與班表變動之依據，以期其應用於智慧型運輸系統的路網動態分析。

1.2 研究目的

本研究主要探討在動態運輸規劃程序下，公路路網模型中加入大眾運輸路網，進行大眾運輸指派，並探討大眾運輸動態運輸規劃模型之程序。

有鑒於大眾運輸指派在運輸規劃、管理及設計上之重要性，本研究將探討大眾運輸指派方法；本研究主要的目的可歸納為以下幾點：

1. 蒐集文獻，分析、比較現有之大眾運輸指派模式在假設條件與推導方法上的差異。
2. 描述在大眾運輸路網上，較符合實際情況之乘客選擇路線行為，並建立大眾運輸均衡指派模式。
3. 應用所構建出的大眾運輸指派模式，分析大眾運輸系統在市區道路上經營的績效及改善的策略，以提供市區道路交通量整合指派之基礎。

1.3 研究方法與流程

本研究進行的方法與流程說明如下，圖 1.1 為本研究之流程

1. 針對研究課題進行瞭解

對於本研究的背景與動機做一瞭解，確立研究方向，訂定課題相關的一些假設與限制，以及研究的目的。

2. 文獻回顧

本研究所回顧的相關文獻主要包含傳統運輸規劃之大眾運輸指派原則與方法、大眾運輸旅次指派和公路交通量指派之異同、大眾運輸均衡指派問題，除了針對大眾運輸指派的問題及其解法進行回顧外，更針對大眾運輸指派問題與演算法進行回顧。

3. 大眾運輸均衡指派架構與演算法設計

提出依時性大眾運輸均衡指派概念性架構，根據概念性架構建構本研究的大眾運輸指派模擬評估架構；以依時性均衡指派的概念，設計大眾運輸車輛指派的演算法，並說明其演算法流程。

4. 程式設計

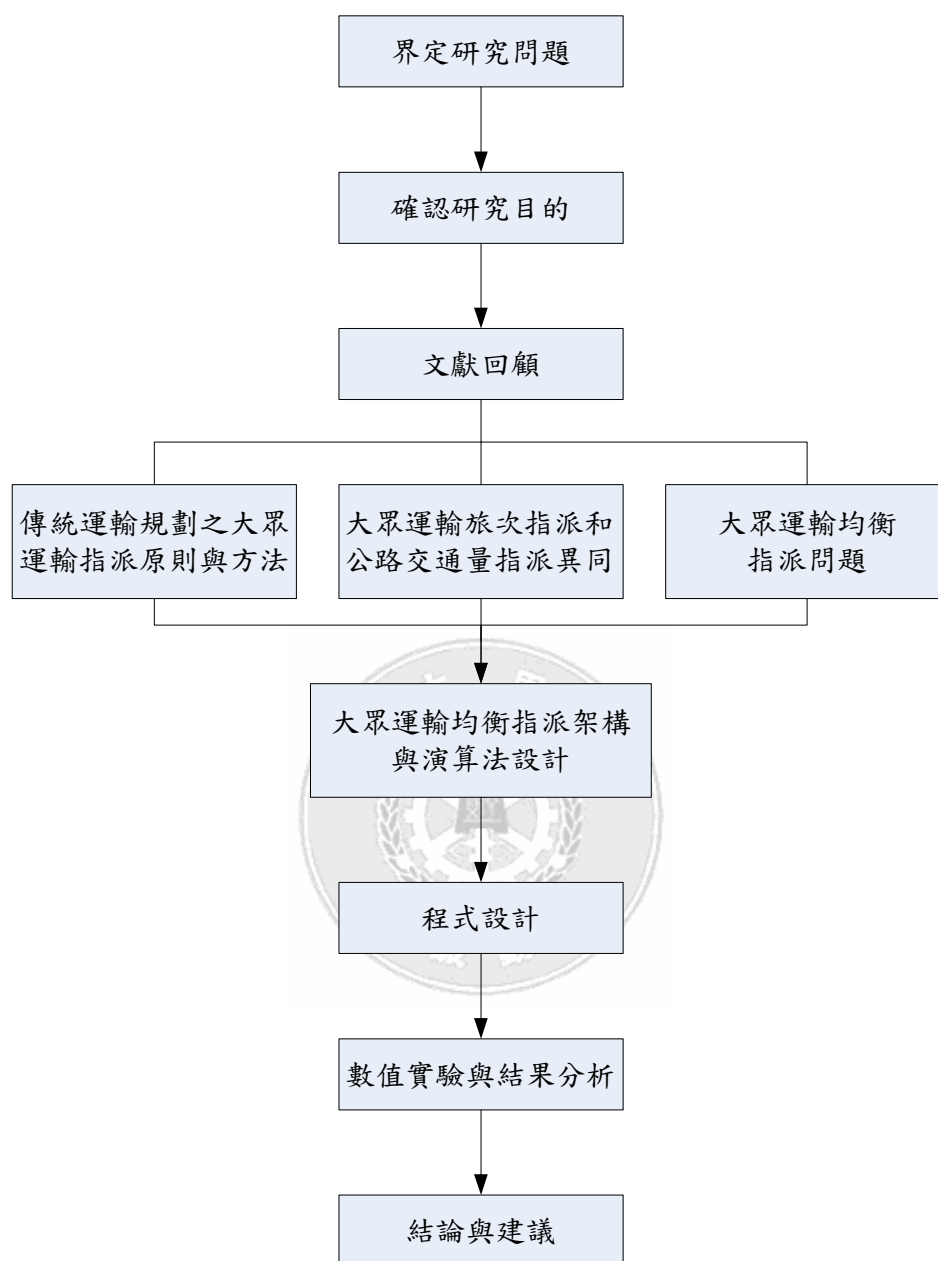
本研究將以物件導向(Object-oriented, OO)程式語言C++，進行程式的設計，並結合交通模擬模組(DynaTAIWAN)，將大眾指派程式所輸出得結果回饋至DynaTAIWAN進行模擬與分析。

5. 數值實驗與結果分析

為了測試演算法的適用性，本研究以構建完成之大眾運輸路網進行數值實驗與演算法的驗證，透過所構建的大眾運輸均衡指派模式，進行大眾運輸車輛路線之評估與分析。此一架構為結合大眾運輸均衡指派之演算法與動態交通模擬指派模式DynaTAIWAN，希望能透過此模擬評估實驗結果並做分析討論。

6. 結論與建議

根據數值實驗所得到的結果，提出本研究的結論與建議，供後續研究此課題作參考。



資料來源：本研究整理

圖 1.1 研究流程圖

第二章 文獻回顧

本章將於2.1節介紹傳統運輸規劃；2.2節說明大眾運輸指派原則與方法；2.3節討論大眾運輸均衡指派問題；2.4節將回顧運輸規劃軟體；2.5節說明DynaTAIWAN模擬指派模式之架構。

2.1 傳統運輸規劃發展

傳統運輸規劃方法主要可分為總體程序性規劃法、總體直接性規劃法以及個體規劃法等，本節將針對總體程序性規劃法作簡單之回顧與介紹。

總體程序性規劃法係以交通分區為分析之基礎，並將所規劃之地區劃分為幾個交通分區，由交通分區中土地及社經特性計算出旅次發生量，並由分區間吸引與旅次目的特性計算出旅次分佈，再依據旅運者之旅運行為選擇所使用運具之運具分配，最後再依據系統最佳化、使用者均衡等原則選擇路線並將旅次指派至路網中以完成規劃預測工作。以下就總體程序性規劃法之四個階段分作介紹【施鴻志等，1988、李克聰，1995、王慶瑞，1996、凌瑞賢，2001】：

1. 旅次發生(Trip Generation)

旅次發生為傳統運輸規劃之第一個階段，依據某交通分區之土地使用、區位及社會經濟等特性，估計離開或進入該分區之總旅次數；而離開該分區之總旅次數即為旅次產生(Trip Production)，進入該分區之總旅次數則為旅次吸引(Trip Production)。旅次發生主要方法可分為成長率法、土地使用產生或吸引率法、多元線性迴歸法、類目分析法等。

2. 旅次分佈(Trip Distribution)

旅次分佈為總體程序性運輸規劃之第二步驟，主要在分析交通分區間流動之旅次數，並考量各交通分區之旅次生產量以及未來運輸設施資料與影響，當旅次分佈程序在旅次發生之後，即可獲得所有運具之流動旅次數，進而建立一個模式作為預測及規劃

之依據。使用方法包括成長因素法(Growth Factor Method)、重力模式(Gravity Model)、機會模式(Opportunity Model)。

3. 運具分配(Modal Split)

一般運輸規劃是將運具分成大眾運具(Public Transit)和私人運具(Private Vehicles)。旅運者對運具的選擇通常是基於社經特性與價值觀而有所不同，運具分配主要的工作係從總旅次數中推估各種運具的旅次數。通常運具分配可以在旅次分佈之前後，如果在旅次分佈之前就是旅次端點之運具分配，主要用於個人運具不發達之中小型都市；在旅次分佈之後就是旅次交換運具分配，主要用於大眾運具較為發達之都市。

近年來由於個體選擇模式(Individual Choice Model)快速發展，個體選擇模式已普遍被用來作為運具選擇分析與預測，其他主要常用模式有旅次端點運具分配模式(Trip End Modal Split Modal)、旅次交換運具分配模式(Trip Interchange Modal Split Modal)、二階段運具分配模式(Two Stage Modal Split Model)、個體行為運具分配模式(Disaggregate Behavioral Modal Split Model)、模式估計方法(Model Estimation)、多項羅吉特模式(Multinomial Logit Model)以及普羅比模式(Probit Model)等。

4. 交通指派(Traffic Assignment)

交通量指派又稱路網指派(Network Assignment)，為程序性需求分析之最後階段，可分為公路交通量指派與大眾運輸交通量指派兩種。依據成本最低或時間最短之原則，將旅次分佈之交通分區間旅次量依照旅運者運具分配比率指派到路網上，以得到路段上之交通量。

指派方法以Wardrop(1952)提出使用者均衡指派原則(User Equilibrium Assignment Principle)與系統最佳化指派原則(System Optimum Assignment Principle)為基礎。

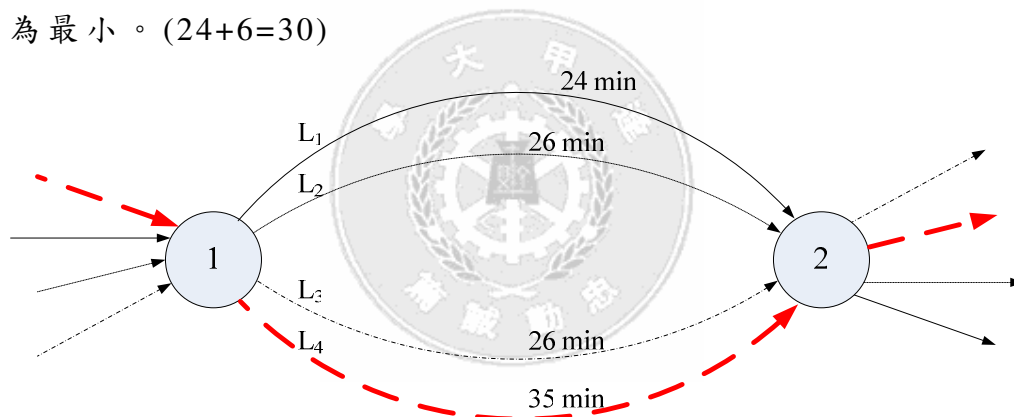
2.2 大眾運輸指派原則與方法

本節於2.2.1節介紹大眾運輸指派基本概念，2.2.2節簡單回顧國內外之大眾運輸指派方法，並於2.2.3節比較大眾運輸旅次指派和公路交通量指派之異同。

2.2.1 大眾運輸指派基本概念

大眾運輸指派模型最主要的不同即依據使用者在路徑選擇決策時不同的行為而有不同之假設。最簡單的方法是假設大眾運輸的旅客從起點A到迄點B去選擇路線或一連串的路徑，如果旅客選擇一連串的路徑，則一般的旅行成本應該較少。

舉例來說，如圖2.1，依據Naive的方法，我們可以知道一個乘客從點1到點2有4條替代路線，我們可以知道路線1的總旅行時間為最小。(24+6=30)



資料來源：D.A. Hensher and K.J. Button, 2000

圖 2.1 大眾運輸路網(一)

Le Clerq(1972)是第一個介紹共同路線(common lines)的概念，也就是說，從點1到點2的每一條路線都有不同的班表和旅行時間。他的演算法也考慮了起迄點間所有的路線。因此，假設：

1. 等車時間等於班距/2
2. 車內旅行時間等於所有路段旅行時間的平均

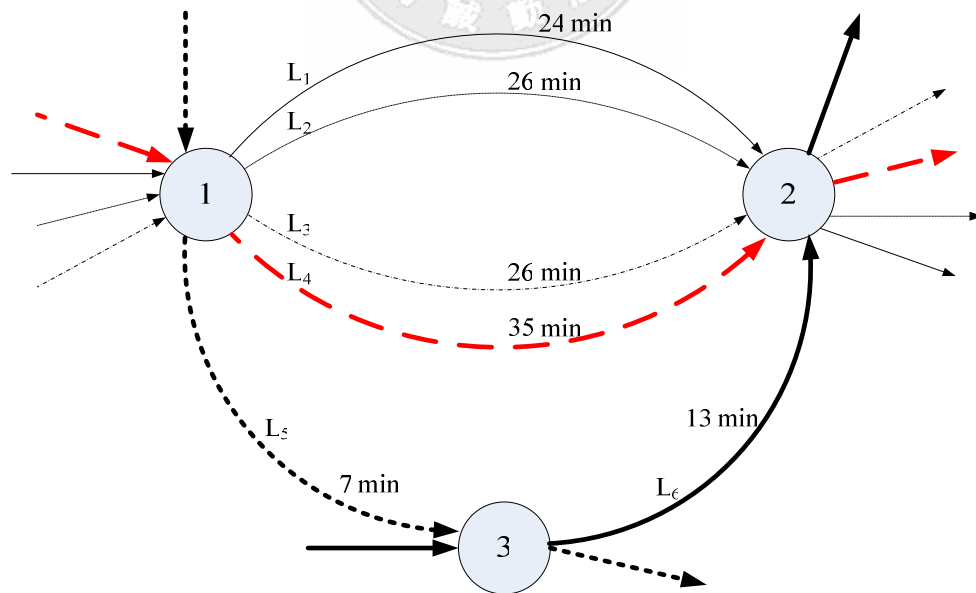
舉例來說，假設一虛擬班表，每小時40輛車，現在期望的總旅行時間為29.25分鐘(等車時間為1.5分鐘，車內旅行時間為27.75分鐘)，他比依據缺乏經驗的行為(naive behavior)所需的總旅行時間低0.75分鐘。

在這個例子，假如旅客要從點1到點2，當有車到達點1時，旅客就會去搭乘，這個方式跟使用缺乏經驗的行為(naive behavior)比較，會產生較少的旅行時間(如果車內旅行時間是相同的)。如果車內旅行時間是不相同的，依據班表和車內旅行時間，可能會得到較大的值。

Chriqui(1974)和 Chriqui and Robillard(1975)補充LeClerq所提出的方法。假設在起迄點間只考慮一部分的路線，如上例，僅考慮3條路線L1，L2，L3。總旅行時間為27.33分鐘(等車時間2分鐘，旅行時間25.33分鐘)，如果有一個旅客在點1要到點2，他將會搭乘第一部到達點1的車輛，而這部車輛所走的路線是屬於L1，L2，L3集合的任意路線，而旅行時間25.33分鐘。

為了解釋Spiess(1983)所提出策略的觀念(允許大眾運輸使用者從每一個起點到一個被給定的迄點的一些規則)，並區分路線(route)的觀念，我們重新定義圖2.2的路網，增加一個新的節點(node 3)，這兩條新路線的班表為每小時10輛車，如圖2.2所示。

1. 沒有考慮轉運：總旅行時間為27.33分鐘(2+25.33)
2. 考慮轉運：總旅行時間32分鐘(12+20)



資料來源：D.A. Hensher and K.J. Button, 2000

圖 2.2 大眾運輸路網(二)

2.2.2 國內外大眾運輸指派方法

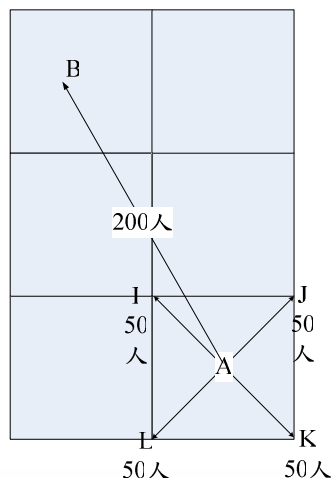
本節探討在已知路網下，進行大眾運輸指派問題(Transit-Assignment Problem, TAP)。在1980年之前，即有許多學者開始研究大眾運輸指派相關問題，直至目前已經提出了許多的演算法【Dial(1967)、Le Clerq(1972)、Chriqui(1974)、Chapleau(1974)、Andreasson(1976)以及Rapp et al.(1976)】，並且都用以解決實際大眾運輸路網問題。此外，Lampkin and Saalmans(1967)、Scheele(1977)、Mandl(1980)以及Hasselstrom(1981)在設計大眾運輸路網的模型上(如網狀路網)，亦考慮此演算法。而Florian(1977)以及Florian and Spiess(1983)將其運用於多運具的路網(multimodal-network)均衡上。

大眾運輸使用者在選擇路徑上，不僅考量其所面對之大眾運輸路線，且尚有時間方面之限制，因此大眾運輸路網的路徑選擇較公路運輸路網的選擇行為複雜。時間方面限制如服務路線的行車時刻表、轉車次數、轉車時間、以及兩端出入的步行時間等。而大眾運輸旅次指派首要工作即利用已建構之大眾運輸路網，計算各起迄點間之旅運時間，再進行旅次指派。

目前有關大眾運輸指派的文獻，不僅止於探討原則性問題，尚有很多實作系統，詳細指出影響旅客選擇路線的因素為旅行時間、旅行距離、停靠站數、等車時間、及站時間(Access time)、票價與舒適等，目前已發展出整套的指派模式。一般大眾運輸指派方法，多採用全有或全無指派法或多重路徑指派法來進行指派。以下為國內外大眾運輸指派方法之討論【鄭樂堯，1985，王慶瑞，2001】：

1. 1973年，宋明山在研究公車系統網路交通量分派時，以分區中心點至該分區節點距離之大小，指派到各節點上，在進行節點間最短路徑指派，如圖2.3所示。若A分區至B分區旅次需求量为200人，假設各分區中心點到各分區內不同節點之距離相等，而以同一比率，將旅次指派到各節點上，因此A-I，A-J，A-K，A-L各指派50人，同理，I-O，I-N，I-P，I-Q各指派12.5人。缺點為

未考慮到公車路線是固定的，僅把公車旅次指派到現有的道路上。



資料來源：宋明山，1973

圖 2.3 分區旅次指派到該分區之節點

2. 1966年，美國住宅及都市發展部(Department of Housing and Urban Development ,HUD)，以等車時間(班距之半)、車上時間、全程距離及停靠站數做為選取最短路徑之依據，並對不同型態的旅次採用不同的指派方法。如分區內至分區外採全有或全無指派，分區內至分區內則以各路線之班次比率做為旅次指派之依據。各分區之旅次均假設發生在分區中心上，再依據分區中心至鄰近車站之步行時間，將旅次分配到各車站上，最後再進行站與站間的指派工作。缺點為僅考慮各分區之旅次將其分派至鄰近車站，未視路網情況從事整體考慮。
3. 1967年Lampkin和Saalmans在研究公車路線設計中，以班距為主要因素，對於站間可以直達之路線，進行指派。假設不採用任何行駛時間大於最佳路徑行駛時間加上1/2班距之路線，以減少可選用之路線。缺點為此法適用於簡單之路網指派，但如涉及轉車或站數龐大之路網，則指派工作甚為困難。
4. 1972年Clercq假設每個分區中心與一特定的節點相連接，並以轉車因子、等車時間與車上時間為主要考慮因素，尋找站與站間

的最短旅行時間路徑，而將分區旅次全數指派到所選取路徑上。而等車時間的計算採用班距的一半，即：

$$WT=0.5*60/FQ$$

WT：等車時間，

FQ：該路線每小時到達之班次數，

Clerc所提出之指派方法，可應用於較複雜之路網和轉車情形，但是對於重複路線(Overlap routes)多之路網較不適用。

5. 1974年Silman, Barley和Passy研究公車路線規劃時，以班次為因素，對區間可直達之路線計算其被乘客選用之機率，選取區間小於最短路徑1.5倍旅行時間之直達路線，做為指派之路線，並提出兩點主要假設：

- (1) 假設各路線班車之到達間隔，遵循事先所排定之固定班距。
- (2) 假設乘客是隨機地到達車站，而選搭最早到且可到目的地之班車。

此指派方式與Lampkin和Saalmans所提出的分配方式相似，但此方法只適用於區間指派。

6. 1976年Last和Leak發表一公車規劃模式TRANSPET，該研究認為路線上游(Upstream)乘客搭乘的多寡，會影響到下游乘客選搭路線班車的行為，因為上游乘客如果過多，該班車之容量或座位已被占滿，下游乘客將感受到擁擠而改搭下一班或其他可達目的地之路線。缺點為在重複路線多之路網，處理較為困難。

7. 1982年Han和Wilson在研究高度使用重複路線之路網公車車輛配置時，將指派之流量分為兩類：

- (1) 沒有選擇流量(Captive flow)

乘客只能使用唯一的路徑到達目的地，這些旅次指派量就定義為沒有選擇流量。

- (2) 可變流量(Variable flow)

乘客從出發地前往目的地，有多條路徑可選用，這些旅

次指派量就定義為可變流量。

對於沒有選擇流量係選取旅行時間最短之路徑，採全有或全無指派法，至於可變流量則依班次分配原則(Frequency-share rule)，將區間旅次量分派到各路徑上。此方法可應用於有轉車及重複路線多之複雜路網上，但是它的分派比率係利用簡單之班次比，立論基礎較為薄弱。

8. 2001年王慶瑞於「運輸系統規劃」提出大眾運輸旅次指派方法依其所需之結果，可分成三種同之方法：系統指派法、路線指派法及時刻表指派法。

(1) 系統指派法

該指派法類似公路路網的全數指派法(All or Nothing Assignment)，將兩交通區間的大眾運輸旅次全部指派到最短路徑上，再累積計算各路鏈(Link)及路線(Line)的旅客運量，此種指派的結果因沒有路線容量限制，故可視為大眾運輸的期望路線網(Desired Line Network)，但不能有效反應實際路線的營運績效，因此僅適合應用於新路線網方案研擬的初步分析。

(2) 路線指派法

路線指派法之作業程序類似多重路徑比例指派法(Multipath Proportional Assignment)，可分為路徑搜尋(Route Search)、路徑篩選(Route Choice)、及路徑分配(Route Split)等三個步驟。

(3) 時刻表指派法

適用於班次較疏或行車班距較長的大眾運輸路網以及按時刻表行車的鄉區或城際大眾運輸路網。基於行車時刻表的考量，可利用最短路徑法計算兩交通分區的最佳連線(Best Connection)，在某特定時段裡，每個不同開車時間(Departure Time)皆可計算出一使用不同大眾運輸路線及不同轉車站之最佳連線，最佳連線之旅次指派方法可分為：類似全數指派法

與似路線指派法。

綜合上述，可知目前各種大眾運輸指派方法所存在之問題為：路線選取問題、區間成本表示、轉車之處理問題、重複路線之處理問題。

2.2.3 大眾運輸旅次指派和公路交通量指派之異同

大眾運輸旅次指派和公路交通量指派都是模擬使用者選擇路線之行為，但是在處理過程卻有顯著的不同。將可歸納為以下幾點：

1. 路網之描述不同

公路交通量指派係經由現有的道路路網來指派，而大眾運輸指派由於具有固定之路線，因此，一般均以路段來描述站與站間之部分路線，而實際上大眾運輸路網是由許多路線所組成的。

2. 路網屬性不同

一般公路路網之屬性為路段長度、車流速度及道路容量等。而大眾運輸路網還須加入班距(Headway)、容量、站間距離、行駛時間及場站設備等。

3. 使用者路線選擇因素不同

公路交通量指派中，係以最短距離或最短時間來當作選擇路線之依據。但是在大眾運輸指派中，仍須考量等車時間(waiting time)、車上時間(in vehicle time)、進出大眾運輸系統所需時間、可搭乘之路線數、轉車次數等，將上述合併所得之最大效用當成路線選擇之依據。且在公路路網上，使用者可以任意改變路線，但是大眾運輸使用者只能在少數的節點上改變路線，因此公路路網之使用者比大眾運輸使用者有較多的選擇性。

2.3 大眾運輸均衡指派問題

啟發式演算法起初並未考慮大眾運輸系統的擁擠影響，即假設所有大眾運輸車輛均有無限大的容量滿足任何的需求。Last and Leak 於1976年則首先考量擁擠影響，此研究在每一條路線上的車輛容量和遞增的等車時間加上限制條件，如轉車懲罰，但該模型僅適用於輻射狀路網，因此並未有進一步實際的研究與應用。

Spiess(1983)和Spiess and Florian 於1989年提出了大眾運輸指派的線性規劃問題與演算法。其目標式即假設旅客將一般的旅行時間減到最小，而旅客在大眾運輸路網的起迄點間可選擇多條獨立的路徑，且在這些路徑上的旅行時間跟等車時間都跟旅客人數多寡有關。De Cea(1986)以及De Cea and Fernanelez(1989)則依據Spiess(1983)、Le Clerq(1972)與Chriqui(1974)所提出共同路線(common lines)和大眾運輸路線(transit route)的觀念進而提出另一個大眾運輸指派的線性規劃模型。而Last和Leak提出此兩模型特例之演算法，並不考慮擁擠的影響，但若是用於高擁擠程度之大眾運輸路網下，則必須考量擁擠變數的影響，因此隨後之發展階段均考慮擁擠的情況，即大眾運輸均衡指派問題(Transit Equilibrium Assignment Problem TEAP)。以下為國外所提出之大眾運輸均衡指派問題。

1. Spiess(1983)和Spiess and Florian(1989)提出車內旅行時間會隨著旅客流量的增加而增加的線性模型，並假設在停靠站的等車時間不受大眾運輸容量的影響。
2. Nguyen和Pallottino(1988)提出了一個類似公式，係以hyperpath的概念，可運用於擁擠道路路網。
3. Gendreau(1984)提出擁擠狀況下的車內旅行成本跟等車時間，它運用在特殊的大眾運輸系統，且並未考慮共同路線(common lines)。對於兩個停靠站間有多條路線，Gendreau則根據等候理論提出一個均衡的模型，Gendreau並未實際發展出TEAP的公式，但提出了等車時間會隨著旅客流量遞增而增加。
4. De Cea和Fernandez(1993)提出了一個新的TEAP公式，此公式

裡的等車時間都是依據旅客的流量。該研究亦使用相同的特殊路網去解決他們在1989年所提出的TAP，他們考慮一般的成本函數，而成本函數中的等車時間為自由流下的等車時間之總和，而且也依據旅客流量的增加而增加等車時間。

5. Spiess and Florian(1994)提出了一個非線性模型，這個模型是延伸他們在1989年所提出的模型，其中的等車時間是依據班次以及擁擠情況下的函數。

2.3.1 大眾運輸指派問題的公式化

1. 不考量擁擠的情況

基於大城市裡的大眾運輸路網存在著共同路線的概念，故若不考量擁擠情況，會使其結果不符合實際情況。因此Spiess 和 Florian 提出大眾運輸路網的數學式：

$$G=(N,A)$$

其中，N：大眾運輸停靠站的集合

A：路線的集合

另外仍須考量每一路線上、下車之停靠站數。

已知旅行時間等於車內旅行時間加上等車時間，在路線的部份(line segments)，假設一個班次密集的班表之等車時間為零，故其旅行時間等於車內旅行時間。在上車的停靠站，由於沒有旅行時間，因此可由班表決定等車時間。並且利用線性規劃的方法來處理這個找出最小化問題，並導出一個多項式時間演算法。

Nguyen和Pallottino依據相似概念把模型運用在Torino。該研究定義一個叫做hyperpath的策略，係以一個無循環之有方向性的路網(acyclic directed graph)來表示。此模型與Spiess and Florian在1989年非常相似【Nguyen和Pallottino, 1986】。

Chriqui(1974)使用之模型並未考量時間因素，而De Cea(1986)則依據Spiess(1983)所提出之觀念將大眾運輸指派到最短路徑上的線性規劃模型公式化。

De Cea and Fernanelez(1989)於聖地牙哥的大眾運輸路網針

對一非線性之公式做測試【De Cea et al., 1988、De Cea and Fernanelez, 1989】，此公式相當於大眾運輸指派到最短路徑上的線性問題，但該結果比Chriqui的演算法更有效率，而且比Spiess的演算法快兩倍。此非線性的模型不但能確實的得到起點至各點間之旅次指派，亦能獲得各起迄點間之最短路徑。

2. 考量擁擠的情況

於擁擠的大眾運輸系統下可由TAP得到兩種TEAP模型，由於大眾運輸車輛的容量限制，故此兩個模型都考慮在大眾運輸停靠站的等車時間，而等車時間是一個旅客流量的遞增函數。

De Cea and Fernandez(1993)依據路線的觀念提出一個大眾運輸均衡問題。即當考慮大眾運輸車輛的容量限制時，使用一個虛擬(nominal)的班表決定一條路線，故不符合實際情況，應該用一個有效的班表來取代。因此，依據大眾運輸路網的流量給定一個有效的班表，就能區分De Cea and Fernandez所提出的非擁擠的模型。一個連結的存在與否是跟路線的變化有關，應該依據擁擠程度去結合班表和總期望的旅行時間。為了解決擁擠的問題，定義一個修正過的路網，該路網在起迄點之間可以有不只一條的連結，其定義如下：

$G'=(N,S')$ 表示路線的集合(groups of lines)

其中，N：節點集合

S' ：大眾運輸連結的集合

非擁擠路網下，第一條連結會跟有吸引力的路線有關，所選擇的路線就叫做最快速的路線(fastest lines)。如果兩條路線仍具相同的起迄點，第二條平行的連結，就要考慮這些路線是否重疊，而此步驟一直持續到所有新的路線和某些連結相關，起迄點間的路線數量會隨著擁擠狀況而變化，因為當路網上的流量增加，將會使用平行的連結去維持均衡的狀況。

在定義大眾運輸連結的特殊路網後，TEAP變得相似於標準道路路網的交通指派，而且還有不對稱的成本函數，該成本符合大

眾運輸連結的使用，並假設為下列之總合：

- (1) 一般車內旅行成本(車內旅行時間加上票價)
- (2) 自由流下的等車時間
- (3) 等車時間的可變部份，為一個在大眾運輸路線上旅客流量的遞增函數

不對稱的成本函數(舉例如圖 2.4 所示)：

有一條路線點1到點3，會經過點2，當有一個旅客從點1要到點3，會影響在點2上車旅客的等車時間，因為他們使用部份的容量，而且當通過點2時，某些車輛可能是滿載的情況。但是，在點2上車的旅客並不影響在點1上車旅客的等車時間。



資料來源：本研究整理

圖 2.4 不對稱成本函數範例

第二個TEAP的數學式是由Wu et al.(1994)提出的，這個模型方法類似於De Cea and Fernandez所提出的方法，但以hyperpath的觀念取代大眾運輸路線，其定義為：

$$G=(N,A)$$

其中，N：大眾運輸停靠站數，包含轉運的節點

A：連結的集合，符合大眾運輸路線、步行的連結、停等的連結、上下車的連結

由於在自由流情況下的等車時間是依據路線的班表，所以在hyperpath的流量問題不能轉換成變分不等式的問題。

Wu et al.(1994)提出一個symmetric linearization algorithm 去解決變分不等式問題。

2.4 運輸規劃軟體之回顧

目前可採用之運輸規劃應用軟體，基本上大都採用旅次發生、旅次分佈、運具分配及交通量分派等四個階段來作運量預測。近年來又有將地理資訊系統(GIS)及資料庫(Data Base)等功能相互結合，使路網圖形的處理及使用者操作界面更具親和力，以下是較常使用之運輸規劃軟體說明，整理後如表2.1示【亞聯工程顧問股份有限公司，2001】：

1. DOTS I

DOTS I 模式主要是以台北市捷運局之TRTSⅢ軟體而得，其採傳統循序性集體旅運需求模式—旅次發生、旅次分佈、運具分配、交通量分派四大步驟，其包含之運輸規劃分析軟體為TRANSPORT及TRANPLAN及台北捷運局自行發展之分析模組TAIPEI三部分整合而成，其交通量分派模組包括大眾運輸路網運量分派模組及道路路網交通量分派模組兩部份，而其大眾運輸路網運量分派則運用全數分派法(All-or-Nothing Assignment)，將大眾運輸人旅次分派到大眾運輸路網上，再利用模式反覆運算之方式，修正大眾運輸路網旅行時間，以反映真實之情況。同時在尖峰小時利用過飽和流量再分派之技術，以解決部份路段容量過飽和之不合理現象，而且由於公車路線是行駛於一般道路上，如果依照公車之承載率跟PCE當量因子，轉成小汽車當量數(PCU)，放到道路路網中，則會影響道路路網之交通量分派，因此，當考慮大眾運輸分派模組時應當考慮其與道路路網交通量分派間之影響，以下為DOTS I模式中兩個主要的運輸規劃分析軟體整理：

(1) TRANSPORT

TRANSPORT 為英國 HFA 所發展之道路路網運輸規劃套裝程式，軟體模式主要是架構於傳統運輸規劃模式上，而主要應用於中長期之運輸規劃。

TRANSPORT 模式可分為：

a、調查資料分析

- b、路網分析
- c、由交通量推估旅次矩陣
- d、旅次分佈與運具分配
- e、矩陣運算
- f、路網經濟效益評估
- g、繪圖

TRANSPORT 模式曾經應用在台北、台中、台南及桃園等都會區的整體規劃研究中。其本身並不提供旅次發生及大眾運輸分析模組，且其交通分區的容量限制在 500 分區，路網最大節線限制在 6000 個，為其主要缺點。

(2) TRANPLAN

TRANPLAN 為由美國 DKS Associates 所發展之運輸規劃套裝軟體。可同時運用在道路路網及大眾運輸路網，且兩路網之應用程式完全相容，在實際運用時主要以中長期規劃較適。雖然 TRANPLAN 未包含多運具選擇的新功能，但具備了傳統運輸規劃之四程序。

TRANPLAN 主要功能可分為：

- a、旅次發生
- b、旅次分佈與運具分配
- c、道路路網分析
- d、大眾運輸路網分析
- e、矩陣運算
- f、小區域分析
- g、路網分派
- h、路網繪圖
- i、選擇路段分析

TRANPLAN 模式曾應用在高雄、台北、台中、台南以及桃園都會區整體規劃研究報告中。TRANPLAN 本身同時兼具

道路路網分析功能及大眾運輸分析功能等兩種模組的功用，在傳統運輸規劃的四個步驟已相當完備，並提供許多附加功用程式，在交通分區及大眾運輸節線的限制上比較不受軟體的限制，但與電腦本身的硬體限制，可能對於執行的速度方面有較大影響。TRANPLAN 軟體在 DOTS I 模式中，主要被使用來進行大眾運輸路網運量分派。

2. UTPS

Urban Transportation Planning System Package(UTPS)套裝程式，是由美國聯邦政府運輸部都市大眾運輸管理局(UMTA)和聯邦公路總局(FHWA)於1970年代共同發展出來的一套都市運輸規劃多模式組合套裝程式，發展至今已作多次修正，目前由台北市政府委託中華民國運輸學會進行程式安裝與測試之引進工作，其引進的三個系統版本(MVS、CMS和PC)皆為1984年版本。

UTPS具有模擬多種運具網路的功能，可用來評估運輸策略的可行性，或運輸設施改善對整體運輸系統及週遭環境所造成的影響，從而選擇有利的方案。UTPS系統除應用於運輸規劃方面外，同時也使用於小區域分析、運輸走廊分析、或商業中心的交通分析。UTPS最大缺點乃是其資料的輸入及操作頗為困難，且極為複雜，使該模式的應用受到極大的限制。

1984年UTPS套裝程式，應用於一般運輸規劃過程共有25個主程式。依其功能可分為七大類，主要功能如下：

- (1) 土地使用規劃程式
- (2) 道路路網分析程式
- (3) 大眾運輸路網分析程式
- (4) 運輸需求預測模式
- (5) 資料處理運算程式
- (6) 綱要計畫程式
- (7) 系統公用程式

而其在處理大眾運輸路網分析程式方面又分：

(1) 大眾運輸路網建立程式：

道路路網和大眾運輸路網資料的讀取，並產生一描述大眾運輸路網的資料組。

(2) 大眾運輸路網路徑尋求程式：

決定分區間的最小阻抗大眾運輸路徑，計算分區內的大眾運輸費率、距離和阻抗。

(3) 大眾運輸路徑摘要程式：

分析分區內的大眾運輸路徑以決定這些路徑的分區間時分量(如等車時間，車上時間，轉車時間)。

(4) 大眾運輸旅次指派程式：

指派大眾運輸人旅次至大眾運輸路網，依路線、公司或運具形式等整理成表，另繪出附有旅次負載的大眾運輸路網。

(5) 大眾運輸車站間旅客流量分析程式：

產生大眾運輸站間的各旅次表和出入車站的旅次表，及依運具區分的出入站或大眾系統線上(line-haul)的旅次表。

(6) 大眾運輸費率建立程式：

由大眾運輸路徑檔和費率資料，產生一大眾運輸費率矩陣。

(7) 大眾運輸路網繪圖程式：

可繪出大眾運輸各路線圖，或將指派後的各路線旅客負載情形繪示於路線圖上。

3. MINUTP

MINUTP軟體是由美國COMSIS公司針對個人電腦需要，將原有大型電腦上的UTPS轉換成可於個人電腦上執行的運輸規劃軟體，適合短中長期之運輸規劃。

MINUTP軟體本身提供一系列之模組，各分析功能皆以模組撰寫，除了UTPS中旅次產生、旅次分佈、運具分配、交通量分派等四個模組外，仍提供了一些編輯功能，如矩陣運算、路網編修、

疊合及繪製等諸多功能。其分析結果亦能以圖形方式展現，但未能提供模式校估的功能，故部份回歸模式必須要先建成MINUTP的程式檔方可使用，其他如旅次分佈、運具分配之校估必須利用其他軟體來完成之。

MINUTP模式本身兼具道路及大眾運輸分析的功能，其大眾運輸路網運量分派模組無法與道路路網交通量分派模組回饋整合。

4. TransCAD

TransCAD軟體為美國Caliper公司於近年來發展出之運輸地理資訊系統(GIS)，是第一個結合地理資訊系統與運輸規劃功能之軟體。能將運輸資料經大量處理、運算而展示在地理空間上，協助決策者分析、瞭解規劃過程。此軟體應用於運輸規劃、路線選擇、公共設施區位選擇、交通事故分析、緊急事件處理、場站選擇、道路及運輸規劃、環境影響評估等。

TransCAD之運輸規劃模組包括旅次發生、旅次分佈、運具分配及交通量分派，並提供旅次分佈及運具分配之模式校估功能，其他分析功能與回歸調整功能必須利用Caliper語言自行撰寫之。

TransCAD 之指派方式，是將人旅次轉換為車旅次之區間交通量，依據使用者對於路線的選擇行為評準，再將車旅次指派於路網上，構成交通流量之實質空間分佈型態。【施鴻志等，1992、朱宏祥，1995】

一般而言，依運輸工具特性之不同，可分為兩大類型：

(1) 公路路網指派 (Highway network assignment)

為私人運輸工具的駕駛者在路網中路徑選擇的情況。

(2) 大眾運輸路網指派 (Transit network assignment)

為搭乘大眾運輸工具的乘客路線選擇之情況。

大眾運輸路網不同於一般道路路網，是因為其有固定的路線及班次。所以通常要作交通量指派時先做大眾運輸路網指派，其實大眾運輸路網指派的基本原則與公路路網相同，都在兩個交通

區間的交通量指派到最短時間或最少成本的路線上。

TransCAD所採用之指派方法為全有或全無指派法。原則為：用已知的公車OD矩陣代表公車旅次之需求，將各個不同OD起迄對之集合，依照成本最小化原則，求出大眾運輸路網之最小化成本路徑，然後將各OD起迄對的旅次數，全部指派至各自對應之路徑上，如果不是最短路徑，則完全不指派旅次。而各路段之總旅次為各OD起迄對指派至該路段旅次數總和。

5. UFOSNET

UFOSNET為美國RST International Inc.所發展出之交談視窗式之運輸規劃套裝軟體，也是唯一由華人研發之運輸地理資訊系統軟體。主要應用於旅次需求預測、容量及服務水準分析、大眾運輸路線規劃、運輸資訊系統發展、交通衝擊研究、及區位分析等。UFOSNET可同時應用在道路路網及大眾運輸路網，主要以中長期規劃較為適用。

UFOSNET主要功能如下：

- (1) 旅次發生
- (2) 旅次分佈與運具分配
- (3) 道路路網分析
- (4) 大眾運輸路網分析
- (5) 矩陣運算
- (6) 小區域分析
- (7) 交通量分派
- (8) 路網繪圖
- (9) 選擇路段分析

UFOSNET在模組功能上提供相當完備的分析模組，同時兼具道路及大眾運輸路網的分析功能，且在容量限制方面僅視電腦本身記憶體的限制，但未提供完整之模式校估功能，UFOSNET本身的圖形介面很強且操作簡單，讓使用者在操作過程中更容易了解研究地區的地理位置。

6. EMME/2

EMME/2為加拿大所發展出之運輸規劃軟體，可同時應用在道路路網及大眾運輸路網之上，在實際運用時主要以中長期規劃較為實用。

EMME/2主要功能有下列幾項：

- (1) 旅次發生
- (2) 旅次分佈
- (3) 運具分配
- (4) 道路路網分析
- (5) 大眾運輸路網分析

EMME/2在模組功能上具備完整的分析模組，同時兼有道路及大眾運輸雙重運輸路網之分析功能。

7. VISION

VISION軟體為德國PTV公司所發展之互動式運輸需求分析預測軟體，結合旅次活動鏈之觀念及大眾運輸多重路徑分派功能，並且具有GIS基本功能。

VISION主要包括VISEM與VISUM兩個模組，VISEM模式是將傳統的循序性運輸需求模式之旅次發生、旅次分佈及運具分配等三個步驟結合成一個整合模式，分別建立私人運輸與公共運輸的旅次矩陣。VISUM則是使用旅次分派的方法，分別評估運輸系統對於使用者、營運者及環境等之影響。

8. TP+

TP+是由加拿大Urban Analysis Group新近研發之運輸規劃軟體，可以處理道路及大眾運輸分派等，本身具有傳統運輸規劃基本分析功能外，並可與TRANPLAN與MINUTP等運輸規劃軟體格式相互運用。

在圖形展示方面，沒有提供強大之圖形介面，而是結合Viper軟體進行GIS連結及圖形展示編修之功能。

TP+軟體還提供外掛模組功能，擴大其使用之彈性，且可進

行路網疊圖等分析，提升分析的精度。TP+於模式參數校估方面之功能較缺乏，且並無法取得原始程式檔，因此在運輸規劃時，會面臨到本土化特性及調整模式參係數問題。



第二章 文獻回顧

表 2.1 常用運輸規劃軟體整理表

軟體名稱		DOTS I (TAIPEI+TRANSPORT +TRANSPPLAN)	UTPS	MINUTP	TransCAD	UFOSNET	EMME/2	VISION	TP+
曾應用在運輸規劃		台 北	台 北	嘉義、新竹	---	---	---	---	---
軟體內容	模組功能	<ul style="list-style-type: none"> ◆ 傳統運輸規劃程序完備，兼具道路及大眾運輸分析功能 ◆ 結合交通模擬分析功能兼具長短期運輸分析之工具 ◆ 包括旅次發生、初步運具分配、旅次處理、尖峰小時旅次矩陣處理 	<ul style="list-style-type: none"> ◆ 傳統運輸規劃程序完備，部分模式採個體理論 ◆ 與短期交通模擬分析軟體結合困難 	<ul style="list-style-type: none"> ◆ 傳統運輸規劃程序完備，兼具道路及大眾運輸分析功能，但大眾運輸分析模組功能較不足 ◆ 與短期交通模擬分析軟體結合困難 	<ul style="list-style-type: none"> ◆ 兼具道路及大眾運輸分析功能，但部份旅次分析及大眾運輸分析模組功能不完備 ◆ 需另執行交通模擬分析軟體，格式轉換易產生問題 	<ul style="list-style-type: none"> ◆ 傳統運輸規劃程序完備，兼具道路及大眾運輸分析功能 ◆ 需另執行交通模擬分析軟體，格式轉換易產生問題 	<ul style="list-style-type: none"> ◆ 傳統運輸規劃程序完備，兼具道路及大眾運輸分析功能 ◆ 需另執行交通模擬分析軟體，格式轉換易產生問題 	<ul style="list-style-type: none"> ◆ 兼具道路及大眾運輸分析功能，以旅次鏈活動模式分析為主，偏重個體行為分析 ◆ 需另執行交通模擬分析軟體，格式轉換易產生問題 	<ul style="list-style-type: none"> ◆ 兼具道路及大眾運輸分析功能，可與TRANPLAN、MINUTP 等其他規劃軟體格式相容 ◆ 需外掛 GIS 展示軟體
	如何處理大眾運輸	◆ 大眾運輸路網運量分派則運用全數分派法(All-or-Nothing Assignment)，將大眾運輸人旅次分派到大眾運輸路網上，再利用模式反覆運算之方式，修正大眾運輸路網旅行時間。	◆ 大眾運輸路網運量分派則運用全數分派法(All-or-Nothing Assignment)，將大眾運輸人旅次分派到大眾運輸路網上，再利用模式反覆運算之方式，修正大眾運輸路網旅行時間。	◆ 大眾運輸路網運量分派則運用全數分派法(All-or-Nothing Assignment)，將大眾運輸人旅次分派到大眾運輸路網上，再利用模式反覆運算之方式，修正大眾運輸路網旅行時間。	◆ 大眾運輸路網運量分派則運用全數分派法(All-or-Nothing Assignment)，將大眾運輸人旅次分派到大眾運輸路網上，再利用模式反覆運算之方式，修正大眾運輸路網旅行時間。				
軟體操作		須由 MS-DOS 系統下進行操作	<ul style="list-style-type: none"> ◆ 須由 MS-DOS 系統下進行操作 ◆ 具模組化設計 	<ul style="list-style-type: none"> ◆ 須由 MS-DOS 系統下進行操作 ◆ 具模組化設計 	<ul style="list-style-type: none"> ◆ Windows 作業系統進行操作 ◆ 具視窗界面 ◆ 圖形資料具編修功能 	<ul style="list-style-type: none"> ◆ Windows 作業系統進行操作 ◆ 具視窗界面 ◆ 圖形資料具編修功能 	<ul style="list-style-type: none"> ◆ Windows 作業系統進行操作 ◆ 具視窗界面 ◆ 圖形資料具編修功能 	<ul style="list-style-type: none"> ◆ Windows 作業系統進行操作 ◆ 具視窗界面 ◆ 圖形資料具編修功能 	<ul style="list-style-type: none"> ◆ Windows 作業系統進行操作 ◆ 具視窗界面 ◆ 圖形資料具編修功能
使用者介面		一般化輸出入操作視窗	需在 DOS 下執行	需在 DOS 下執行	一般化視窗畫面	一般化視窗畫面	一般化視窗畫面	一般化視窗畫面	一般化視窗畫面

資料來源：本研究整理(以亞聯台北都會區整體運輸規劃基本資料之調查與驗校(二)為基礎整理)

2.5 DynaTAIWAN 系統架構【胡大瀛，2005】

DynaTAIWAN系統初期參考美國公路聯邦總署(FHWA)所發展之兩套動態指派模式DYNASMART及DynaMIT，命名為DynaTAIWAN(Dynamic Traffic Assignment and Information in Wide Area Network)。

DynaTAIWAN整體系統架構包括模擬層與即時控制層，如圖2.5所示。模擬層為該系統之基礎，模擬產生之結果亦能提供系統外相關應用，整個模擬層之運作流程如圖2.6，其架構如圖2.7所示；即時控制層可提供模擬系統具有處理反應即時資訊影響之能力，交通模擬中，若能考量配合即時資訊來調整模擬，將使系統更具反應真實路網之特性。

DynaTAIWAN描述個體行為之決策反應下的依時流量變化情形，該系統模擬層主要針對車輛產生、車輛行進、交通背景資料、行前旅運決策以及途中旅運決策作相關模擬運作。並且依據「歷史資料庫」所提供之資料進行狀態初始化，以產生模擬背景資料，而背景資料之產生機制則可歸納為以下兩方面：

1. 需求面

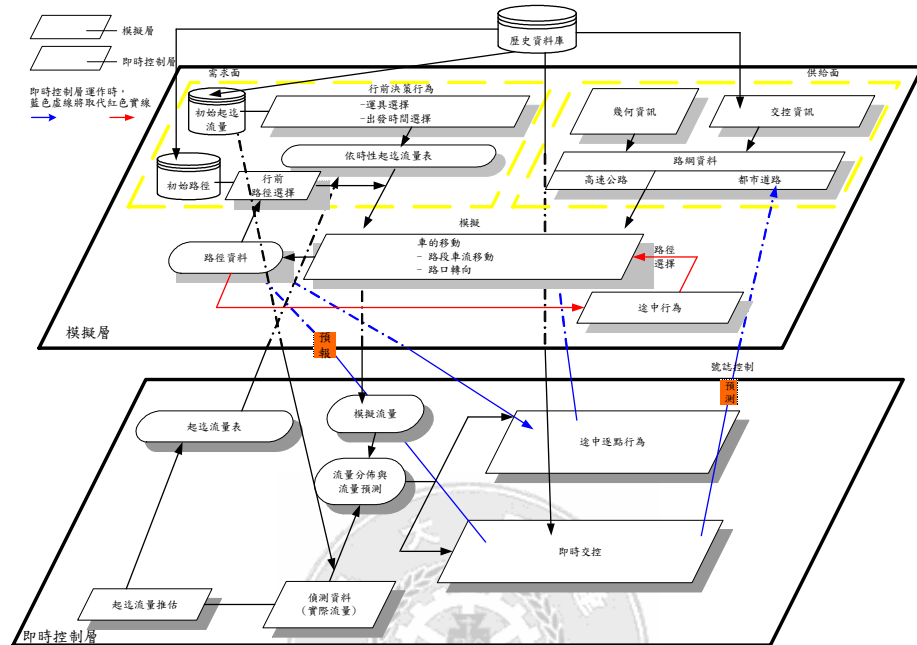
系統經由初始起迄流量需求資料，並藉由行前旅運行為決策機制產生各時段下各旅運者選擇運具之旅次量，透過行前路徑選擇模式產生各旅運者之初始路徑資量，即考量了「行前決策行為」機制。

2. 供給面

依據交通資訊產生路網之結構與屬性，路網分為高速公路路網以及一般道路路網兩種不同型態，並依道路型態設定不同之號誌。

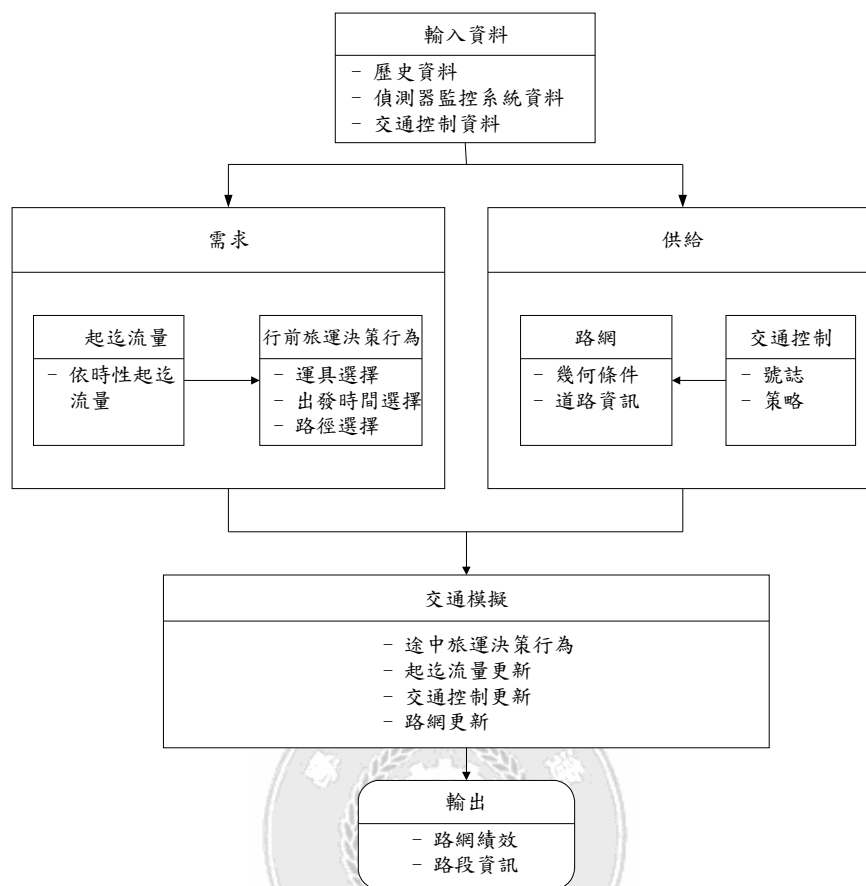
旅運者選擇之運具經由產生屬性進入路網後，依循該初始路徑於路段中往迄點移動，並給予速度屬性以進行移動，若行至路口，則系統會提供旅運者途中資訊(包含前方路段資訊、路徑資訊)，經由途中行為決策決定車輛是否接受新的路徑或是仍然選擇

原有路徑。而模擬過程中所產生之路徑資訊亦可用來作為未出發之旅運者之參考資訊。



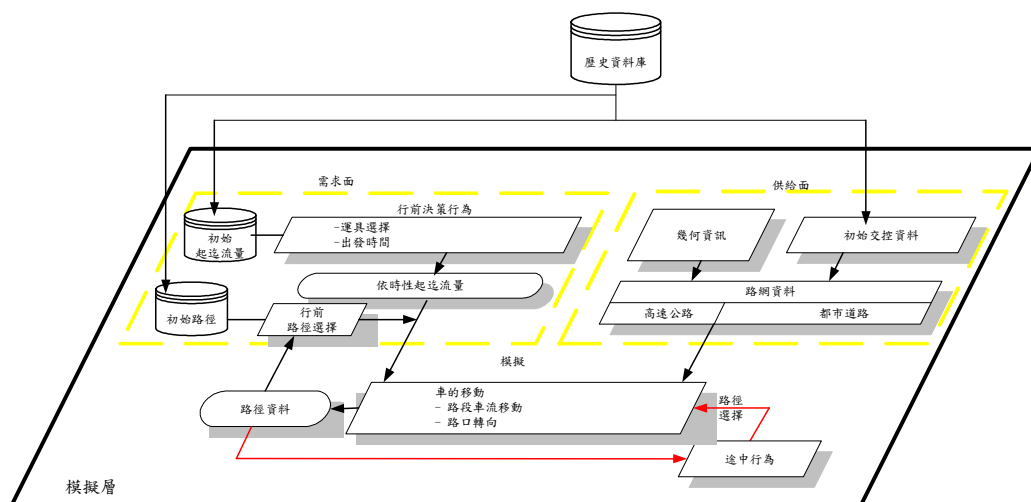
資料來源：胡大瀛等，2004

圖 2.5 DynaTAIWAN 系統架構圖



資料來源：胡大瀛等，2004

圖 2.6 系統模擬進行流程



資料來源：胡大瀛等，2004

圖 2.7 DynaTAIWAN 系統架構之模擬層



第三章 大眾運輸指派模型架構之建立

本章將於3.1節中介紹動態運輸規劃模型之基本觀念與架構，並討論各模組間之關係；3.2節就依時性大眾運輸均衡指派模型之運作流程進行說明；3.3節探討大眾運輸旅客指派；3.4節說明大眾運輸均衡指派模式。

3.1 動態運輸規劃架構

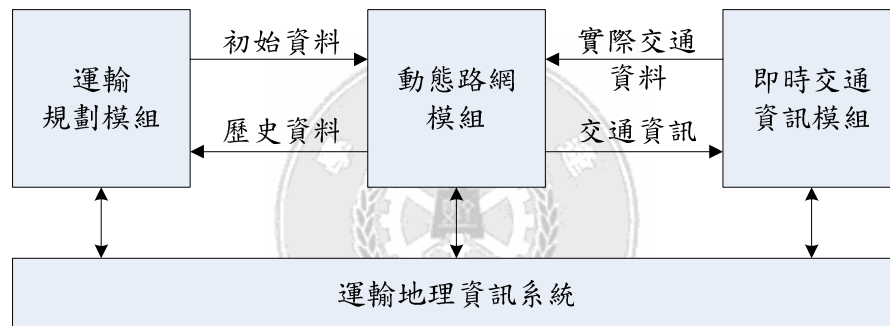
在傳統運輸規劃定義中，可以發現運輸規劃在本質上是一連串的作業過程，而且是一種持續性的作業程序。【王慶瑞，1996】。因此，運輸規劃基本上在相隔一段時間後必需針對前期規劃目標重新修正輸入資料並檢查規劃目標是否改變。也就是說整個運輸規劃程序應隨著時間的改變而變動，因此可能變動的部分包括：規劃時間點、規劃交通基本資料之內容、規劃的結果等。由於交通基本資料的不容易更新取得，且每隔一段時間後要更新交通基本資料之內容必須花費大量金錢與人力，所以持續性的作業程序不容易達到。

為了解決更新交通基本資料之問題，因而提出動態運輸規劃模型，模型分為三個部份(如圖3.1所示)，可分成傳統運輸規劃模組、動態路網指派模組及即時交通資訊等三個模組，動態路網指派模組與即時交通資訊模組係擔任即時性交通控制之工作，由即時交通資訊模組所提供之實際偵測到之交通量資料，並結合運輸規劃模組所產生之初始OD資料，能給予動態路網模組產生依時性OD資料並進行模擬分析，以提供管理策略給即時交通資訊模組使用，並彙整交通基本資料成歷史資料供運輸規劃模組使用。

因此，可知即時交通資訊模組可解決交通基本資料的更新問題，交通基本資料在即時交通資訊模組中隨時間被偵測，並被記錄成為歷史資料儲存於運輸規劃模組中隨時可供規劃人員使用，因此規劃人員可在任何時間點取用歷史交通基本資料進行檢測規劃目標是否改變、是否達到預期的規劃成果。

透過此三個模組之結合後，運輸規劃人員可在任一時間點進行持續性運輸規劃工作，因為即時性交控系統的歷史資料被累積，可使運輸規劃模組總計彙整分析交通基本資料，使短期之運輸規劃推估預測資料得到適度之修正，讓使用者得到足夠的交通資訊，獲得最佳路線。

經由上述之介紹，可以得知運輸規劃模組、動態路網指派模組及即時交通資訊模組間之資料傳遞係密不可分的。動態路網指派模組與即時交通資訊模組可在交通管理單位設置即時交通分析系統，因此便可不斷獲得即時交通資訊。而運輸規劃模組即可從這些資料獲得所需資料，如旅行時間及OD資料等供分析使用。



資料來源：陳文能，2003

圖 3.1 動態運輸規劃架構圖

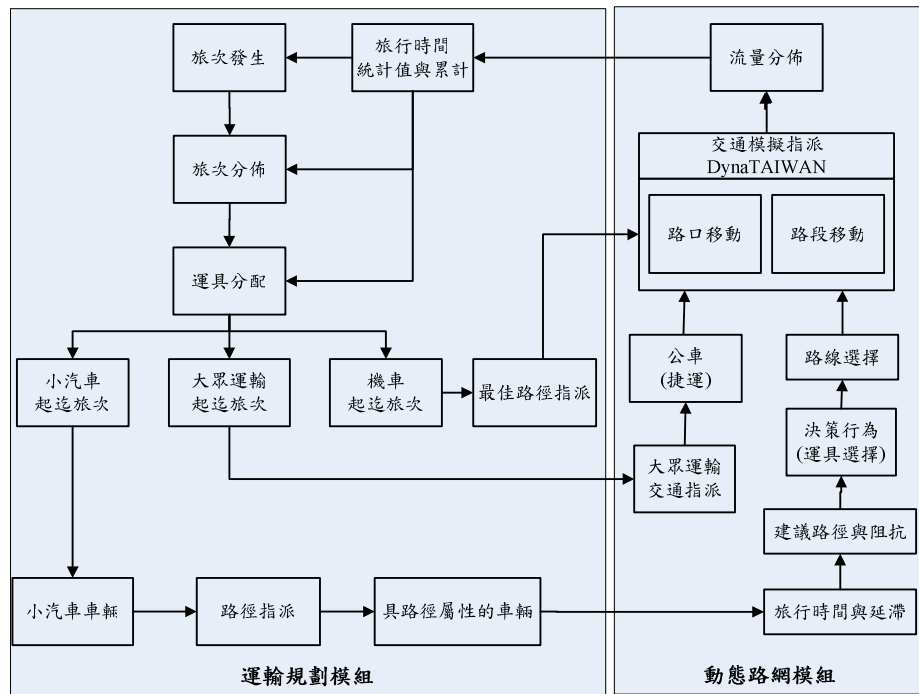
3.2 依時性大眾運輸指派模型運作流程

本節將探討依時性大眾運輸指派模型之運作流程，此運作流程係依據圖3.1之運輸規劃模組及動態路網模組而來，其中動態路網模組在依時性OD變化下，進行路網中車流分佈的交通管理、控制策略之建議，並分析動態路網模組、運輸規劃模組與即時交通資訊模組資料傳遞之內容與方式。由下圖3.2可以知道運輸規劃模組首先依據傳統之運輸規劃程序，從旅次發生、旅次分佈及運具分配，但於此過程本研究考量了旅行時間與統計值之累計，此資料是從動態運輸規劃模組所產生之結果，於運具分配後將產生三種輸出，此三種輸出代表各種運輸工具之起迄旅次表，包括大眾

運輸OD旅次、小汽車OD旅次及機車之OD旅次，於【陳文能，2003】其小汽車起迄旅次包括將旅次轉成小汽車車輛，利用小汽車車輛配合路徑指派，將產生之具有路徑屬性之車輛輸入至動態模擬指派模組中進行模擬，本研究著重於大眾運輸OD旅次表，經過大眾運輸指派將旅次轉成公車車輛進行動態模擬，而機車之OD旅次量經過最佳路徑指派後與大眾運輸、小汽車所產生之流量全部回饋至DynaTAIWAN中進行模擬，即可輸出流量之分佈。最後將此流量分佈資料回饋至運輸規劃模組中，以重新調整旅次產生、旅次分佈及運具選擇。

有了上述流程之後，便可結合即時交通資訊模組之道路流量偵測所獲得之流量資料(依時性OD)，將OD資料放入動態路網指派模組(本研究採用DynaTAIWAN模擬軟體)中運作，即可獲得即時性之交通控制策略與資訊提供給使用者參考，並可累積交通資料日後可供交通規劃人員使用。而大眾運輸使用者亦可透過動態路網模組獲得路網上即時資訊，進而選擇所需的大眾運輸工具。

因此，當大眾運輸停靠站發生擁擠現象時，動態路網模組就能發揮其功能，提供資訊給大眾運輸使用者，而使用者就能依據當時的情況決定是否改變其原本搭乘方式。



資料來源：本研究整理

圖 3.2 依時性大眾運輸指派模型架構

3.3 大眾運輸旅客指派

一般大眾運輸旅客指派問題皆會先給定一個起迄點之流量，之後再依據最小化旅行時間或成本，將乘客流量指派至路線上。而如何決定最小成本之路線為大眾運輸乘客指派之重要因素，其中成本屬性包括：最小成本路線、大眾運輸路網之容量限制與擁擠影響等。

而當產生最小成本問題時大多採用傳統之最短路徑演算法來解決。近年來常用來考量旅行時間之方法包括隨機的及依時性的描述，其中乘客到達、車輛到達及旅行時間皆有可能為隨機的。因此，這些隨機的過程對於路線指派之結果會有很大之影響。而本研究用來考量旅行時間之方法將採用依時性的因素，本研究將假設一個適當的班表，則一般化成本函數之車上旅行時間將可由此班表而來，則最小成本之路徑即可被當成依時性最短路徑問題。

最後，大眾運輸指派所會遇到之問題為大眾運輸之容量限

制，在大眾運輸指派一開始的發展大多忽略此因素，使得指派結果不適用在所有之情況。因為，當考慮容量限制時，乘客可能就不能搭乘第一部到達停靠站之車輛，因此就會影響到乘客之等車時間。本研究於下分別討論不考量擁擠影響與考量擁擠影響之大眾運輸指派以及不考量大眾運輸容量限制與考量容量限制之大眾運輸指派。

3.3.1 不考量擁擠影響之大眾運輸指派

如果大眾運輸指派不考量擁擠影響，當停靠站發生擁擠現象時，則將不考量乘客於停靠站受到擁擠影響所增加之等車時間，一開始對於不考量擁擠影響之指派皆會使用最短路徑法，將起迄點之需求全部指派至最短路徑上，則一般成本函數如下表示：

$$c_{ij} = t_{iju} + t_{ijw} + t_{ijv} + t_{ijx} \quad (3.3-1)$$

其中，

t_{iju} ：到達i區車站及離開j區車站之步行時間，

t_{ijw} ：在i區車站之等車時間，

t_{ijv} ：從i區到j區之車上旅行時間，

t_{ijx} ：從i區到j區路徑中之轉車時間，

3.3.2 考量擁擠影響之大眾運輸指派

如果大眾運輸指派考量擁擠影響，則將考量乘客於停靠站受到擁擠影響所增加之等車時間，由於大眾運輸停靠站為一個複雜之排隊等候(queueing)系統。乘客到達停靠站之後將會搭乘通過此停靠站之大眾運輸車輛。因此，必須考量到停靠站之不同排隊等候。

一般而言，在停靠站之等車時間將依據停靠站之總乘客數決定。因此，等車時間將隨著大眾運輸使用者之流量遞增，且將假設一般成本函數為：

$$c_{ij} = t_{iju} + t_{ijw} + t_{ijv} + t_{ijx} + \phi_{ij} \quad (3.3-2)$$

其中，

$$\varphi_{ij} = V_{ij} + \tilde{V}_{ij} / K,$$

V_{ij} ：路段ij之停靠站所新增之乘客數，

\tilde{V}_{ij} ：停靠站上因容量限制而剩餘下來之乘客數，

K ：大眾運輸容量。

而上述成本函數之最後一項考量了擁擠影響下之等車時間。當新增之乘客數加上剩餘下來之乘客數大於或等於大眾運輸容量，則 φ_{ij} 等於等車時間；當新增之乘客數加上剩餘下來之乘客數小於大眾運輸容量，則 φ_{ij} 等於0。

3.3.3 不考量大眾運輸容量限制之大眾運輸指派

當大眾運輸不考量容量限制時，將假設大眾運輸能服務路網上全部的乘客，但實際上當需求增加時，某些大眾運輸會有超載之情況，將產生不合理的結果，但此方法為較簡單的。因此，將不考量加班車的加入，大眾運輸系統將依據原本之班表運行。

3.3.4 考量大眾運輸容量限制之大眾運輸指派

而當大眾運輸考量容量限制時，如果乘客在停靠站之人數接近大眾運輸容量時，等車時間將會增加，而當乘客數大於大眾運輸容量時將考慮加派加班車來滿足需求，因此，當考量容量限制時，需考量加入加班車之後旅行時間之改變。

3.4 均衡指派模式

一般在大眾運輸路網中，乘客於任何兩停靠站之間可以有許多條大眾運輸路線供選擇，假設乘客會選擇使得其旅行成本最小之路線。而大眾運輸皆有容量限制，因此，在考量旅行成本時尚需考量到大眾運輸之容量限制，因為當停靠站之乘客人數增加時等車時間也會增加，進而影響到旅行成本。而當某些停靠站變得擁擠時，乘客將考量使用另一條大眾運輸路線。

本研究將假設大眾運輸路網的擁擠現象大多發生在大眾運輸

的停靠站。而乘客的等車時間是依據大眾運輸容量限制跟乘客數來決定。所以，當乘客數接近大眾運輸容量時，等車時間會增加。而且乘客將考量最小成本的路線，且將搭乘第一部到達停靠站的大眾運輸車輛，如果大眾運輸容量已滿載時，乘客將被指派至另一條大眾運輸路線，以達到路網均衡情況。

而在此節，本研究將分析大眾運輸均衡指派問題之理論基礎，均衡指派法之使用者均衡，可以數學式表示如 3.4-1 式：

$$C_{pij} \begin{cases} = C_{ij}^*, \forall T_{pij}^* > 0 \\ \geq C_{ij}^*, \forall T_{pij}^* = 0 \end{cases} \quad (3.4-1)$$

其中，

C_{pij} ：由 i 區到 j 區使用路徑 p 之旅行成本，

C_{ij}^* ：由 i 區到 j 區之最小旅行成本，

T_{pij}^* ：由 i 區到 j 區使用路徑 p 之交通量。

由於路線上之旅行成本係交通流量之函數，基於使用者皆追求最小化旅行成本，上式可以一非線性數學規劃模式表示如 3.4-2 式，其目標函數為所有交通流量在各路段之旅行成本總和，當目標函數為最小時，即可達成使用者均衡。

$$\begin{aligned} \text{Min} Z &= \sum_a \int_0^{V_a} c_a(V) dV \\ \text{S.T.} \quad &\sum_{pij} T_{pij} = T_{ij} \\ &T_{pij} \geq 0 \end{aligned} \quad (3.4-2)$$

其中，

V_a ：為路段 a 之交通流量，

$c_a(V)$ ：路段 a 交通量為 V 時，相對之旅行成本。

為證實 (3.5-2) 足以代表使用者均衡，由先前之定義如 3.4-3 式可知：

$$V_a = \sum_p \sum_i \sum_j \delta_{pij}^a T_{pij} \quad (3.4-3)$$

其中，

$\delta_{pij}^a = 1$ ，當路段 a 在 i 至 j 之路徑 p 上時，

$\delta_{pij}^a = 0$ ，其他。



第四章 大眾運輸指派流程建立

本章針對大眾運輸車輛指派實驗路網建立的程序進行說明。4.1節說明大眾運輸路網建立；4.2節說明大眾運輸指派流程；4.3節說明本研究之大眾運輸指派之程式架構；4.4節進行DynaTAIWAN公車模擬之功能測試。

4.1 大眾運輸路網構建

大眾運輸路網主要係模擬真實路網，而大眾運輸路網之組成主要包括節點、區中心、路段及大眾運輸路線等，本研究於大眾運輸路網中設計五條大眾運輸路線，所構建之大眾運輸路網圖如圖4.1所示。而路網構建過程為首先於一般道路路網上設計五條公車路線，此五條公車路線分別服務不同的停靠站，假設停靠站皆設置於道路路網節點上，由於所設計之大眾運輸路線停靠站將會與一般道路路網連結，因此，於停靠站與道路路網之關係中，假設有一條dummy link相互連結。而在同一個路段上，可能會有兩條以上之大眾運輸通過，因此會有轉車之情形發生，故於轉車之停靠站皆設有可供轉車之dummy link。

1. 節點

一個節點代表一個交叉路口，或是一個供旅客上下車的停靠站。節點的屬性資料為節點代號。本研究以符號 \bigcirc 表示節點。

2. 區中心

區中心代表旅次的起迄點，在路網中，可視為同一個節點。每一個區中心至少要有一條路段連接到大眾運輸停靠站及公路路網的節點，以確保旅運需求可以使用公路路網或大眾運輸路網。而大眾運輸路網的路段通常採用步行的方式。其屬性資料為代號。本研究以符號 \bigcirc 表示區中心。

3. 節線

節線是連接兩節點的路段，每一條節線都包含一個起始節點及一個終止節點，其屬性資料尚有節線長度。本研究以符號——表示節線。

4. 大眾運輸路線

大眾運輸路線包括許多連續的停靠站還有行車時刻表。每一條大眾運輸路線可以包括若干條次路線，不同的行車時刻表即可視為一條次路線，而每條次路線的資料為次路線代號、路線之行車路徑。本研究所假設之5條大眾運輸路線如下：

1. 2→1→7→8→14→15→9→3→4→5→44

2. 3→2→8→7→13→19→25→31→35→37

3. 7→13→19→25→26→27→28→29→23→17

4. 32→35→37→36→33→34→30→24→18


5. 35→36→33→28→29→30→24→23→17→11→12→6

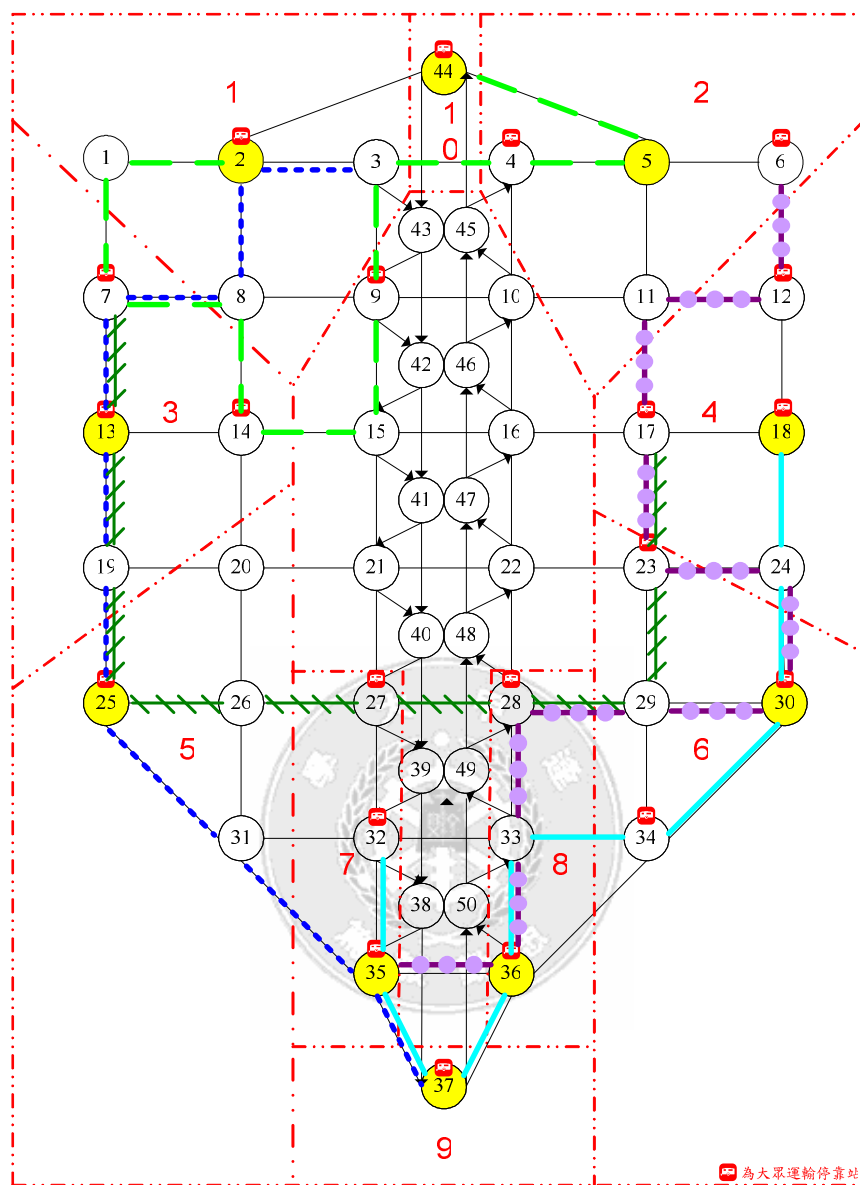
5. 大眾運輸停靠站

由於大眾運輸路線上的各個節點不一定皆能提供旅客上車之服務，因此，本研究於大眾運輸路線上假設了大眾運輸之停靠站。

本研究以符號表示停靠站。

6. Dummy link

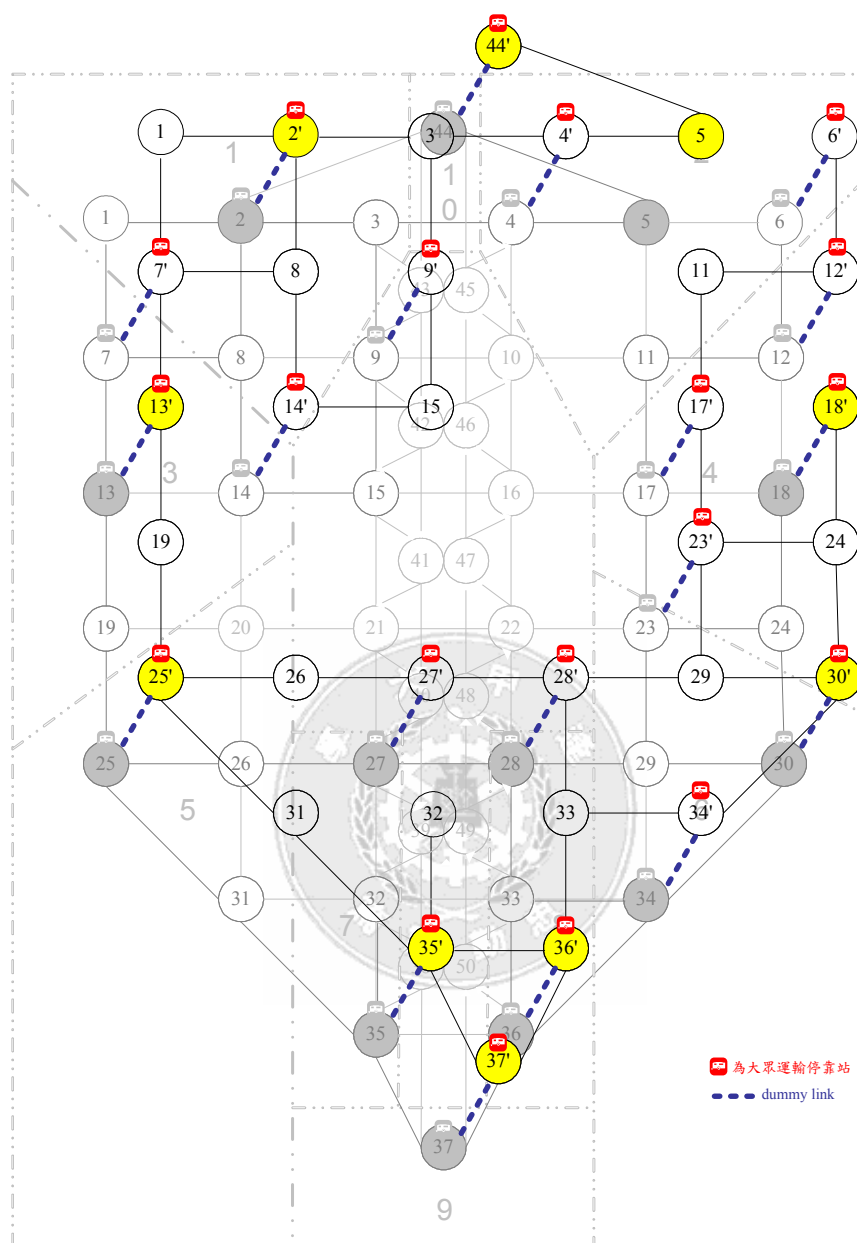
由於本研究所設計之大眾運輸路線停靠站將會與一般道路路網連結，因此，於停靠站與道路路網之關係中，本研究假設有一條dummy link相互連結，如圖4.2 所示。而在同一個路段上，可能會有兩條以上之大眾運輸通過，因此會有轉車之情形發生，故本研究於轉車之停靠站皆設有可供轉車之dummy link，如圖4.3 所示。本研究以符號表示dummy link。



—	路網(節線)	⊙	節點
	分區之區塊	●	區中心
	路線一		路線二
	路線三		路線四
	路線五		停靠站

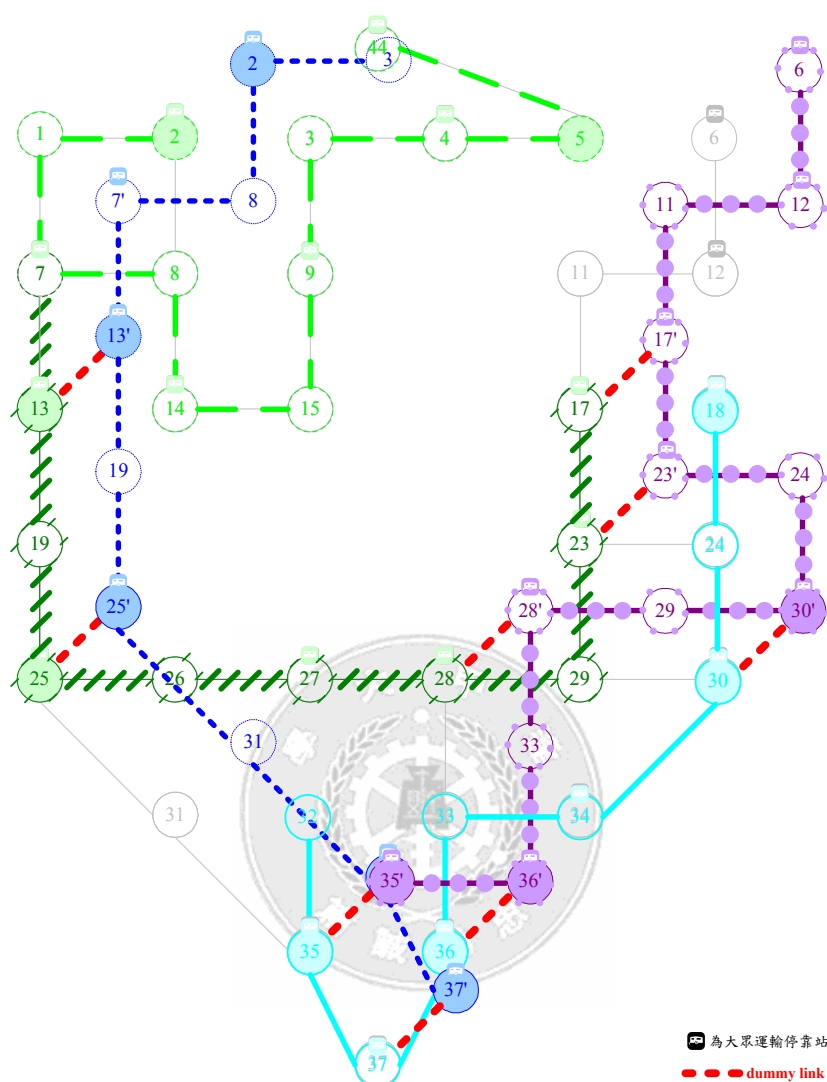
資料來源：本研究整理

圖 4.1 大眾運輸路網圖



資料來源：本研究整理

圖 4.2 停靠站與道路路網之關係圖



資料來源：本研究整理

圖 4.3 大眾運輸路網轉車圖

4.2 大眾運輸指派流程

大眾運輸旅次指派首要工作即利用已建構之大眾運輸路網，計算各起迄點間之旅行成本，產生各起迄點間之最短路徑，最後再進行旅次指派，而指派原則可分為將OD點之間的交通量全部指派到起迄點之間的最短路上的全有全無指派(All-or-Nothing)、在每一階段指派一定百分比的交通需求且每次指派是基於全有全無指派法。每指派一次之後，旅行時間要根據當前交通量重新計算

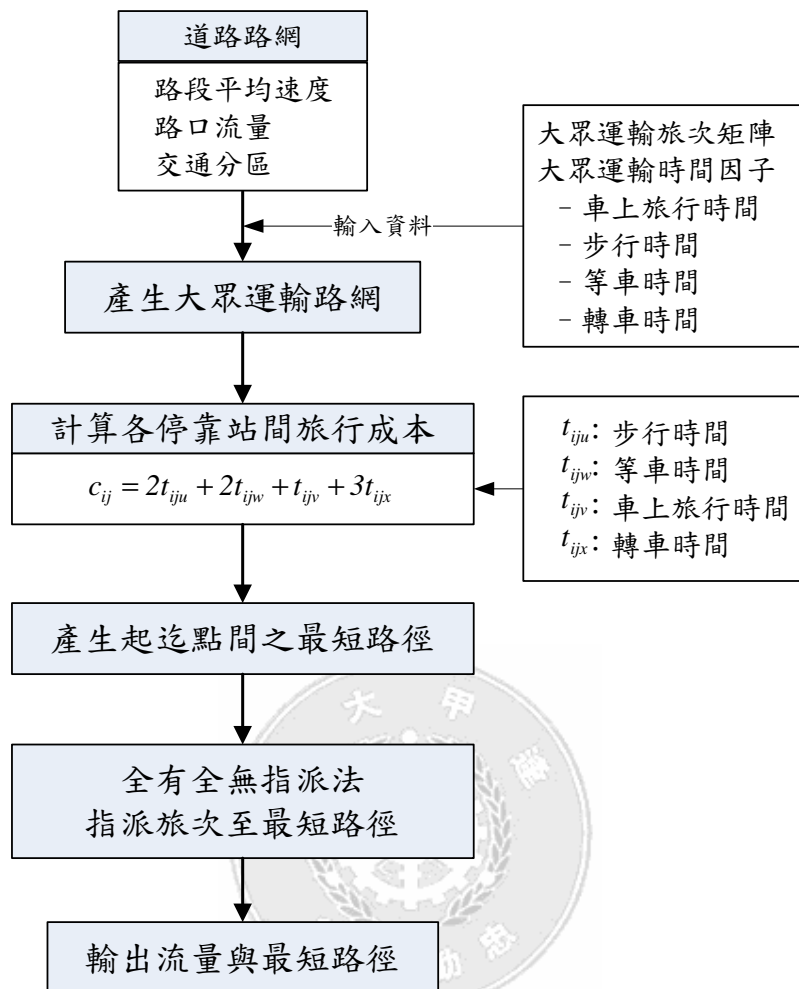
的增輛指派法(Incremental Assignment)、反覆的採用全有全無指派且根據一個反映路段容量的擁擠函數反覆的計算路段旅行時間的容量限制(Capacity Restraint)、採用一個反覆的過程來得到一個收斂解，在此方法中旅行者不能通過改變路線來改變旅行時間的使用者均衡(User Equilibrium)等，本研究以全有全無指派進行旅次指派，其中大眾運輸路網之格式如附錄一所示，而本研究之大眾運輸指派之流程圖如圖4.4所示，首先讀入大眾運輸路網資料，包括大眾運輸旅次矩陣、大眾運輸路網資料、大眾運輸之車上旅行時間、步行時間、等車時間、轉車時間、公車追蹤檔及交通分區等資料，其中時間權重之設定係參酌交通部運研所、亞聯顧問公司及台北市政府捷運局等研究報告整理而成，如表4.1所示，假設車上旅行時間權重為1、步行時間權重為2、等車時間權重為2、轉車時間權重為3，將大眾運輸路網資料輸入後利用一般成本函數計算各停靠站之旅行成本，進而產生起迄點間之最短路徑，之後以全有全無指派方法將旅次指派至最短路徑上，其輸出結果為大眾運輸流量與最短路徑等資料。

且在指派前本研究假設各分區之旅次均發生在分區中心上，再依據分區中心至鄰近停靠站之步行時間，將旅次分配到各大眾運輸路線進行站與站間的指派工作。

表4.1 時間權重整理表

參考文獻	車上時間 權重	步行時間 權重	等車時間 權重	轉車時間 權重
交通部運研所	1	2	2	每次轉車 多6分鐘
亞聯顧問公司	1	2	2	每次轉車 多6分鐘
台北市政府捷運局	1	2	2	每次轉車 多6分鐘

資料來源：本研究整理



資料來源：本研究整理

圖 4.4 大眾運輸指派之流程圖

4.3 程式架構

本節針對本研究所撰寫與使用之程式架構進行說明，大眾運輸指派程式一開始會先讀入OD資料及大眾運輸路網資料進行最短路徑指派，之後將大眾運輸指派程式所產生之結果回饋至DynaTAIWAN模擬程式中進行模擬，模擬完成即輸出旅行時間資料，將此旅行時間資料再回饋至大眾運輸指派程式，一直反覆運算。本研究之程式如附錄二所示。

本研究所使用的程式架構如圖4.5 所示，在程式架構上分成了3個主要的部分，分別為大眾運輸指派模組、交通模擬模組

(DynaTAIWAN)，以及輸入、輸出處理。整個程式架構中各模組之間的關係可以分為以下4個部份來說明：

1. 輸入處理

在程式的初始化會載入所需的相關資料，包含大眾運輸旅次矩陣、大眾運輸路網、路段平均速度及交通分區等資料。

2. 大眾運輸指派模組

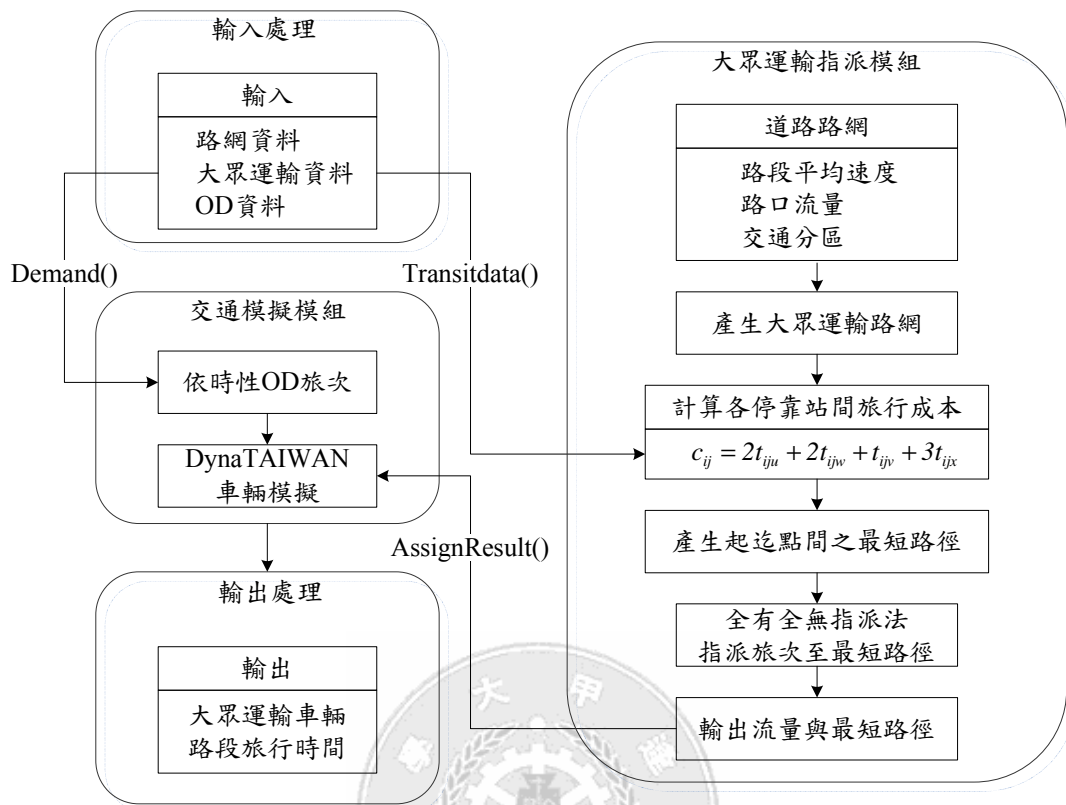
在交通模擬模組進行模擬前，大眾運輸指派模組將依據 Dijkstra's algorithm 找出最少花費之路徑，並計算出最短路徑、路段 OD 量、OD 總量、最短路徑花費總時間，並輸出成文字檔。大眾運輸車輛指派程式執行之畫面如圖 4.6 所示，此時系統會先要求使用者輸入情境之後才進行模擬。

3. 交通模擬模組

交通模擬模組在取得依時性 OD 資料後，根據每一需求時段的 OD 量產生一般車輛，將一般車輛與大眾運輸車輛於路網中進行交通模擬。

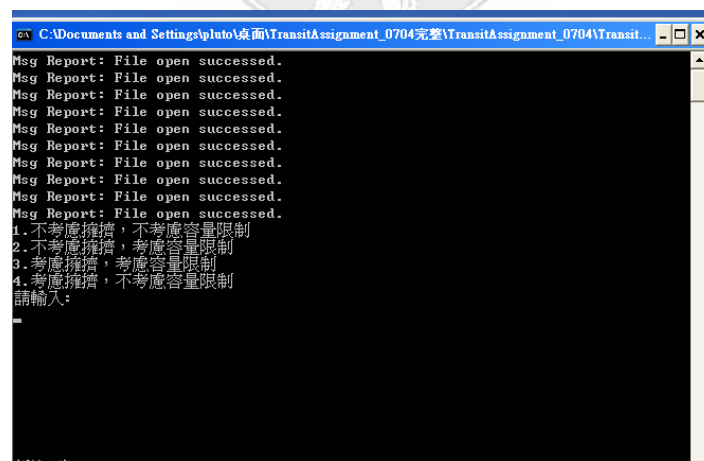
4. 輸出處理

進行最短路徑指派後，將大眾運輸指派程式所產生之結果回饋至 DynaTAIWAN 模擬程式中進行模擬，模擬完成即輸出旅行時間資料，將此旅行時間資料再回饋至大眾運輸指派程式，一直反覆運算。



資料來源：本研究整理

圖 4.5 程式架構圖



資料來源：本研究整理

圖 4.6 大眾運輸車輛指派程式執行之畫面

4.4 公車模擬功能測試

目前DynaTAIWAN中的大眾運輸車輛模擬功能，必須由使用者自行輸入路線資料，每一部大眾運輸車輛必須指定一組路線資料，因此在此測試中設定一條大眾運輸路線與一般車輛進行模擬，其輸入格式如表4.2所示，包含了公車數目、公車ID#、產生路段之起點節點編號、產生路段之迄點節點編號、公車發車時間、該公車每站停靠時間、該條公車路線所經節點數目、所經節點編號及停靠方式。

表4.2 大眾運輸車輛輸入檔

公車數目	公車ID#		車輛產生起點編號	車輛產生迄點編號	發車時間(分)	每站停靠時間(秒)		所經節點數
20	37		32	35	0.0	30		8
所經節點編號	35	37	36	33	34	30	24	18
停靠方式	1	1	1	1	1	1	1	1
	0：不停							
	1：停靠轉入路口近端							
	2：停靠路段中段							
	3：停靠轉入路口遠端							
	4：停靠於公車彎							

資料來源：本研究整理

完整模擬結束後可得到大眾運輸車輛的追蹤檔，如表4.3所示，大眾運輸車輛追蹤檔內記錄了包含車輛種類(Veh_type)、車輛編號(Veh_ID)、是否註記(tag)、是否能接收資訊(info)、產生之路段上游節點編號(From_node)、車輛起點(Origin)、車輛迄點(Destination)、出發時間(Start_time)、車輛進入路網時間(Enter_time)、抵達迄點時間(Arrival_time)、路線節點數目(PathNodes_#)、實際通過節點數目(Nodes_#)、總旅行時間(Total

Travel Time)、路徑節點編號、經過各節點之時間點、通過各路段節線之旅行時間及各節點停靠時間。

表 4.3 大眾運輸車輛追蹤檔

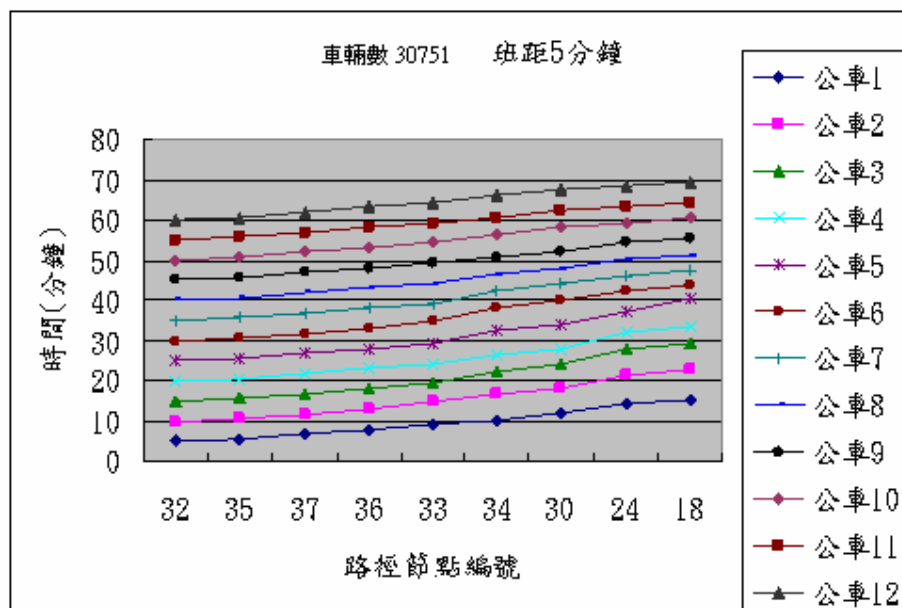
class#	ID	tag	inf	From	O	D	ST	ET	AT	PNod#	Nod#	TT
12	4	0	0	32	32	18	0	0.5	13.79	9	9	17.29
路徑節點編號		經過各節點時間點(分)		通過各路段節線旅行時間(分)		該車輛於各路段節線等車時間(分)						
32												
35		2.00				1.50		1.18				
37		3.15				1.15		0.50				
36		4.50				1.35		1.25				
33		6.50				2.00		1.39				
34		8.50				2.00		1.31				
30		10.50				2.00		1.10				
24		12.50				2.00		1.25				
18		13.79				1.29		0.50				

資料來源：本研究整理

於DynaTAIWAN模擬大眾運輸後，將大眾運輸車輛所經過之各節點時間及通過各節點之編號，繪製成時間與路徑關係的時空圖，如圖4.7、4.8、4.9所示，X軸為大眾運輸車輛所行經之節點編號，Y軸則為時間點，由圖可知，於班距分別為5、10及15分鐘時，車輛數為30751時，大眾運輸車輛於每個停靠站停等30秒所產生之時空圖。假設模擬一小時之結果，大眾運輸班距為5分鐘時，將會產生12部公車，班距10分鐘產生6部公車，班距15分時產生4部公車，每部公車經過各節點的時間點皆被記錄下來。

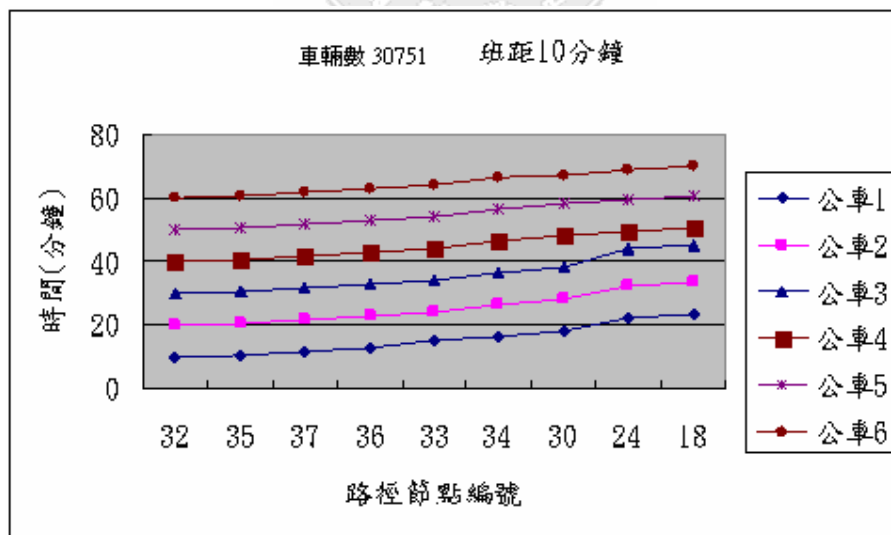
若以班距5分鐘為例，12部公車皆從點32出發，因此出發時間點分別被紀錄為5、10、15、20...60分鐘，且如圖4.8所示每台車之發車間距皆為5分鐘，而當第一部公車行經點35時其時間點被紀錄為6.8分鐘，當行經點37時其時間點被紀錄為8.1分鐘，因此，即可求得點35至點37之旅行時間為1.3分鐘。又以班距15分鐘為例，可以發現公車於每個停靠站皆會停等幾秒鐘。所以，由此測

試結果可知DynaTAIWAN具備動態模擬功能，故其大眾運輸車輛功能為本研究所適用。



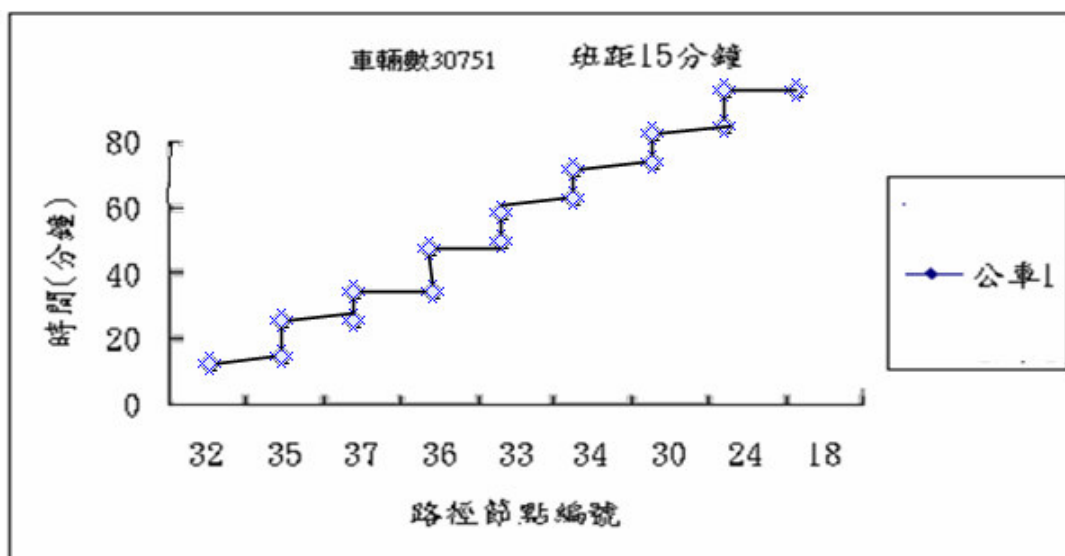
資料來源：本研究整理

圖 4.7 大眾運輸車輛於班距5分鐘之時空圖



資料來源：本研究整理

圖 4.8 大眾運輸車輛於班距10分鐘之時空圖



資料來源：本研究整理

圖4.9 大眾運輸車輛於班距15分鐘之時空圖



第五章 數值實驗

為驗證本研究所設計之大眾運輸指派模式架構適用性，本章將以數值實驗方式進行整體模式測試，以瞭解本研究所構建的指派模式之適用性。5.1節說明50節點中型路網及情境設定；5.2節將進行本研究之初步數值實驗測試；5.3節進行敏感度分析測試；5.4節進行大眾運輸均衡指派測試。

5.1 50 節點中型路網

50節點路網主要係模擬真實路網，該路網由172條節線、50個節點及10個分區所組成，除市區道路外，另設置一條高速公路，並以匝道聯繫市區道路，市區道路與高速公路路段長度皆設為500公尺，匝道路段長度設為250公尺，路網中的號誌型態包含無號誌及定時號誌型態，50節點路網屬性設定值如表5.1所示。

表5.1 50節點路網屬性設定值

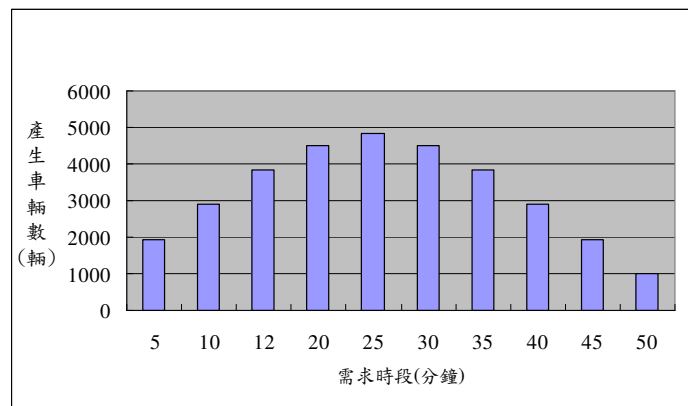
屬性	項目	假設值
節線屬性	長度	高速公路：500 公尺 一般都市：500 公尺 匝道：250 公尺
	速限	高速公路：100 公里/小時 上匝道：40 公里/小時 下匝道：40 公里/小時 市區道路(型態 6)：50 公里/小時
	道路屬性	高速公路 上匝道 下匝道 市區道路(有中央無快慢)
	飽合流率	0.5 veh/sec
背景流量設定 (起迄點流量需求)	分區至路網中 其他分區	50 部/每 5 分鐘
號誌設定		無號誌 定時號誌

資料來源：本研究整理

本研究實驗設計方法為透過實驗因子設計不同條件的實驗，因此以下針對實驗設計時，所考慮的情境因子進行說明：

假設50節點路網車輛需求產生型態為常態分佈型態。設定車輛的需求產生型態呈常態，考慮了現實交通的變化及其尖離峰特性，如圖5.1所示，主要探討當交通路網在明顯的擁擠情況下，大眾運輸車輛於路網中變化。總需求時段為10個時段，共連續產生50分鐘，而測試車種包含小汽車、機車及公車。

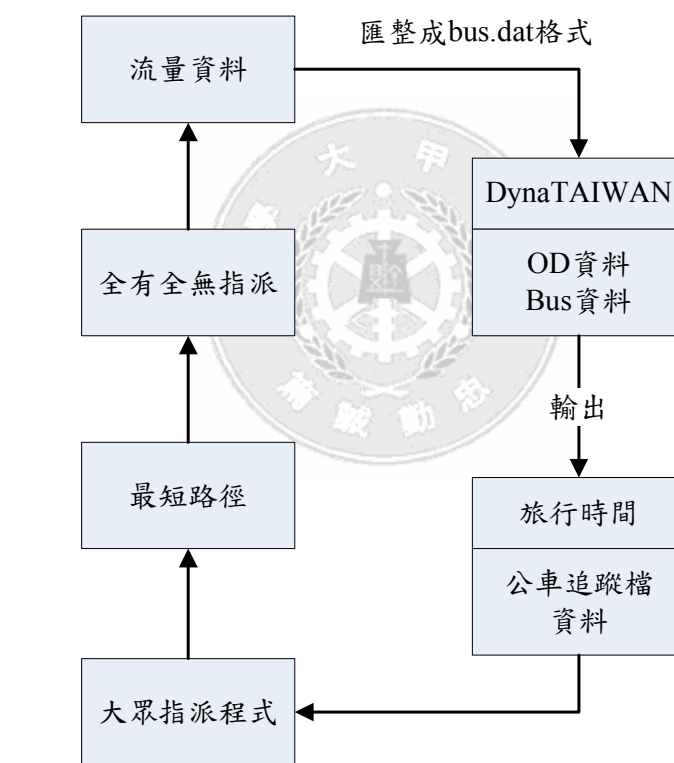
由於本研究希望能觀察路網於擁擠時之流量變化與影響，故增量因子則設定為0.7，增量因子為DynaTAIWAN控制路網中車輛數方式，其功能主要為：在一個固定交通路網下，不同的車輛數呈現出不同的交通狀況。當路網中車輛數越多，交通壅塞狀況越容易產生；車輛數越少時，路網中車流則會較接近自由車流狀況。因此，透過增量因子改變可以觀察路網中車流量分佈與容量。增量因子主要目的為控制路網中車輛數，以觀察於不同增量因子下車流量之變化與影響。透過需求增量因子的設定，可以將整個原始車輛OD以及機車OD資料直接作倍數的放大或縮小，以利直接增減需求量進行數值實驗進行比較。增量因子0.7時汽車產生30,751輛，機車產生30,611輛，合計產生61,362輛，其平均旅行時間為10.01分鐘，平均旅行距離為2.46公里，平均停等時間為5.62分鐘。



資料來源：本研究整理

圖5.1 車輛需求產生型態

本研究首先於50節點上設計5條公車路線，每條路線有其服務的停靠站，且給予這5條路線初始的班表，首先執行DynaTAIWAN，將產生輸出檔如：大眾運輸通過各路段之旅行時間。之後再將DynaTAIWAN之輸出資料與大眾運輸OD資料輸入至大眾運輸指派程式中，作初始之成本計算，成本算完之後將會產生最短路徑，並產生最短路徑所對應之輸出資料，而將最短路徑所經過的公車路線資料轉化成DynaTAIWAN之公車輸入格式之後回饋至DynaTAIWAN進行模擬，本研究之測試流程圖如圖5.2所示。



資料來源：本研究整理

圖5.2 測試流程圖

5.2 初步數值實驗規劃

本節將對已構建完成之大眾運輸路網進行初步之數值實驗規劃，測試之路網如圖4.1所示，路網共分成10個交通分區，每條節線之長度為500公尺，路網內共有5條大眾運輸路線，且每條大眾運輸路線上皆設有停靠站，另外還有一組分區至分區之需求量當作一開始之OD旅次資料，假設一般成本函數之全部權重設為1。而本研究假設班距分別為5、15及30分鐘，於不同班距下透過已構建之大眾運輸路網進行測試與分析。以下將分別探討在不同路網擁擠情況與大眾運輸容量限制時對於路網旅行時間與成本之影響。

5.2.1 不考量擁擠影響及不考量大眾運輸容量限制

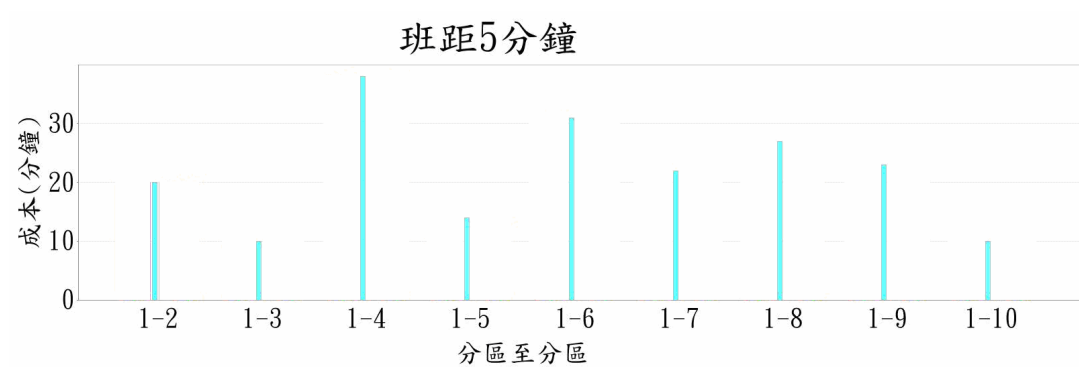
假設不考慮乘客於擁擠狀況下於停靠站所增加之等車時間，且假設大眾運輸可以服務路網上全部之乘客，故其平均旅行時間將直接模擬所設定好的班表而來，而不考量擁擠，故其一般成本函數將不會增加額外的等車時間。本研究假設起迄點為區中心至區中心，分別於班距5、15及30分鐘下，產生不同班距下之大眾運輸路線平均旅行時間等統計值，如表5.2所示；可以發現，路線一跟四之平均旅行時間於班距30分鐘時大於班距15分鐘大於班距5分鐘，路線二跟三的平均旅行時間於班距30分鐘時皆為最小，而路線三於班距5分鐘下之平均旅行時間較其他路線大，其原因為其所經過之停靠站數較多，而路線一分別於班距5分鐘及15分鐘下之平均旅行時間較其他路線小，原因為其所經過的停靠站數較少。而路線二、三跟五於班距5分鐘之平均旅行時間比班距30分鐘時的平均旅行時間高，原因為搭乘班距5分鐘之乘客出發時間剛好在尖峰的時候，而班距30分鐘之乘客出發時間在離峰，故班距5分鐘的平均旅行時間比30分鐘的平均旅行時間高。

而分區1至各分區於三種班距下之大眾運輸成本，如圖5.3至圖5.5所示；在任何班距下，某一分區(如分區1)至另一分區(如分區2)所指派之公車成本均會相等。而在不同班距下，各分區間之大眾運輸成本會不相等，顯示出在不同班距下會產生不同的成本。

表5.2 不同班距下各路線統計值(單位：分鐘)

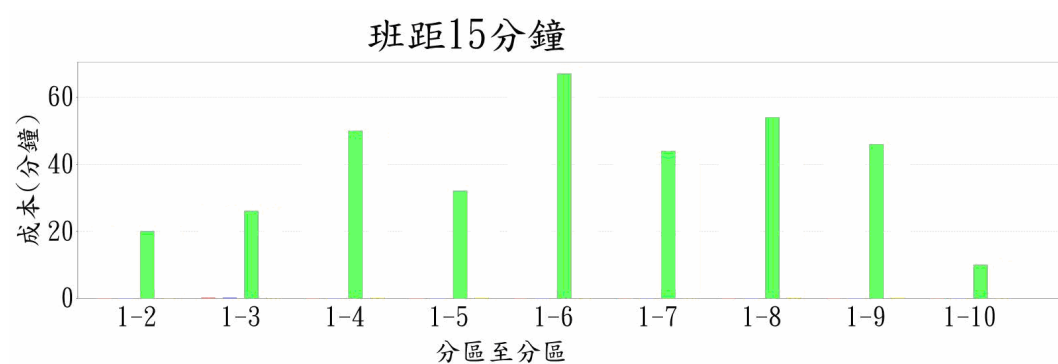
統計值 \ 路線		路線一(32→18)	路線二(7→17)	路線三(3→37)	路線四(2→44)	路線五(35→6)
班距 30 分鐘	每部車旅行時間	34.42、12.34	25.34、11.26	29.26、11.27	37.79、12.73	32.73、14.74
	平均旅行時間	23.38	18.30	20.27	25.26	23.74
	標準差	15.61	9.95	12.72	17.72	12.72
班距 15 分鐘	每部車旅行時間	17.5、27.55、19.57 、12.26	31.57、25.24、 24.25、11.26	32.3、29.22、23.24 、11.27	20.67、38.74、 24.74、12.74	19.74、30.76、 23.75、13.75
	平均旅行時間	19.22	23.08	24.01	24.22	22.03
	標準差	14.92	10.17	9.32	15.11	15.03
班距 5 分鐘	每部車旅行時間	11.82、13.86、16.55 、16.54、27.31、 24.44、23.49、 19.39、24.39、 14.41、10.54、9.48	14.27、15.46、 18.64、29.48、 27.4、32.39、 23.58、20.37、 15.27、15.27、 12.26、11.27	18.68、23.35、 29.45、30.49、 28.27、28.23、 28.22、36.22、 36.23、32.27、 29.24、25.24	14.66、18.79、 16.76、17.52、 22.41、32.64、 26.77、31.78、 21.87、16.88、 12.76、12.74	17.75、21.8、27.75 、26.84、26.74、 28.74、30.73、 28.72、24.73、 22.73、20.75、 17.75
	平均旅行時間	17.69	19.81	28.82	20.47	24.59
	標準差	17.43	16.27	20.07	17.99	17.20

資料來源：本研究整理



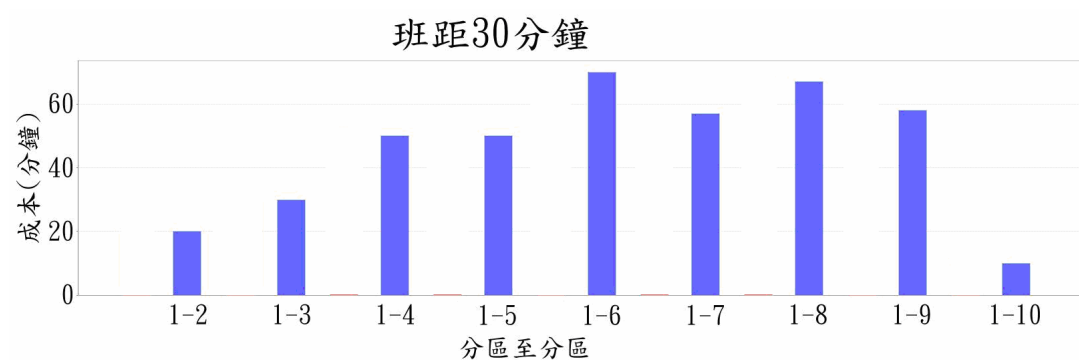
資料來源：本研究整理

圖 5.3 班距5分鐘各分區間大眾運輸成本



資料來源：本研究整理

圖 5.4 班距15分鐘各分區間大眾運輸成本



資料來源：本研究整理

圖 5.5 班距30分鐘各分區間大眾運輸成本

5.2.2 不考量擁擠影響及考量大眾運輸容量限制

故本情境將針對不同之起迄點，考量大眾運輸之容量限制，當停靠站之人數大於大眾運輸容量50人時，將無法服務全部的乘客，無法上車之乘客將會累積到下一班公車上，當停靠站之累積人數大於公車容量時，則將再指派另一部公車來服務累積在停靠站之人數，故測試完成之後將可以得到模擬一小時後將產生幾部大眾運輸車輛，班距5分鐘時將產生17部車，班距15分鐘時將產生12部車，班距30分鐘時將產生11部車，之後再將所產生之車輛數輸入DynaTAIWAN模擬產生公車路線之平均旅行時間等統計值，如表5.3及5.4 所示；由結果得知：路線一、三跟四的平均旅行時間於班距15分鐘時大於班距5分鐘大於班距30分鐘，而路線二跟五的平均旅行時間於班距5分鐘時大於班距15分鐘大於班距30分鐘，路線三分別於班距5分鐘及15分鐘下之平均旅行時間較其他路線大，其原因為其所經過之停靠站數較多，而路線一分別於班距5分鐘及15分鐘下之平均旅行時間較其他路線小，原因為其所經過的停靠站數較少。於班距5分鐘之平均旅行時間比班距30分鐘時的平均旅行時間高，原因為搭乘班距5分鐘之乘客出發時間剛好在尖峰的時候，而班距30分鐘之乘客出發時間在離峰，故班距5分鐘的平均旅行時間比30分鐘的平均旅行時間高。而於班距30分鐘時，可以發現考慮加班車之後班距愈大的班次，其平均旅行時間將大幅降低。

表5.3 不同班距下各路線統計值(單位：分鐘)

<div> <div>統計值</div> <div>路線</div> </div>		路線一(32→18)	路線二(7→17)	路線三(3→37)	路線四(2→44)	路線五(35→6)
班距 30 分鐘	每部車旅行時間	24.45、25.42、19、 14.17、13.29、16.13 、14.76、14.36、 9.36、8.96、9.17	24.24 、 22.23 、 11.07 、 12.98 、 16.77 、 11.29 、 12.19 、 11.69 、 11.26 、 10.77 、 10.96	29.28 、 20.26 、 17.77 、 14.95 、 12.66 、 14.06 、 13.28 、 14.17 、 11.26 、 10.74 、 10.95	36.78 、 17.29 、 13.49 、 24.91 、 12.57 、 12.58 、 12.11 、 12.74 、 12.75 、 12.25 、 12.45	30.75 、 24.73 、 21.23 、 18.45 、 17.55 、 16.67 、 16.75 、 17.16 、 13.75 、 13.26 、 13.47
	平均旅行時間	15.37	14.13	15.40	16.36	18.52
	標準差	10.62	12.00	11.06	10.77	10.88
班距 15 分鐘	每部車旅行時間	17.48、22.16、27.37 、27.68、29.13、 26.04 、 24.42 、 25.44 、 25.4 、 18.63、16.11、13.54	30.94 、 24.77 、 27.18 、 25.23 、 23.86 、 22.95 、 21.25 、 20.37 、 22.17 、 13.75 、 11.07 、 11.26	29.59 、 31.03 、 36.97 、 35.26 、 31.78 、 28.98 、 27.27 、 30.03 、 29.15 、 22.77 、 20 、 16.28	21.79 、 22.83 、 32.49 、 35.77 、 33.27 、 28.62 、 30.53 、 19.41 、 26.91 、 27.92 、 12.8 、 16.49	19.69、30.3、26.52 、26.24、33.75、 21.54 、 23.47 、 22.65 、 23.73 、 19.23 、 16.45 、 13.74
	平均旅行時間	22.78	21.23	28.26	25.74	23.11
	標準差	12.29	7.88	9.38	11.60	10.60

資料來源：本研究整理

表5.4 不同班距下各路線統計值(續)(單位：分鐘)

	路線	路線一(32→18)	路線二(7→17)	路線三(3→37)	路線四(2→44)	路線五(35→6)
統計值						
班距 5 分鐘	每部車旅行時間	11.82、13.86、16.58 、14.97、16.47、 22.26、24.35、 26.57、29.06、 24.16、23.58、24.96 、14.41、24.4、21.25 、11.39、9.46	14.27、15.47、 20.69、19.09、 31.39、33.15、 28.82、31.42、 28.51、28.06、 23.86、26.38、 23.31、20.29、 15.48、11.26、 11.25	18.69、23.65、 30.39、30.78、 31.28、29.78、 28.36、26.84、 27.23、26.74、 30.23、28.76、 27.25、26.26、 27.29、18.26、 23.27	14.66、18.62、 17.35、17.11、 16.74、18.78、 19.24、29.4、 30.74、33.26、 32.76、32.24、 25.74、30.78、 21.75、16.76、 12.75	17.76、21.79、 26.74、27.27、 25.76、25.19、 24.69、27.16、 31.74、32.24、 27.25、30.67、 25.75、22.74、 16.75、21.74、 13.75
	平均旅行時間	19.39	22.51	26.77	22.86	24.65
	標準差	15.95	14.64	14.78	17.82	14.47

資料來源：本研究整理

5.2.3 考量擁擠影響及考量大眾運輸容量限制

由於考量擁擠影響，故考量乘客於擁擠狀況下於停靠站所增加之等車時間，且本情境亦將考量大眾運輸之容量限制，當停靠站之人數大於大眾運輸容量時，將無法服務全部的乘客，所以於停靠站將會出現無法上車之乘客，故這些乘客將會累積到下一班大眾運輸上，當停靠站之累積人數大於公車容量時，則將再指派另一部公車來服務累積在停靠站之人數。因此，當乘客搭乘加班車時，其成本需額外增加一項等車時間。

分區1至各分區之大眾運輸成本，如圖5.6 至圖5.8 所示，由測試結果得知：由於考量加班車之緣故，在任何班距下，某一分區(如分區1)至另一分區(如分區2)所指派之公車輸成本不完全相同，其中加班車之成本將會提高，原因為乘客需多增加等車時間，而在不同班距下，各分區間之大眾運輸成本會不相等，顯示出在不同班距下會有不同的成本考量，且各分區間之加班車成本將較大，以圖5.8 為例，當班距為30分鐘時，其第一部加班車還是無法滿足乘客需求，故需另外派一部加班車，且此部加班車之成本亦將增加。

由於大眾運輸皆有容量限制，故須考量到因為擁擠情況導致乘客無法搭乘大眾運輸的情況，因此需加派加班車來服務此需求；當大眾運輸班距為5分鐘時，路線2公車於停靠站點13至點27之需求發現明顯增加，故需加派加班車來滿足需求。而班距不一樣，需求增加將發生在不同的停靠站，以班距15分鐘為例，於停靠站點25發現需求增加，但於班距5分鐘時，需求量增加係發生於停靠站點13。

而由輸出結果發現：班距不一樣時，乘客所選擇搭乘大眾運輸的方式亦將不同；當班距15分鐘時，乘客由分區一到分區六將先搭乘路線3之大眾運輸由停靠站點2至點7，於點7轉搭路線4之大眾運輸到點28，再轉路線2之大眾運輸到點30。而於班距30分鐘時，乘客將不選擇搭乘大眾運輸工具，而直接採用步行的方式，由點2->點44->點5->點6->點12->點18->點24->點30。

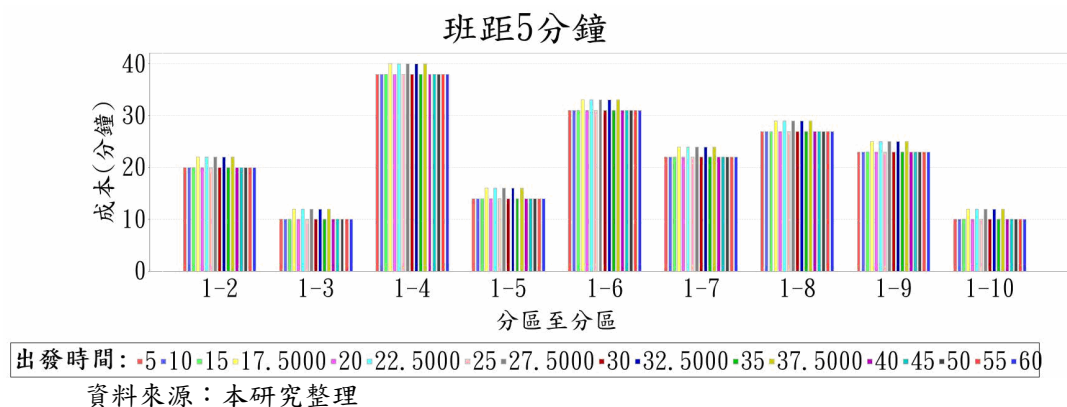
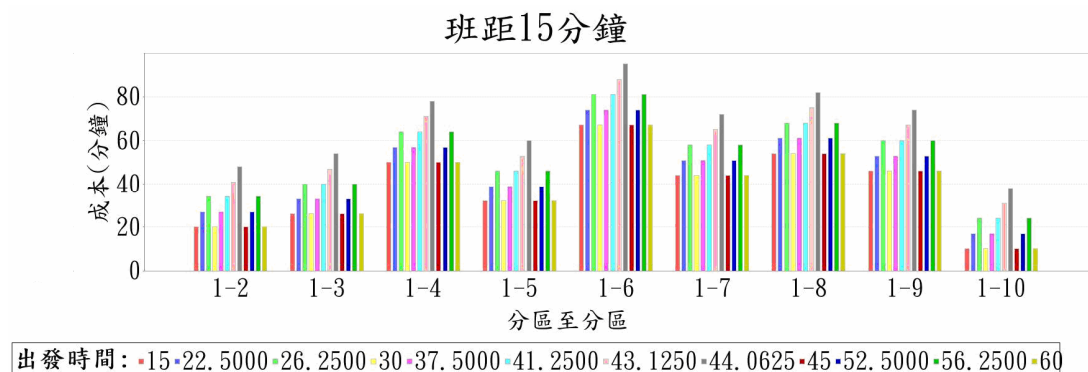
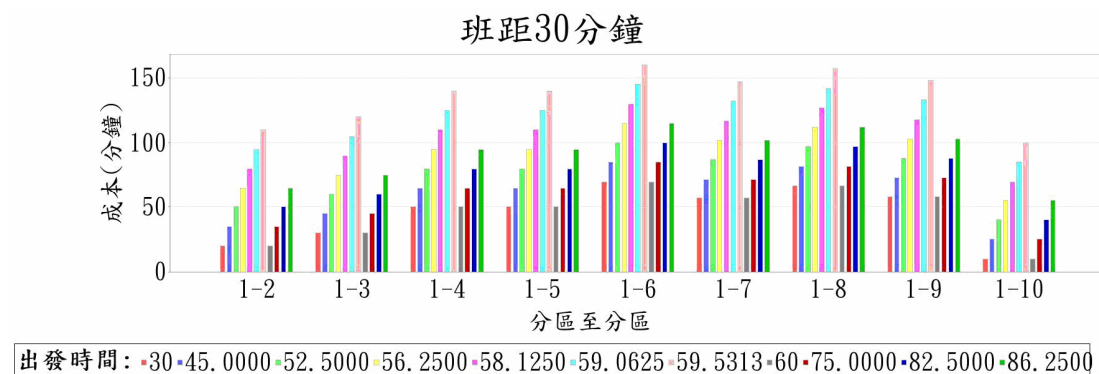


圖5.6 班距5分鐘各分區間大眾運輸成本



資料來源：本研究整理

圖5.7 班距15分鐘各分區間大眾運輸成本



資料來源：本研究整理

圖5.8 班距30分鐘各分區間大眾運輸成本

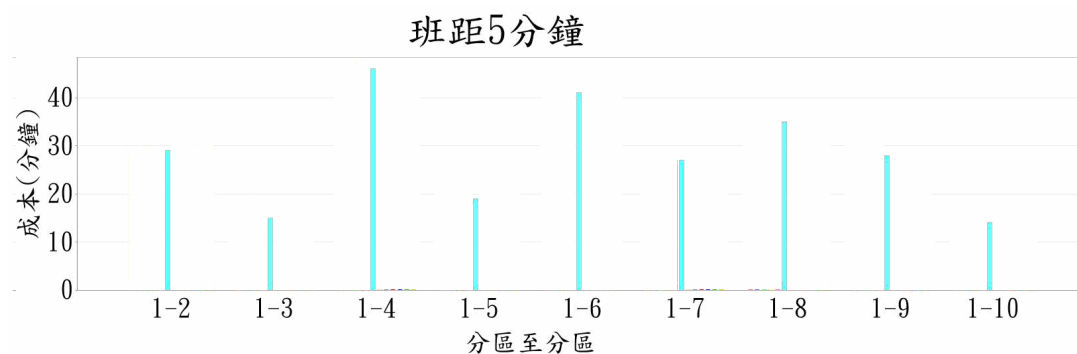
5.3 敏感度測試

本節假設一般成本函數考量了權重的影響，其中权重之設定係參酌交通部運研所、亞聯顧問公司及台北市政府捷運局等研究報告整理而成，假設車上旅行時間权重為1、步行時間权重為2、等車時間权重為2、轉車時間权重為3，針對不同之班距設計不同條件下之數值實驗，而本研究假設班距分別為5、15及30分鐘，於不同班距下透過已構建之大眾運輸路網進行測試與分析。

以下將分別探討在考量权重因子下，於不同路網擁擠情況與大眾運輸容量限制時對於路網旅行時間與成本之影響以及當需求改變時大眾運輸路線所產生之變化。

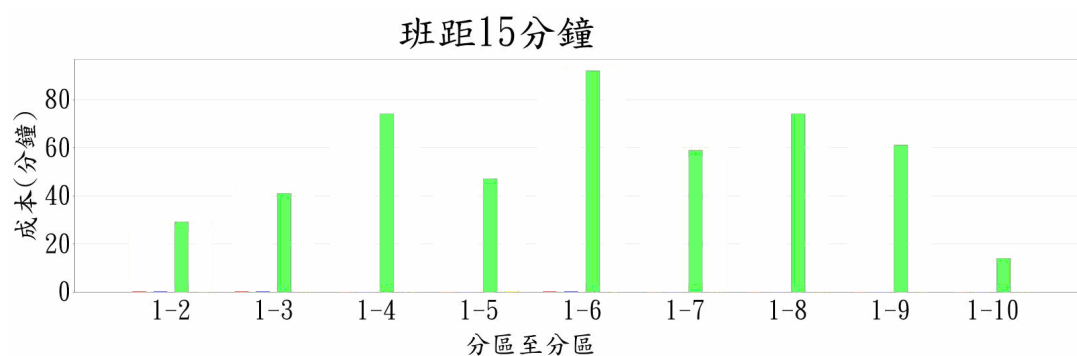
5.3.1 不考量擁擠影響及不考量大眾運輸容量限制

假設不考慮乘客於擁擠狀況下於停靠站所增加之等車時間，且假設大眾運輸可以服務路網上全部之乘客，亦即不考量擁擠影響也不考量大眾運輸容量限制。因此，將不考量加班車的加入，故其平均旅行時間將直接模擬所設定好的班表而來，而不考量擁擠，故其一般成本函數將不會增加額外的等車時間。本研究假設起迄點為區中心至區中心，分別於班距5、15及30分鐘下，產生不同班距下之大眾運輸路線平均旅行時間及分區1至各分區之大眾運輸成本，如圖5.9 至圖5.11 所示；在任何班距下，某一分區(如分區1)至另一分區(如分區2)所指派之公車成本均會相等。而在不同班距下，各分區間之大眾運輸成本會不相等，顯示出在不同班距下會產生不同的成本。



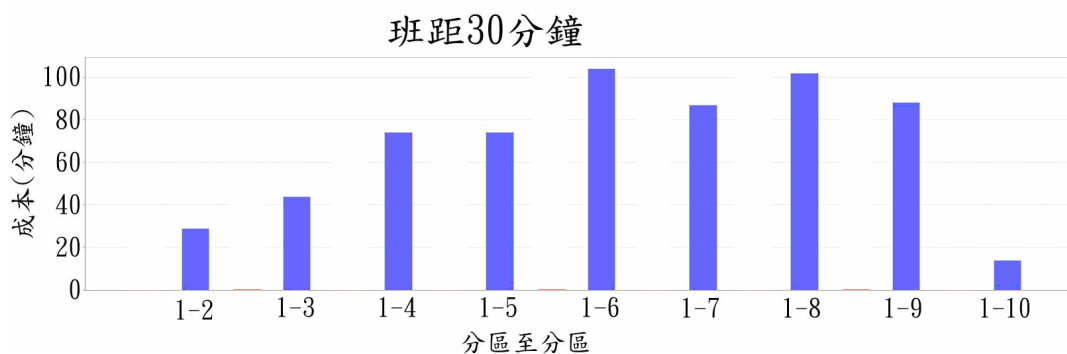
資料來源：本研究整理

圖5.9 班距5分鐘各分區間大眾運輸成本



資料來源：本研究整理

圖5.10 班距15分鐘各分區間大眾運輸成本

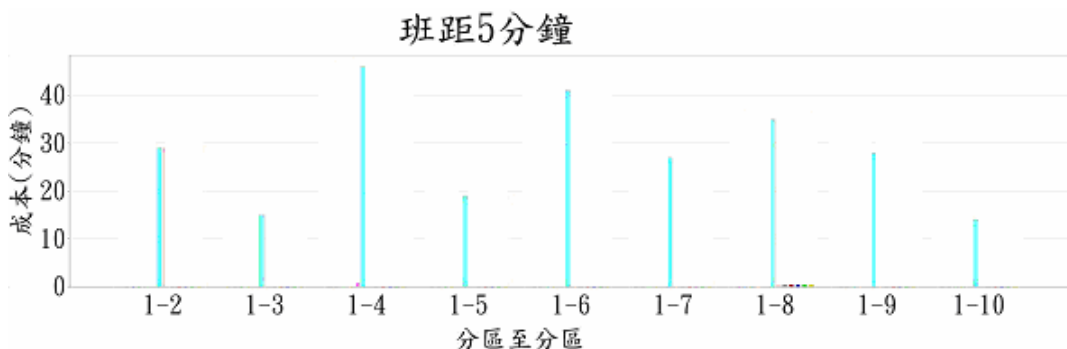


資料來源：本研究整理

圖5.11 班距30分鐘各分區間大眾運輸成本

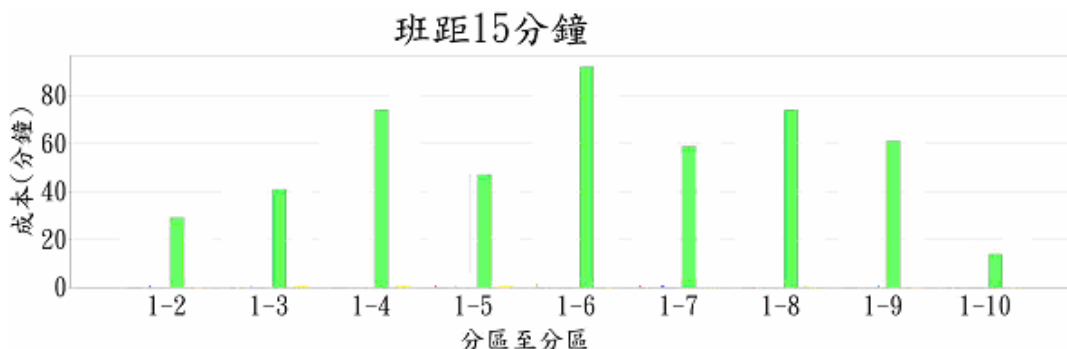
5.3.2 不考量擁擠影響及考量大眾運輸容量限制

故本情境將針對不同之起迄點，進行實驗設計並考量大眾運輸之容量限制，當停靠站之人數大於大眾運輸容量50人時，將無法服務全部的乘客，所以於停靠站將會出現無法上車之乘客，故這些乘客將會累積到下一班大眾運輸上，當停靠站之累積人數大於公車容量時，則將再指派另一部公車來服務累積在停靠站之人數，故測試完成之後將可以得到模擬一小時後將產生幾部大眾運輸車輛，班距5分鐘時將產生17部車，班距15分鐘時將產生12部車，班距30分鐘時將產生11部車，之後再將所產生之車輛數輸入DynaTAIWAN模擬產生大眾運輸路線之平均旅行時間及分區1至各分區之大眾運輸成本，如圖5.12 至圖5.14 所示；在任何班距下，某一分區(如分區1)至另一分區(如分區2)所指派之公車成本均會相等。而在不同班距下，各分區間之大眾運輸成本會不相等，顯示出在不同班距下會產生不同的成本。



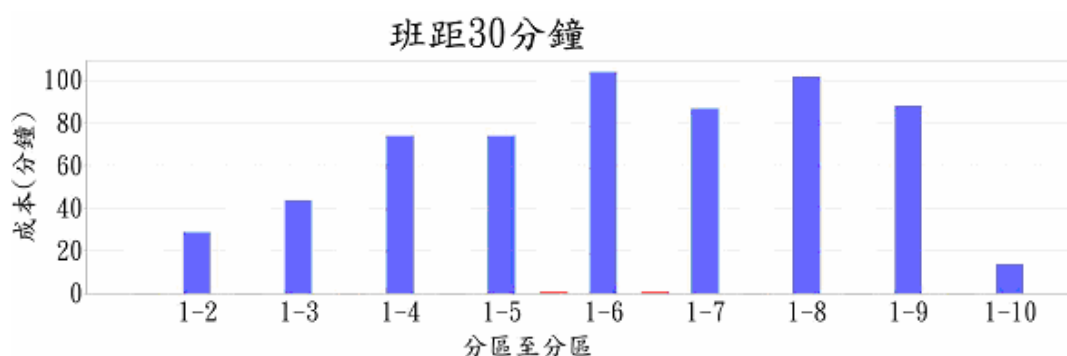
資料來源：本研究整理

圖5.12 班距5分鐘各分區間大眾運輸成本



資料來源：本研究整理

圖5.13 班距15分鐘各分區間大眾運輸成本



資料來源：本研究整理

圖5.14 班距30分鐘各分區間大眾運輸成本

5.3.3 考量擁擠影響及考量大眾運輸容量限制

本情境將考量大眾運輸之容量限制，當停靠站之人數大於大眾運輸容量時，將無法服務全部的乘客，所以於停靠站將會出現無法上車之乘客，故這些乘客將會累積到下一班大眾運輸上，當停靠站之累積人數大於公車容量時，則將再指派另一部公車來服務累積在停靠站之人數。因此，當乘客搭乘加班車時，其成本需額外增加一項等車時間。

分區1至各分區之大眾運輸成本，如圖5.15 至圖5.17 所示，由測試結果得知：由於考量加班車之緣故，在任何班距下，某一分區(如分區1)至另一分區(如分區2)所指派之公車成本不完全相同，其中加班車之成本將會提高，原因為乘客需多增加等車時間，而在不同班距下，各分區間之大眾運輸成本會不相等，顯示出在不同班距下會有不同的成本考量，且各分區間之加班車成本將較大，以圖5.17 為例，當班距為30分鐘時，其第一部加班車還是無法滿足乘客需求，故需另外派一部加班車，且此部加班車之成本亦將增加。由於此情境考量了擁擠影響及大眾運輸容量限制，故本情境之服務水準為所有情境中最高。

由於大眾運輸皆有容量限制，故須考量到因為擁擠情況導致乘客無法搭乘大眾運輸的情況，因此需加派加班車來服務此需求；當大眾運輸班距為5分鐘時，路線2公車於停靠站點13至點27之需求發現明顯增加，故需加派加班車來滿足需求。而班距不一樣，需求增加將發生在不同的停靠站，以班距15分鐘為例，於停靠站點25發現需求增加，但於班距5分鐘時，需求增加係發生於停靠站點13。

而由輸出結果發現：班距不一樣時，乘客所選擇搭乘大眾運輸的方式亦將不同；當班距5分鐘時，乘客由分區一到分區六將先搭乘路線3之大眾運輸由停靠站點2至點7，於點7轉搭路線4之大眾運輸到點28，再轉路線2之大眾運輸到點30。而於班距30分鐘時，乘客將不選擇搭乘大眾運輸工具，而直接採用步行的方式，由點2->點44->點5->點6->點12->點18->點24->點30。

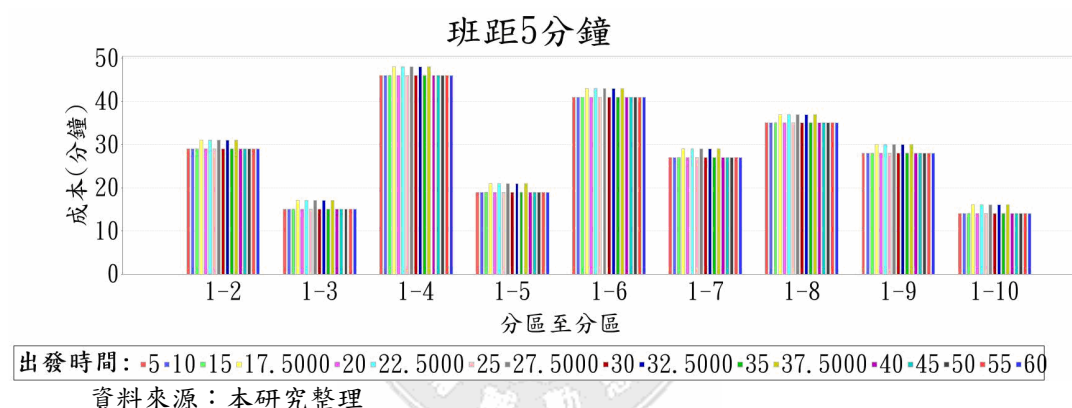


圖5.15 班距5分鐘各分區間大眾運輸成本

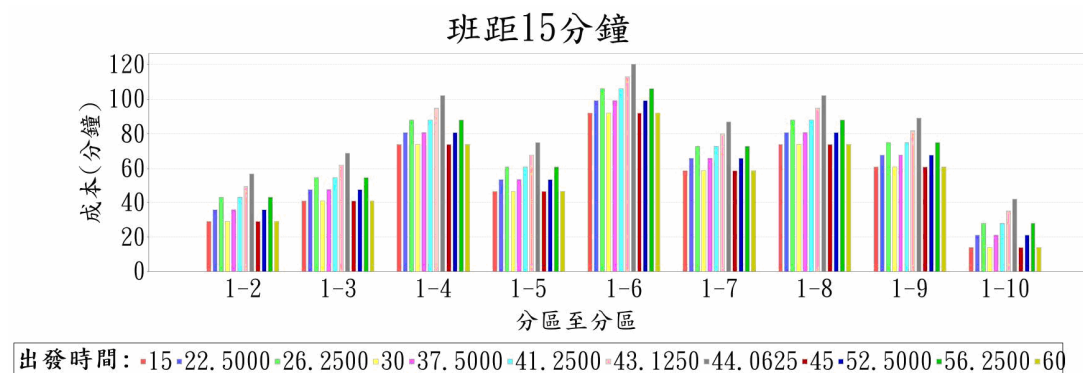
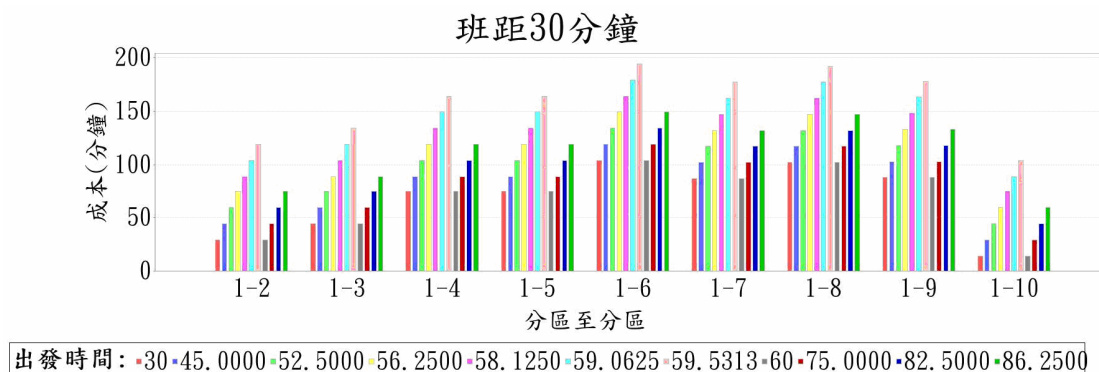


圖5.16 班距15分鐘各分區間大眾運輸成本

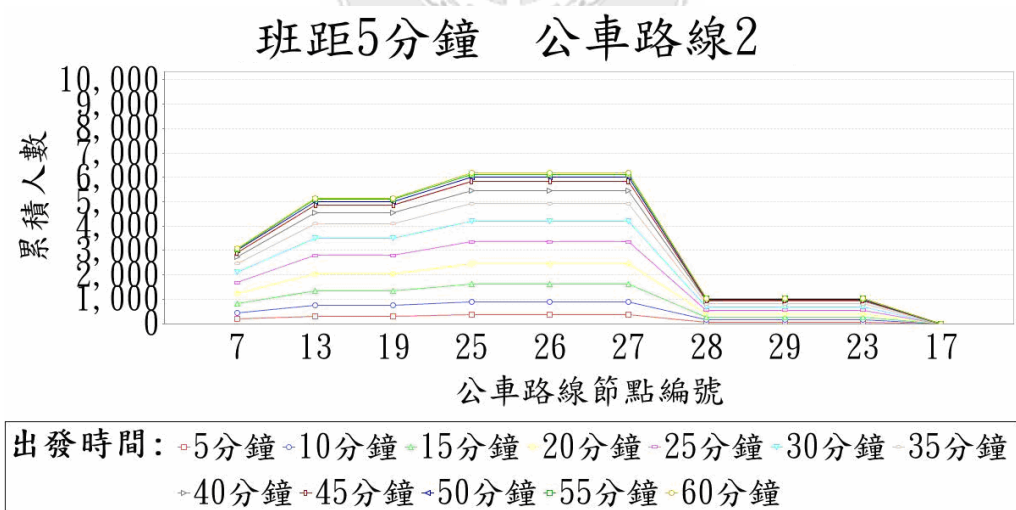


資料來源：本研究整理

圖5.17 班距30分鐘各分區間大眾運輸成本

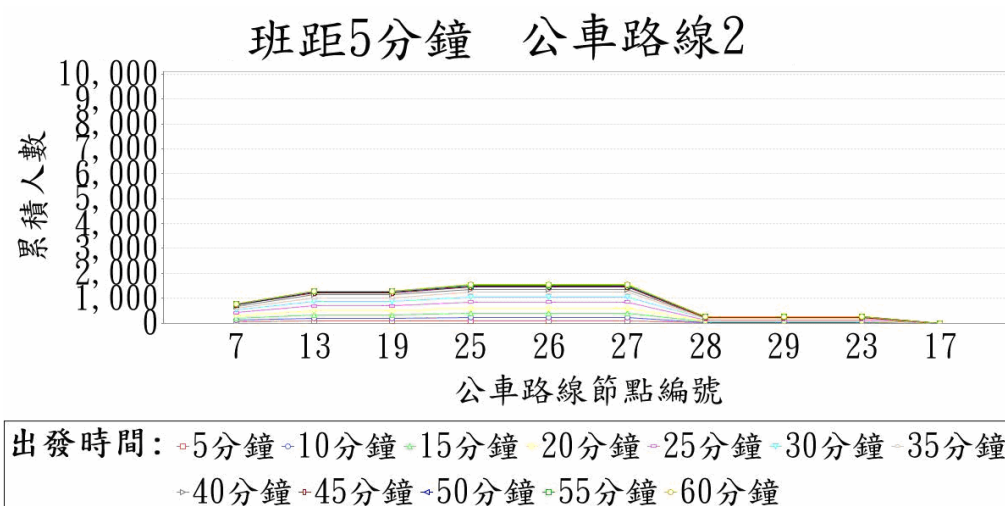
5.3.4 不同 OD 需求測試

本節將測試於不同需求時，大眾運輸路線反應的情況，當需求加倍時，如圖5.18 所示；路線2之大眾運輸所需服務的乘客數亦增加，因此需加派更多的加班車來服務乘客。而當需求降低時，如圖5.19 所示，路線2之大眾運輸所需服務的乘客數亦減少，因此業者可考量是否增加班距，讓班距加大來降低成本。



資料來源：本研究整理

圖5.18 公車路線2於班距5分鐘下累積人數



資料來源：本研究整理

圖5.19 公車路線2於班距5分鐘下累積人數

5.4 均衡指派測試

一般在大眾運輸路網中，乘客於任何兩停靠站之間可以有很多條大眾運輸路線供選擇，假設乘客會選擇使得其旅行成本最小之路線。而大眾運輸皆有容量限制，因此，在考量旅行成本時尚需考量到大眾運輸之容量限制，因為當停靠站之乘客人數增加時等車時間也會增加，進而影響到旅行成本。

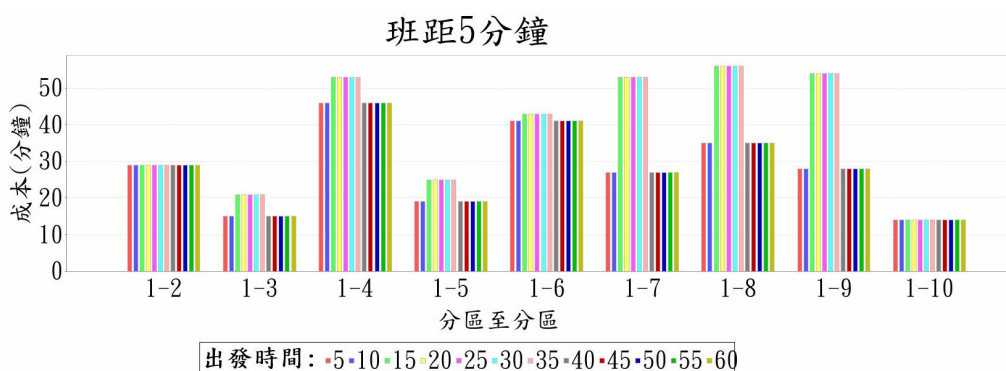
本研究將假設大眾運輸路網的擁擠現象大多發生在大眾運輸的停靠站。而乘客的等車時間是依據大眾運輸容量限制跟乘客數來決定。所以，當乘客數接近大眾運輸容量時，等車時間會增加。而且乘客將考量最小成本的路線，且將搭乘第一部到達停靠站的大眾運輸車輛，如果大眾運輸旅行時間變大時，乘客將被指派至另一條大眾運輸路線，以達到路網均衡情況。

測試結果如下圖5.20 至如5.22 所示；以班距5分鐘為例，分區一至分區三之大眾運輸成本於某些出發時間會有變大的現象，表示此時間下之大眾運輸路線已經呈現滿載的現象，所以成本會提高，因此乘客將選搭另一條大眾運輸路線，而當班距為30分鐘時，並無成本提高之現象，原因為當班距變大時，乘客大多採取步行的方式，因此該大眾運輸可以服務全部的乘客。

而在大眾運輸所服務的總人數方面，如圖5.23 至圖5.26 所示；圖5.23 為未考量均衡時，公車路線2於班距15分鐘下累積人數，而圖5.24 為考量均衡時公車路線2於班距15分鐘下累積人數，可以發現同一條路線所服務之人數將會明顯減少，而所減少的人數將會移轉至其他路線，如圖5.25 及

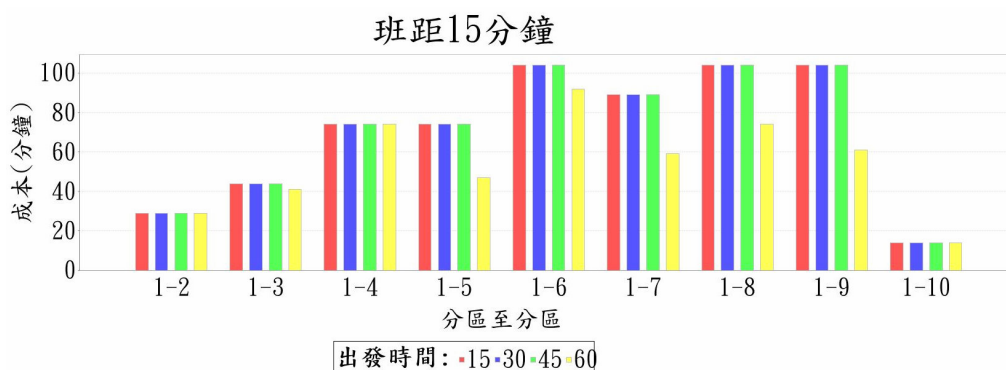
圖5.26 所示；圖5.25 為未考量均衡時，公車路線3於班距15分鐘下累積人數，而圖5.26 為考量均衡時公車路線3於班距15分鐘下累積人數，可以發現路線3一開始所服務之人數將會增加，此增加之人數為原本搭乘路線2之乘客因為無法搭乘原本的路線而移轉至路線3。

而乘客所選搭大眾運輸的路徑方面，如圖5.27 及圖5.28 所示；原本從分區一至分區六的乘客一開始所走的路徑為搭乘路線5之大眾運輸由停靠站點2至點35，再轉路線4之大眾運輸由停靠站點35至點30，由於此路線之大眾運輸無法滿足全部乘客，因此無法搭乘此路徑之乘客將選擇搭乘另外一條路線，其新的路徑為搭乘路線3之大眾運輸由停靠站點2至點7，再轉搭路線2之大眾運輸至停靠站點28，最後再採用步行的方式，由點28->29->30。



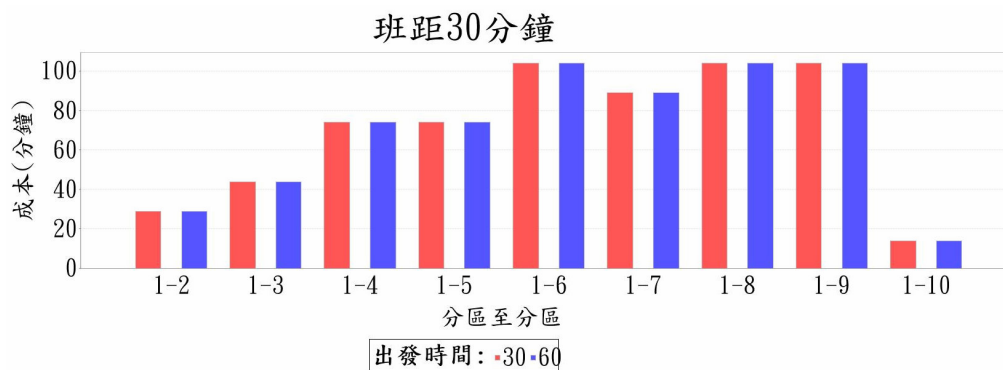
資料來源：本研究整理

圖 5.20 班距 5 分鐘各分區間大眾運輸成本



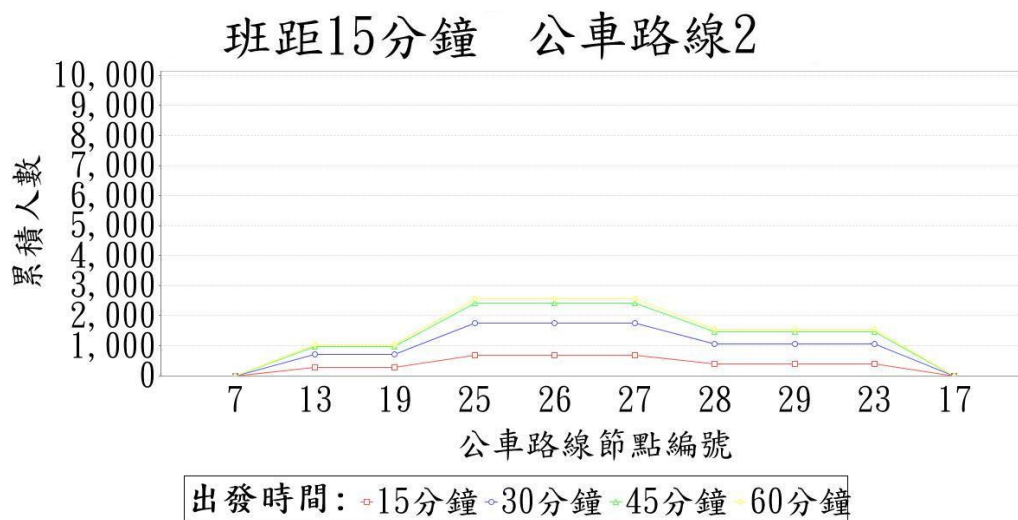
資料來源：本研究整理

圖 5.21 班距 15 分鐘各分區間大眾運輸成本



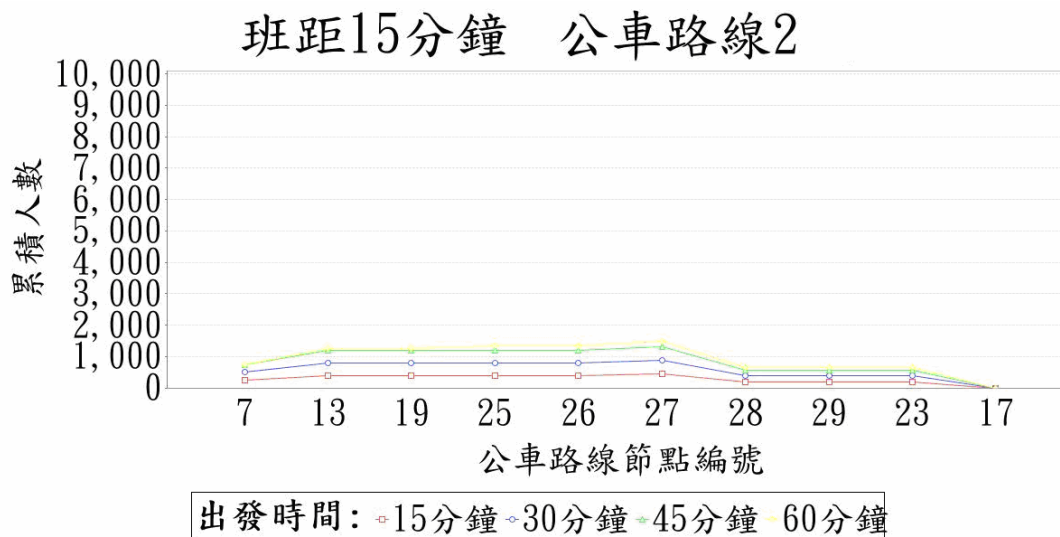
資料來源：本研究整理

圖5.22 班距30分鐘各分區間大眾運輸成本



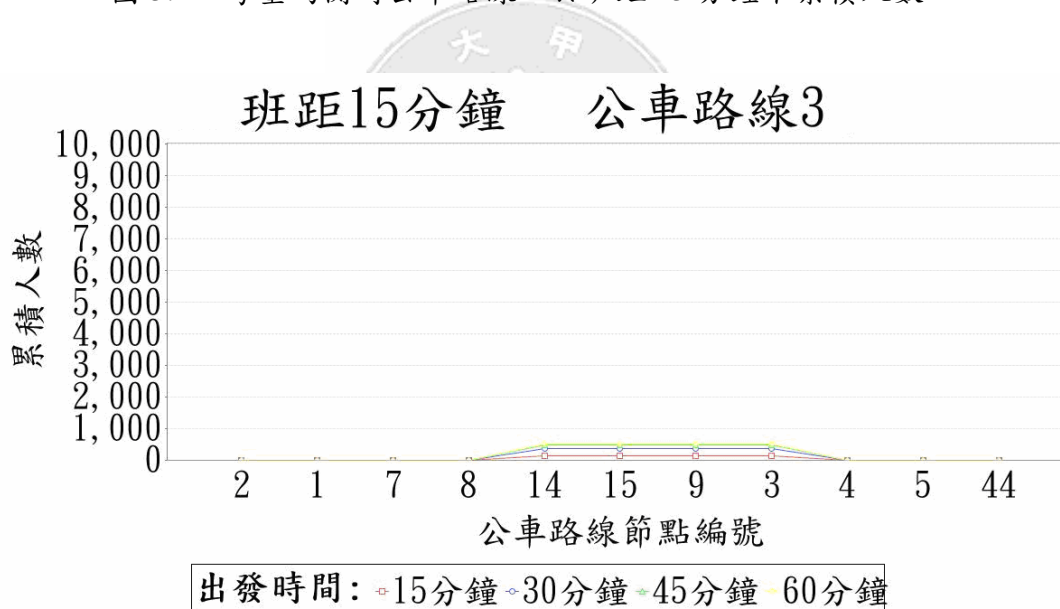
資料來源：本研究整理

圖 5.23 公車路線 2 於班距 15 分鐘下累積人數



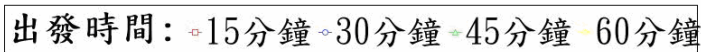
資料來源：本研究整理

圖 5.24 考量均衡時公車路線 2 於班距 15 分鐘下累積人數



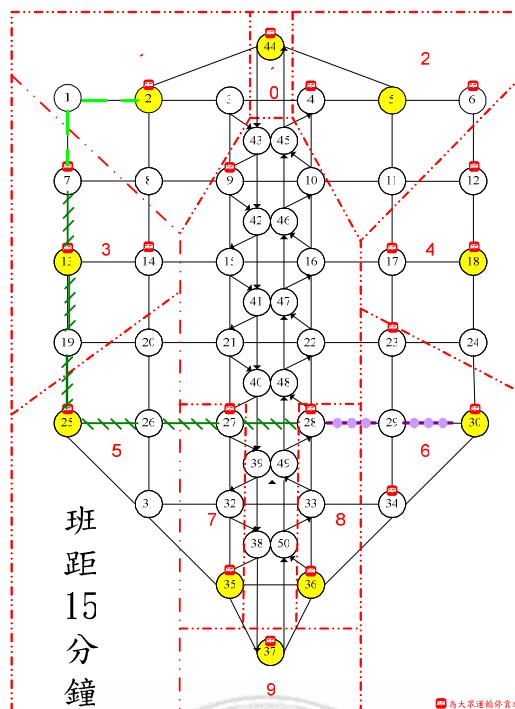
資料來源：本研究整理

圖 5.25 公車路線 3 於班距 15 分鐘下累積人數



累積人數

乘方式



資料來源：本研究整理

圖 5.28 考量均衡時，班距 15 分鐘分區一至分區六乘客搭乘方式

5.4.1 小結

實驗結果中本研究使用人數移轉百分比表示考量均衡時所帶來的路線效益，人數移轉百分比定義如下：

$$\text{人數移轉百分比} = \frac{NELTP - ELTP}{NELTP} \times 100\%$$

$NELTP$ ：未考量均衡條件下，大眾運輸路線總乘客數

$ELTP$ ：考量均衡條件下，大眾運輸路線總乘客數

結果如表 5.5 所示；出現負值表示大眾運輸路線於考量均衡時總乘客數增加，而正值表示乘客數有移轉到其他大眾運輸路線的現象，而當班距為 30 分鐘時，人數移轉百分比出現 0% 的情況，是由於路線 1、3 及 4 於班距 30 分鐘時並無乘客搭乘，乘客皆採取步行的方式。由結果發現：班距 15 分鐘時，所獲得之效益較高。

表5.5 不同班距下各路線人數移轉百分比

路線 班距	5min	15min	30min
路線 1	1.50%	7.77%	0.00%
路線 2	35.76%	42.91%	80.58%
路線 3	8.25%	45.63%	0.00%
路線 4	11.43%	52.34%	0.00%
路線 5	11.00%	63.11%	80.58%

資料來源：本研究整理

而當需求加倍時，由表5.6 所示；發現當乘客數增加，考量均衡情況所得到之效益越高。以班距5分鐘為例，當需求加倍時，人數移轉百分比從原本最高之11.43%提高至48.18%，所以當需求增加時，所產生之效益更明顯。

表5.6 不同班距下各路線人數移轉百分比

路線 班距	5min	15min	30min
路線 1	13.33%	41.75%	0.00%
路線 2	57.12%	68.35%	90.29%
路線 3	15.37%	22.33%	0.00%
路線 4	48.18%	73.61%	0.00%
路線 5	40.13%	79.55%	90.29%

第六章 結論與建議

本研究主要是探討動態運輸規劃模型中大眾運輸指派，應用大眾運輸指派程式產生最短路徑及最短路徑上之旅行時間及成本，研究中透過模擬指派模式DynaTAIWAN考慮了不同班距下之大眾運輸旅行時間。以下針對研究研究所得之結果，提出相關結論與後續可進行研究之建議。

6.1 結論

1. 動態運輸規劃之模式規劃應考量依時性旅次、即時性資訊與動態分析之能力，並整合大眾運輸系統的指派以進行交通系統之規劃與分析，傳統的大眾運輸指派，以全有全無指派為主，本研究建立大眾指派流程並考量擁擠影響與大眾運輸容量限制，改善了傳統大眾運輸指派的缺點。
2. 本研究測試結果當不考量擁擠影響也不考量大眾運輸容量限制，所測試的結果並不能符合均衡指派情況，故整體來說，考量大眾運輸之容量限制與擁擠情況較為符合均衡指派情況。
3. 當考量乘客於擁擠狀況下於停靠站所增加之等車時間，且考量大眾運輸之容量限制，所以於停靠站將會出現無法上車之乘客，故這些乘客將會累積到下一班公車上，當停靠站之累積人數大於公車容量時，則將再指派另一部公車來服務累積在停靠站之人數，此方法之服務水準為所有情境中最高。
4. 由測試結果得知，當班距不一樣時，乘客所選擇搭乘大眾運輸的方式亦將不同，而當班距變大時，乘客將可能採用步行的方式來取代搭乘大眾運輸工具。
5. 當考量大眾運輸容量限制時，路網上將會產生因大眾運輸容量已滿載而無法搭乘此班大眾運輸之乘客，故將這些無法搭乘此班大眾運輸之乘客，指派至另一條大眾運輸路線，以達到路網均衡情況，當發現同分區間之大眾運輸成本提高時，表示此時間下之大眾運輸路線已經呈現滿載的現象，因此乘客將選搭另一條大眾運輸路線。

6.2 建議

1. 本研究假設旅次皆集中於區中心，而以距離區中心最近之停靠站為乘客可選搭之停靠站，但並非所有分區旅客均會考量最近之停靠站，實際上其他非最短路徑亦可能被乘客所選用。故未來可考量非最短距離之停靠站來進行旅次指派之工作。

2. 本研究為了簡化大眾運輸路網構建之工作，故採用虛擬的小路網，故建議未來可構建真實之路網，如台中市路網，進行旅次指派。
3. 本研究所假設之一般成本函數為線性關係，而真正之函數可能為非線性函數，故之後可以嘗試不同函數來分析。
4. 大眾運輸指派完成之後可提供資訊給路網使用，而此資訊可做為路網績效評估之依據，故如何將指派結果應用於路網設計中，亦為日後研究之重點。



參考文獻

1. 王慶瑞(2001)，「運輸系統規劃」，亞聯工程顧問公司。
2. 宋明山(1973)，「公車系統網路交通量分派之研究」，國立台灣大學土木所碩士論文。
3. 李克聰(1995)，「運輸規劃」，鼎漢國際工程顧問公司。
4. 亞聯工程顧問股份有限公司(2001)，「台北都會區整體運輸規劃－基本資料之調查與驗校（二）」，台北市交通局。
5. 施鴻志、段良雄、凌瑞賢(1988)，「都市交通計畫理論與實務」，茂昌圖書有限公司。
6. 胡大瀛(2005)，「區域級智慧型運輸系統示範計畫－核心交通分析與預測系統(第2年期)」，交通部運輸研究所。
7. 凌瑞賢(2001)，「運輸規劃原理與實務」，鼎漢國際工程顧問公司。
8. 陳文能(2003)，「智慧型運輸系統下一動態運輸規劃模型之建立」，逢甲大學交管所碩士論文。
9. 黃俐嘉(1997)，「公車路網績效評估模式之研究」，國立台灣大學土木工程所碩士論文。
10. 鄭樂堯(1985)，「大眾運輸旅次指派方法研究-以台南市公車系統為例」，國立成功大學交管所碩士論文。
11. Clercq, F. K.(1972), “ A public transport assignment method” , Traffic Engineering & Control, Vol. 14, No. 2, 91-96.
12. De Cea, J. and J.E. Fernandez (1993),” Transit Assignment for Congested Public Transport Systems: An Equilibrium Model”, Transportation Science, Vol. 27, NO. 2,133-147.
13. De Cea, J. and J.E. Fernandez(2000),” Transit-Assignment Models” , Chapter 30, Handbook of Transport Modeling, edited by Hensher and Button, pp. 497-508.
14. Desaulniers, G. and Hickman, M.D. (2003), “ Public Transit” , Les Cahiers du GERAD, G-2003-77, ISSN: 0711-2440.
15. Han, F. A. and Wilson, H. M.(1982), “ The allocation of buses in heavily utilized networks with overlapping routes”, Transportation Research, Vol. 16B, NO. 3, 221-232.
16. Lampkin, W. and Saalmans, P. D.(1967),” The design of routes, service frequencies, and schedules for a municipal bus undertaking: a case study”, Operational Research Quarterly, Vol. 18, 375-397.

17. Last, A. and Leak, S. E.(1976),” Transept: a bus model” , Traffic Engineering & Control, Vol. 17, No. 1, 14-17.
18. Silman, L. A., Barley, Z. and Passy, U.(1974),” Planning the route system for urban buses” , Computer & Operational Research, Vol. 1, 201-211.
19. Wu, J.H, Florian, M. and Marcotte, P. (1994),” Transit Equilibrium Assignment: A Model and Solution Algorithms“, Transportation Science, Vol. 28, NO. 3,193-203.





輸入資料說明

1. 大眾運輸路網資料 (Transit Network data)

檔名：transitnetwork.dat

說明：大眾運輸路網道路相關屬性資料可於此進行設定，為模擬基礎資料，藉以產生路網之基本組成，主要內容為路口節點與路段節線之屬性資料。

格式：

(6) 初值資料

Link_num

Link_num：路網總節線數

(7) 節點與交通分區之對應資料 (每一節點分屬哪一分區)

Node_ID Zone_ID

Node_ID：節點編號

Zone_ID：交通分區編號

(8) 節線資料：

From_Node To_Node LENGTH

From_Node：上游節點

To_Node：下游節點

LENGTH：節線長度 (公尺)

Ex：

```
172
1 1
2 1
3 1
4 2
5 2
6 2
7 3
8 1
9 0
10 0
11 2
12 4
13 3
14 3
15 0
```

.....(以下省略)

```
1 2 500
1 7 500
2 1 500
```

```

2    3    500
2    8    500
2    44   500
3    2    500
3    4    500
3    9    500
.....(以下省略)

```

2. 公車資料 (Bus data)

檔名：bus.dat

格式：

(1) 公車數目

Bus_num

(2) 各部公車細部資料

a. Bus_ID Bus_ONod Bus_DNod BusStart_time BusStop_time
BusStop_num

b. Sequence_of_BusLine

c. StopWay

Bus_ID：公車 ID#

Bus_ONod：產生路段之起點節點編號

Bus_DNod：產生路段之起迄點節點編號

Bus_start_time：公車發車時間（單位：分）

Bus_stop_time：該公車每站停靠時間（單位：秒）

stop_num：該條公車路線所經節點數目

Sequence_of_BusLine：所經節點編號

StopWay：停靠方式

0：不停

1：stop at the near block 停靠轉入路口近端

2：stop at the midblock 停靠路段中段

3：stop at the far block 停靠轉入路口遠端

4：stop at the midblock bus bay 停靠於公車彎

Ex:

```

5
1    1    2    1.0 30.0    9
7    8    14    15    9    3    4    5    44
1    1    1    1    1    1
2    2    37    1.0 30.0    7
8    7    13    19    25    31    35
1    1    1    1    1    1    1    1
.....(以下省略)

```

3. 交通分區資料 (Zone data)

檔名：zone.dat

說明：路網交通分區之資料，包含組成數目、區中心節點編號。主要可依據路網交通分區資料來訂定。

格式：

(1) 分區總體資料

zone_num des_num node_num

zone_num：分區數目

des_num：迄點數目

node_num：總節點數

(2) 分區編號與其對應的區中心 (centroid) 節點編號

zone_ID centroid_ID

zone_ID：分區編號

centroid_ID：對應 zone_ID 分區的區中心(centroid) 節點編號

Ex：

10	10	50
1	2	
2	5	
3	13	
4	18	
5	25	
6	30	
7	35	
8	36	
9	37	
10	44	



4. Dummy link

說明：紀錄所有可供乘客上車之停靠站編號。

Ex：

2	3	4	6	7	9	12	13	14	17	18	23
25	27	28	30	32	34	35	36	37	44		

5. 轉車之Dummy link

說明：紀錄所有可供乘客轉車之停靠站編號。

Ex：

7	13	17	23	25	28	30	35	36	37
---	----	----	----	----	----	----	----	----	----

6. 流量需求資料 (Detransit Data)

檔名：detransit.dat

說明：此資料係作為模擬車輛之產生基礎，每一時段，依照 OD 矩陣產生

該時段之車輛數目，並於此時段中依序將車輛放入路網。大眾運輸之 OD 資料分由兩個檔案來輸入。假設 t 個時段有 N 個交通分區，則有 t 個 $N \times N$ 的 O-D 需求矩陣。

格式：

- (1) 時段數

n_ODinterval

- (2) 每一組 OD 矩陣分布至路網之時段

OD_Interval1 OD_Interval2 OD_Interval3 OD_Interval4

- (3) O-D 矩陣 (單位：輛) 10 筆數字一列

假設有 12 個分區，則資料為：

OD[1,1] OD[1,2]OD[1,10]

OD[1,11] OD[1,12]

OD[2,1] OD[2,2]OD[2,10]

OD[2,11] OD[2,12]

.....

OD[12,1] OD[12,2].....OD[12,10]

OD[12,11] OD[12,12]

Ex:

[illegible]

.....(以下省略)



程式碼

```
#include "GetCost/globals.h"
#include "GetCost/util.h"
#include "GetCost/buildNetwork.h"
#include "GetCost/buildBus.h"
#include "GetCost/buildZone.h"
#include "GetCost/buildDetransit.h"
#include "GetCost/buildBusTra.h"
#include "GetCost/buildDummyLink.h"
#include "Dijkstra/Globals.h"
#include "Dijkstra/RoadNode.h"
#include "Output/Output.h"
#include <math.h>
char caLine[LENGTH_LINE] = {0};
int maxBusLine = 6;
int maxNode = 50+1;
int heapSize;
int main(int argc, char *argv[])
{
    //主要輸出檔
    FILE *fpSaveFile = NULL;
    char *cpSaveFile = "SaveFile.dat";
    //詳細路徑檔
    FILE *fpDetailPath = NULL;
    char *cpDetailPath = "DetailPath.dat";
    //公車路線 OD 量檔
    FILE *fpBusLineOd = NULL;
    char *cpBusLineOd = "BusLineOd.dat";
    double dSimInterval = 5.0, dAllInterval = 60.0f, dInterval = 0.1;
    double dPeopleVel = 66.67, dDelayTime;
    double dBusRunTime, dBusStartTime;
    double dDummyLinkBus, dDummyLinkRoad;
    double pentalty, dTmp;
    double aStart;
    int iStartStop, iEndStop;
    int g, h, i, j, k, m, n;
    int count = 0, iTmp = 0, iBusPeople = 50;
```




```
double dOD = 0.0f, rdOD = 0.0f, dODPart = 0.0f;
int iCost, iCase;
int source, destination;
int odIndex = 5, odIndexFrom;
int initOdFlag = TRUE;    //初始路徑 od 量
int initBusLine = TRUE;   //初始公車 od 量陣列
int bWaitingTimeFlag = FALSE; //增加等車成本
int bRemainingOdFlag = FALSE; //加班車
int bRunAnotherLine = FALSE; //設定走過公車路線成本為無窮大，再跑 Dijkstra
linkLine *spTmpLine = NULL, *spStartLine = NULL;
busLine *spTmpBusLine = NULL;
busTra *spTmpBusTra = NULL, *spStartBusTra = NULL;
rNodePointer ptr, odHead;
rNodePointer **adjArray2D;
rNodePointer lPath= NULL, sPath= NULL;
rNodePointer *rBusLineArray;

/* init. */
buildZone();
buildBus();
buildNetwork();
buildBusTra();
buildDummyLink();
newCost(); // 初始化成本陣列
buildDetransit(); // 讀取各時段區間之 OD 量
//開檔
openWriteFile(&fpSaveFile, &cpSaveFile);
openWriteFile(&fpDetailPath, &cpDetailPath);
openWriteFile(&fpBusLineOd, &cpBusLineOd);
printf("1.不考慮擁擠，不考慮容量限制\n"); //不考量加派車的等車時間，也不考慮剩下的人
人
printf("2.不考慮擁擠，考慮容量限制\n"); //不考量加派車的等車時間，考慮剩下的人(加班車)
班車)
printf("3.考慮擁擠，考慮容量限制\n"); //考量加派車的等車時間，考慮剩下的人
printf("4.考慮擁擠，不考慮容量限制\n");
do{
    printf("請輸入:\n");
    scanf("%d", &iCase);
```



```
}  
while( iCase<=0 || iCase>=5);  
switch(iCase){  
    case 1:  
        bWaitingTimeFlag=FALSE;  
        bRemainingOdFlag=FALSE;  
        break;  
    case 2:  
        bWaitingTimeFlag=FALSE;  
        bRemainingOdFlag=TRUE;  
        break;  
    case 3:  
        bWaitingTimeFlag=TRUE;  
        bRemainingOdFlag=TRUE;  
        break;  
    case 4:  
        bRunAnotherLine = TRUE;  
        bRemainingOdFlag=TRUE;  
        break;  
}  
dBusStartTime = (double) getStartTime(spBusTra);  
dSimInterval = (double) getBTraRunTime(spBusTra);  
dBusRunTime = (double) getBTraRunTime(spBusTra)/2;  
dDummyLinkRoad = 2 * dBusRunTime;  
dDummyLinkBus = 3 * dBusRunTime;  
pentalty = dBusRunTime;  
adjArray2D = createRArray2D(maxBusLine, maxNode);  
for( i = floatToInt(dBusStartTime) ; i <= floatToInt(dAllInterval)  ; i = i +  
floatToInt(dSimInterval) ){  
    for( g = 1 ; g <= iZoneNum ; g++ ){  
        for( h = 1 ; h <= iZoneNum ; h++ ){  
            odIndex = 5;  
            initOdFlag = TRUE;  
            //計算開始時間  
            odIndexFrom = (i/10) - getBTraRunTime(spBusTra) + 5;  
  
            source = getZoneCenter(g);  
            destination = getZoneCenter(h);
```

```

        //printf("*****%2.1f*****\n", (double)((int) dBusStartTime));
        cleanCost();
        // 整理公車路線成本 ( 第二次算法 )
        spTmpBusTra = spBusTra;
        while( ( spTmpBusTra = searchBusTra( spTmpBusTra, ((int) dBusStartTime) ) ) !=
        NULL ){
            for( k = 0 ; k < spTmpBusTra->iBusStopNum-1 ; k++ ){
                spTmpLine = searchLinkLine( spLine, spTmpBusTra->ipSeqBusLine[k],
                spTmpBusTra->ipSeqBusLine[k+1] );
                daCost[spTmpLine->iLinkID].dBusCost =
                spTmpBusTra->dpBusLineCost[k];
            }
            spTmpBusTra = spTmpBusTra->next;
        }
        // 計算公車與人之路段時間，與 delay time
        for( k = 0 ; k < iLinkNum ; k++ ){
            spTmpLine = getLinkLine(spLine, k);
            // 計算走路之成本
            daCost[k].dPeopleCost += spTmpLine->iLength/dPeopleVel;
        }

        //init memory
        initRArray2D(adjArray2D, maxBusLine, maxNode);
        //建立路網資訊 (一般路網)
        k = 0;
        spStartLine = spLine;
        while( spStartLine != NULL ){
            spTmpLine = getLinkLine(spLine, k);
            if ( spTmpLine->iFromNode == iTmp ) {
                //建相同起始點的目的點
                iTmp = spTmpLine->iFromNode;
                insertRNode(&ptr, createRNode(0, spTmpLine->iToNode, 2 *
                daCost[k].dPeopleCost));
            }
            else{
                //建起始點
                iTmp = spTmpLine->iFromNode;
                adjArray2D[0][spTmpLine->iFromNode] = createRNode(0,

```

```
spTmpLine->iToNode, 2 * daCost[k].dPeopleCost);
    ptr = adjArray2D[0][spTmpLine->iFromNode];
}
k++;
spStartLine = spStartLine->next;
}

//建立路網資訊 (公車路網)
k = 1;
spTmpBusTra = spBusTra;
while ( ( spTmpBusTra = searchBusTra(spTmpBusTra, (int) dBusStartTime) ) !=
NULL ){
    for( m=0 ; m<spTmpBusTra->iBusStopNum-1 ; m++){
        spTmpLine = searchLinkLine(spLine, spTmpBusTra->ipSeqBusLine[m],
spTmpBusTra->ipSeqBusLine[m+1]);
        if(adjArray2D[k][spTmpBusTra->ipSeqBusLine[m]] != NULL){
            //取得 tail node
            ptr = getTailNode(adjArray2D[k][spTmpBusTra->ipSeqBusLine[m]]);
            insertRNode(&ptr, createRNode(k, spTmpBusTra->ipSeqBusLine[m+1],
daCost[spTmpLine->iLinkID].dBusCost));
        }
        else{
            adjArray2D[k][spTmpBusTra->ipSeqBusLine[m]]
                = createRNode(k, spTmpBusTra->ipSeqBusLine[m+1],
daCost[spTmpLine->iLinkID].dBusCost);
        }
    }
    k++;
    spTmpBusTra = spTmpBusTra->next;
}

//建立公車路線陣列
if(initBusLine){
    rBusLineArray = createRArray(maxBusLine);
    k = 1;
    spTmpBusTra = spBusTra;
    while ( ( spTmpBusTra = searchBusTra(spTmpBusTra, (int) dBusStartTime) ) !=
NULL ){
        for( m=0 ; m < spTmpBusTra->iBusStopNum-1 ; m++){
            spTmpLine = searchLinkLine(spLine, spTmpBusTra->ipSeqBusLine[m],
```

```

spTmpBusTra->ipSeqBusLine[m+1]);
        if(rBusLineArray[k] != NULL){
            //取得 tail node
            ptr = getTailNode(rBusLineArray[k]);
            insertRNode(&ptr, createRNode(k,
spTmpBusTra->ipSeqBusLine[m], daCost[spTmpLine->iLinkID].dBusCost));
        }
        else{
            rBusLineArray[k]
                = createRNode(k, spTmpBusTra->ipSeqBusLine[m],
daCost[spTmpLine->iLinkID].dBusCost);
        }
    }
    ptr = getTailNode(rBusLineArray[k]);
    insertRNode(&ptr, createRNode(k, spTmpBusTra->ipSeqBusLine[m],
daCost[spTmpLine->iLinkID].dBusCost));
    k++;
    spTmpBusTra = spTmpBusTra->next;
}
for(k=1 ; k < maxBusLine ; k++){
    ptr = rBusLineArray[k];
    initOdValue(&ptr);
}
initBusLine = FALSE;
}

//建立 dummy link(一般路網到公車路網)，k=停靠站索引，m=公車路線索引
for(k=0; busStops[k] != 0 ; k++){
    //取得起始點的最後一個 Linking-list 的 ptr
    ptr = getTailNode(adjArray2D[0][busStops[k]]);
    for(m=1; m < maxBusLine ; m++){
        //目標點不為空&&目標點不為迄站***
        if(adjArray2D[ m ][ busStops[k] ] != NULL){
            insertRNode(&ptr, createRNode(m, busStops[k], dDummyLinkRoad));
        }
    }
}

//建立 dummy link(公車路網到一般路網)，k=公車路線索引，m=停靠站索引
//(迄站)要建立來源點的頭點

```

```

for(k=1; k < maxBusLine ; k++){
    //取得起始點的最後一個 Linking-list 的 ptr
    for(m=0; busStops[m] != 0 ; m++){
        //來源點不為空&&來源點不為起站***
        if(adjArray2D[ k ][ busStops[m] ] != NULL){
            ptr = getTailNode(adjArray2D[k][busStops[m]]);
            insertRNode(&ptr, createRNode(0, busStops[m], dDummyLinkRoad));
        }
    }
}
//建立公車迄點到一般路網的 dummy link
k = 1;
spTmpBusTra = spBusTra;
while ( ( spTmpBusTra = searchBusTra(spTmpBusTra, (int) dBusStartTime) ) !=
NULL ){
    iStartStop = spTmpBusTra->ipSeqBusLine[0]; // 起點
    iEndStop = spTmpBusTra->ipSeqBusLine[spTmpBusTra->iBusStopNum-1]; //
終點
    if(searchBusStops(busStops, iEndStop)){
        adjArray2D[k][iEndStop] =
            createRNode(0, iEndStop, dDummyLinkRoad);
    }
    k++;
    spTmpBusTra = spTmpBusTra->next;
}
//建立 dummy link(公車路網到公車路網)
for(k=1; k < maxBusLine ; k++){
    for(m=0; busStops[m] != 0 ; m++){
        //來源點
        if(adjArray2D[ k ][ busStops[m] ] != NULL){
            ptr = getTailNode(adjArray2D[k][busStops[m]]);
            for(n=1 ; n < maxBusLine ; n++){
                //目標點
                if( (n != k) && (adjArray2D[ n ][ busStops[m] ] != NULL) )
                    insertRNode(&ptr, createRNode(n, busStops[m], dDummyLinkBus));
            }
        }
    }
}

```

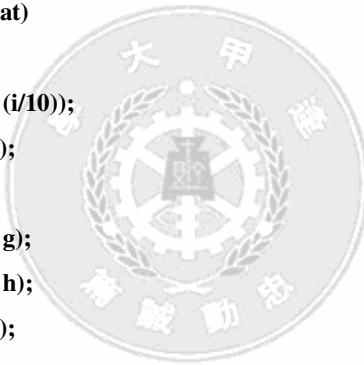
```
}  
//Dijkstra Algorithm  
IPath =(rNodePointer) dijkstra(adjArray2D, source, destination);  
//轉換為簡單表示法  
sPath = convertSimplePath(IPath);  
//顯示最短路徑  
//displayRNode(sPath);  
//displayRNode(IPath);  
//printf("Total cost = %f\n", getTotalCost(IPath));  
//計算 OD 量  
if(initOdFlag){  
    odHead = sPath;  
    ptr = odHead;  
    initOdValue(&ptr);  
    initOdFlag = FALSE;  
}  
//od Total(5 ~ i/10)  
dOD = 0.0f;  
while(odIndex <= i/10){  
    dOD += getODNum( getZoneNum(source), getZoneNum(destination), odIndex);  
    odIndex+=5;  
}  
//od part  
dODPart = 0.0f;  
while(odIndexFrom <= i/10){  
    dODPart += getODNum( getZoneNum(source), getZoneNum(destination),  
odIndexFrom);  
    odIndexFrom+=5;  
}  
//累加 OD 量  
if( !isFloatEqual(dOD, 0.0f, 0.001) && compareLists(odHead, sPath) == true){  
    ptr = odHead;  
    addOdValue(&ptr, dOD);  
}  
if(bRunAnotherLine){  
    //計算剩於 OD 量  
    rdOD = rdOD + dODPart;  
    do{
```

```
//寫入公車路徑檔(DetailPath.dat)
writeDetailPath(fpDetailPath, (i/10), lPath);

//寫入檔案(SaveFile.dat)
//寫入時間
putToken(fpSaveFile, (i/10));
putEscape(fpSaveFile);
//寫入起迄中心
putToken(fpSaveFile, g);
putToken(fpSaveFile, h);
putEscape(fpSaveFile);
//寫入最短路徑
writeShortestPath(fpSaveFile, sPath);
//寫入累加 OD 量
writeSumOd(fpSaveFile, odHead, sPath);
//寫入總時間成本
writeTotalCost(fpSaveFile, (int)getTotalCost(lPath));
//累加 BusLine 的 OD 量
if( (rdOD - iBusPeople) < 0.0f )
    addBusLineOd(lPath, rBusLineArray, rdOD);
else{
    addBusLineOd(lPath, rBusLineArray, iBusPeople);
}
//計算剩於 OD 量
rdOD = rdOD - iBusPeople;
if(rdOD < 0.0f)
    rdOD = 0.0f;
//寫入剩於 od 量
writeRemainingOd(fpSaveFile, rdOD);
putEscape(fpSaveFile);
cleanBusLineCost(adjArray2D, lPath);
//Dijkstra Algorithm
lPath =(rNodePointer) dijkstra(adjArray2D, source, destination);
sPath = convertSimplePath(lPath);
odHead = sPath;
ptr = odHead;
initOdValue(&ptr);
ptr = odHead;
```



```
        addOdValue(&ptr, dOD);
    }
    while(rdOD > 0.0f);
    //公車路線 OD 量檔(BusLineOd.dat)
    writeBusLineOd(fpBusLineOd, rBusLineArray, (i/10), g, h);
}
else{
    //累加 BusLine 的 OD 量
    addBusLineOd(lPath, rBusLineArray, dODPart);
    //公車路線 OD 量檔(BusLineOd.dat)
    writeBusLineOd(fpBusLineOd, rBusLineArray, (i/10), g, h);
    //寫入公車路徑檔(DetailPath.dat)
    writeDetailPath(fpDetailPath, (i/10), lPath);
    //公車路線 OD 量檔(BusLineOd.dat)
    writeBusLineOd(fpBusLineOd, rBusLineArray, (i/10), g, h);
    //寫入檔案(SaveFile.dat)
    //寫入時間
    putToken(fpSaveFile, (i/10));
    putEscape(fpSaveFile);
    //寫入起迄中心
    putToken(fpSaveFile, g);
    putToken(fpSaveFile, h);
    putEscape(fpSaveFile);
    //寫入最短路徑
    writeShortestPath(fpSaveFile, sPath);
    //寫入累加 OD 量
    writeSumOd(fpSaveFile, odHead, sPath);
    //寫入總時間成本
    writeTotalCost(fpSaveFile, (int)getTotalCost(lPath));
    if(bRemainingOdFlag){
        //計算剩於 OD 量
        rdOD = rdOD + dODPart - iBusPeople;
        if(rdOD < 0.0f)
            rdOD = 0.0f;
        //寫入剩於 od 量
        writeRemainingOd(fpSaveFile, rdOD);
        putEscape(fpSaveFile);
        //加班車
```



```

aStart = i/10;
iCost = (int)getTotalCost(IPath);
for(k = 1 ; rdOD > 0.0f || (aStart>=60 && rdOD>0.0f) ; k++){
    aStart = aStart + dSimInterval*pow(0.5, k);
    //printf("----aStart=%f----\n", aStart);
    //增加成本
    if(bWaitingTimeFlag)
        iCost = iCost + dBusRunTime;
    //寫入時間
    putTokenDouble(fpSaveFile, aStart);
    putEscape(fpSaveFile);
    //寫入起迄中心
    putToken(fpSaveFile, g);
    putToken(fpSaveFile, h);
    putEscape(fpSaveFile);
    //寫入最短路徑
    writeShortestPath(fpSaveFile, sPath);
    //寫入累加 OD 量
    writeSumOd(fpSaveFile, odHead, sPath);
    //寫入總時間成本
    writeTotalCost(fpSaveFile, iCost);
    //計算剩於 OD 量
    rdOD = rdOD - iBusPeople;
    if(rdOD < 0.0f)
        rdOD = 0.0f;
    //寫入剩於 OD 量
    writeRemainingOd(fpSaveFile, rdOD);
    putEscape(fpSaveFile);
} //end for loop
} //end for bRemainingOdFlag
else{
    putEscape(fpSaveFile);
    putEscape(fpSaveFile);
}
}
//system("pause");
} //end for h
} //end for g

```

```
}  
    freeRArray2D(adjArray2D, maxBusLine);  
    freeCost();  
    fclose(fpSaveFile);  
    system("PAUSE");  
    return 0;  
}
```

